Gas Liquid Interface Studied by Molecular Beam Scattering From a Liquid Flat Jet

by

Chin Lee

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Chemistry

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Daniel M. Neumark, Chair
Professor Richard J. Saykally
Professor Evan R. Williams
Professor David B. Graves

Fall 2022

Gas Liquid Interface Studied by Molecular Beam Scattering From a Liquid Flat Jet

Abstract

Gas Liquid Interface Studied by Molecular Beam Scattering From a Liquid Flat Jet

by

Chin Lee

Doctor of Philosophy in Chemistry

University of California, Berkeley

Professor Daniel M. Neumark, Chair

A newly designed molecular beam scattering apparatus that utilizes a flat liquid jet to study chemical reactions at the gas-liquid interface is reported. The flat liquid surface combined with a rotatable mass spectrometer that provides energetic, mass and angular selectivity allows for unprecedented insights into this highly relevant chemical interface. To provide a proof of feasibility for studying volatile liquid systems, experiments are performed on dodecane evaporation as well as dodecane as a scattering target for neon beams.

In the first part of this dissertation, the evaporation of neon and dodecane from a neon-doped dodecane flat jet running inside vacuum is investigated. We demonstrate that it is possible to observe sensible velocity and angular distributions using this flat jet setup. The effect of near-surface vapor interactions with evaporating particles is observed by comparing dodecane and Ne evaporation, where both particles show different collision cross sections with the vapor. The angular distribution for both particles also follows the cosine distribution, which has been investigated in gas desorption from solid surfaces.

Then, further exploration of neon scattering off the dodecane surface at varying incident translational energies and angles is performed. We demonstrate how the energy and angular sensitivity of our apparatus provides complete information to map out energetic processes at the liquid interface. Energy loss of the scattered neon atoms can be attributed to excitation of liquid surface modes and is well described by a “soft-sphere” kinematic model, from which we infer effective surface masses and total internal excitation from the collision during the collisional timescale. We believe that our results demonstrate the high potential of flat liquid jet scattering with regards to studying dynamics at the gas-liquid interface. By extending the experiment to water, which is planned for future experiments, we are opening up new experimental opportunities that are highly relevant to water-based catalysis and atmospheric chemistry.

To my parents

Pin-Pin Chang and Yin-Yu Lee

Contents

List of Figures

List of Tables

Acknowledgments

My grad student life from August 2017 to December 2022 has been like the San Francisco marathon I took in 2022. I signed up excitingly at the beginning and kept questioning myself during the run, especially when my blood glucose decays and running fatigue kicks in. Getting a PhD degree is like in a continuously energy-draining process. I wouldn't be able to hang in there till the end without supports from so many people. They are the sport drinks, energy chews, and nutrition gels that boost me through this PhD marathon. I wish to share my greatest gratitude to all of them.

First, I must thank my advisor, Daniel Neumark, for his mentorship on my PhD work. Since my project is mainly transforming the research direction of XBeam, I spent 60% to 80% of time on designing and building new equipment. Dan often brings up amazing ideas on designing these new devices. I am very inspired by him being so flexible on viewing the problem at different aspects and enjoy brainstorming with him. I am also grateful for him giving me so much freedom on trying different experimental parameters on the machine, which I often want to test them all. Even though going through all the tuning processes takes a long time, Dan is always patience for waiting me to figure it out. I am humbled to have him guiding me through my research.

Secondly, I wish to thank Bernd Winter at Fritz Haber Institute for letting me stay in his group as an observer and learner in fall 2019. My vision has been broadened by learning how liquid microjets can be used in photoelectron spectroscopy. Bernd have been so willing to share his great knowledge on handling liquid jets under vacuum. I would not be able to finish developing our flat jet scattering setup at Berkeley without his help. Also, many thanks to Uwe Hergenhahn and Florian Trinter for teaching me great knowledge about PES data analysis during beamtime at DESY and BESSY. Sebastian Malerz and Christian Schewe hosted me during my stay at Berlin. They prove my stereotype about German, who are very precise on performing analytical tasks in research and being fully relaxed after couple mugs of beer.

Thanks to Isaac Ramphal for teaching me everything in Latimer D10. He is the soul of the liquid jet project. Isaac enlightened me every morning when I come to the lab by proposing exciting new jet experiments, which includes using a tequila shot glass made by pink Himalayan salt for potential water jet collection. He also shares his passion of all kinds of memes and British shows, which becomes very helpful for me to understand American jokes. Marvin Pohl has been a super reliable lab mate and always worked very efficiently. I will never forget how fast he can finish a drawing on Fusion 360, then 3D prints it, and tests it out on our setup. All these processes usually are done within one week. I am also grateful for Marvin introducing me Mettbrötchen, a sandwich topped with raw minced pork, which has become my favorite German food. Walt Yang joins the project in a great time after all the buildings are done. I am lucky to have him joining the lab and sharing his enthusiasm

of science. Steve Saric also joins the project very recently but has picked up all the required techniques fairly fast. I have no doubt that they will bring the project to the next stage of publishing tons of papers.

I also wish to thank Doug Scudder and Clint Jessel at the machine shop for their supportive consulting and excellent machining works. Thanks to Yi-Min Hsieh in the pump shop for keeping our old Welch pumps alive. I wish them all enjoy their retirement in the coming future! People in the cryogenic plant, Steven Williams and Kinfe Lita, keep the liquid nitrogen station running 24 hr that we never need to worry about running out of LN_2. Emanuel Druga and David Karst give us precious experiences on designing electronic devices and how to repair them. Thanks to the entire wood shop members, Michael Brateng, Ryan Woloshyn, Jody Brenner and Jihad Poole, for lending us numerous NPT adaptors for plumbing and cutting metal pipes.

Many thanks to my qualify exam/dissertation reviewing committees, Richard Saykally, Evan Williams, Martin Head-Gordon and David Graves. They give me tremendous insights on liquid jets and surface scattering dynamics. I wish to thank Professor Gil Nathanson at University of Wisconsin-Madison for sharing precious experiences on running a salty microjet. Also, many thanks to his grad students for writing their theses in nice content. I am able to work further by standing on their shoulders. I would also like to thank our administrative assistants, Michelle Haskins and Kathleen Buckley, for helping me through the reimbursement forms and LBNL's orders. And of course, many thanks to the entire Neumark group for numerous useful research discussions, equipment/chemicals lending and relaxing group beer time. Many thanks to all my bouldering, snowboarding, national park, traveling and running buddies for having me to share these adventures together.

Finally, I would like to thank my parents for their support and encouragement through my whole life. They embraced my ups and downs no matter what, where and when. Without their nourishment, I would not be here and finished this long PhD marathon. :D

Chapter 1

Introduction

Since the main point of this dissertation is to show the feasibility of a newly developed flat jet molecular beam scattering apparatus to study gas-liquid interfacial reactions, this chapter is used give brief introduction for the readers to step into the field. First, we introduce the importance and motivation of studying gas-liquid interfaces. Then in the second part, historical summary of how researchers study it and why liquid flat jet is proposed are given. In the final part, comparison between the already known cylindrical jet and the newly proposed flat jet are shown using different estimating methods. Pros and cons are then also listed and discussed.

1.1 Chemistry at the Gas-Liquid Interface

Heterogeneous reactions have been acting as interesting chemical environments and providing unique reactions to happen at the phase boundaries. Interfacial reaction at gas-liquid interface is particularly interesting to study due to its ubiquitous occurrence in daily life, such as in industrial processes, atmospheric chemistry, and environmental science. Examples include, but are not limited to, combustion processes inside the automotive engines which includes air-fuel mixing,[1] acid rain formation in clouds,[2, 3, 4] tropospheric chemistry on liquid and solid aerosols surfaces[5] and carbon dioxide uptake at the ocean surface, where around one-third of anthropogenic CO_2 is absorbed.[6] These reactions play important roles and highly affecting our life. Therefore, a molecular-level of understanding of the chemistry at the gas-liquid interface is desired and motivates us to develop a new technique to study it.

Mechanism

A liquid shares similar properties with a solid as it has more flexible molecular arrangement but can hold a confined geometry as long as there are some kind of outer support or inter-molecular interactions to hold its form. Thus, it is straightforward to studied gas-liquid interfaces based on the knowledge of gas-solid interfaces. Considering a non-reactive process, two possible mechanisms have been found when a gas particle impinging onto a solid surface: the impulsive scattering (IS) and thermal desorption (TD) processes.[7, 8]

These two mechanisms can be distinguished by their outgoing particles after being collided with the solid surface. The impulsive scattering process, sometimes called direct inelastic scattering, is a prompt process where the incident particles spend a shorter time interacting with the surface before recoil. The scattered particles are therefore only partially equilibrated with the surface molecule and retain most information of the incident beam. For example, the translational energy distribution is expected to be similar to the incoming gas and the scattered direction is angular specific to the specular angle.

Thermal desorption, or called trapping-desorption scattering, is a process that the gas particles being fully thermalized with the liquid molecules then desorb from the surface into gas-phase or vacuum. Since the gas particles have been equilibrated with the liquid molecules, their desorbed velocity distribution has been predicted to be following the Maxwell-Boltzmann distribution, where the probability distribution for gas particles moving at speed v is

$$f_{MB}(v) = \left(\frac{m}{2\pi k_B T}\right)^{3/2} 4\pi v^2 \exp\left[-\frac{mv^2}{2k_B T}\right] \tag{1.1}$$

where m and T are the mass and temperature of the particles. k_B is the Boltzmann constant. As for the angular distribution, one would expect to follow the cosine law, which is similar as the Lambert's cosine law for a point light source emitting on a surface.[9] However, gas desorbing from solid surface is different from light for having a potential energy well between gas particles and solid surfaces. This well raises an extra interaction term that needs to be considered during the desorbing process and therefore a simple cosine law is no longer suitable. Researchers have traditionally used an empirical form of $\cos^n \theta_f$ instead of $\cos\theta_f$ to describe this process, where θ_f is the outgoing angle and n is a constant usually ranges from 0.6 to 12.[10]

A possible explanation for the non-cosine angular distribution could be influenced by surface impurities. Nitrogen desorbing from tungsten[310] is a great study to show this hypothesis.[11] In this study, the angular distribution of nitrogen desorption from a nitrogen saturated tungsten crystal is observed to be following the cosine law (see Figure 1.1 (a)). However, if the saturated crystal is exposed to oxygen before nitrogen, the desorbing angular distribution is shown to be a bimodal form consists of $\cos\theta_f$ and $\cos^n \theta_f$ (Figure 1.1 (b)). The former one can be easily understood as the normal desorbing process. On the other hand, the later contributes from the interaction between the oxygen adsorbent and the desorbing nitrogen on the surface.

The mechanisms occurring on the gas-solid interface mentioned above show the complexity of interactions on the heterogeneous interfaces. One can imagine the processes for gas-liquid interface being much more complicated due to the flexibility of the liquid phase

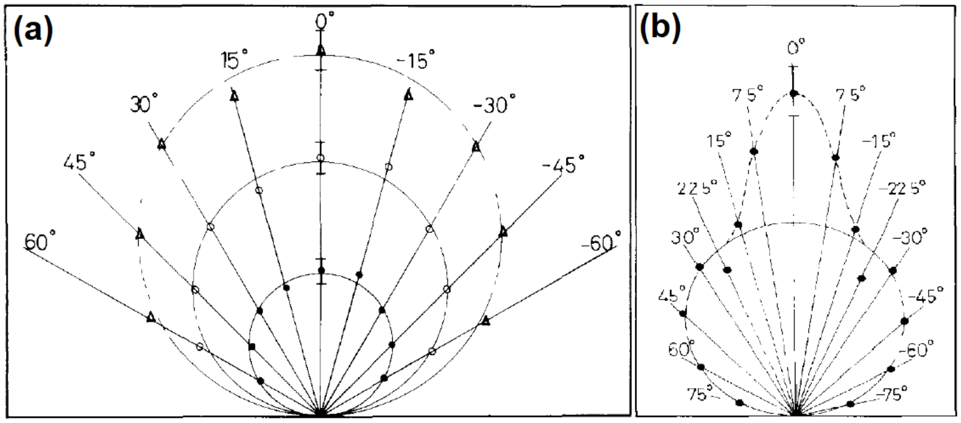


Figure 1.1: Angular plots for nitrogen evaporating from a tungsten[310] surface. (a) Different nitrogen coverage conditions are performed. All of them show cosine desorption dependence. (b) Saturated nitrogen desorption before exposing to oxygen. Both figures are reprinted from Cosser's paper.[11]

providing a more undefined geometry and unique chemical environment. Therefore, the motivation and goal for this dissertation is to develop a new technique to investigate the gas-liquid interfacial reactions which provides more insights to distinguish different mechanisms happen on the liquid surface.

1.2 Methods to Study Gas-Liquid Interface

Many different techniques have been demonstrated to study gas-liquid reactions in the bulk or surface of the liquid.[12, 13, 14] For example, second harmonic sum-frequency generation (SFG), which relies on symmetry-breaking effects at the liquid surface, provides spectroscopic surface-specific information.[15, 16, 17] X-ray photoelectron spectroscopy (XPS) of liquid microjets[18, 19, 20, 21] and aerosols[22] is used to probes solute and solvent molecules lying on the liquid interface with depth of 1–3 nm. Using Langmuir troughs enables analysis of photochemical products at both liquid surface and bulk.[23] Mass spectrometry-based experiments using thin films and droplets to carry microemulsions under vacuum are performed and the gas-phase show significantly enhanced surface chemical reaction rates compared to that in bulk liquids.[24, 25, 26]

Molecular beam scattering, which is originally developed from probing gas-phase interactions[27, 28] and heterogeneous reactions of gas-solid interfacial reactions,[29, 30] has proved to be a direct and powerful method to investigate the dynamics of elementary chemical reactions. Nathanson and others have applied molecular beam scattering technique to gas-liquid interfacial reaction and have observed mechanisms of impulsive scattering and thermal desorption to occur on the surface.[31] This shows the great benefit and potential of using molecular beam scattering technique on exploring gas-liquid interfacial reactions.

To perform molecular beam scattering experiments, high vacuum environment (pressure $< 10^{-5}$ Torr) in the scattering region and a clean sample surface is required. Both requirements are very challenging for liquid samples, especially volatile liquids. Having a liquid sample sitting inside the vacuum chamber is essentially ruining the vacuum with the evaporating liquid molecules. Additionally, the evaporating cooling effect also lowers the liquid temperature and could freeze up the liquid sample into solid. Thus, two major liquid sources have been developed, the wetted wheel and microjet. Both will be discussed in the next Subsection. In the end of this Section, a flat jet is proposed to be the next generation of the liquid source, which combines the advantages of both cylindrical jet and wetted wheel.

Wetted Wheel

In order to keep high vacuum, only liquids with super low vapor pressure below 10^{-8} Torr can be investigated. For example, perfluoropolyether (PFPE) and squalane are studied with colliding neon, xenon and sulfur hexafluoride.[32] The results show more efficient energy transfer of the hydrocarbon liquid surface than the perfluorinated surface.

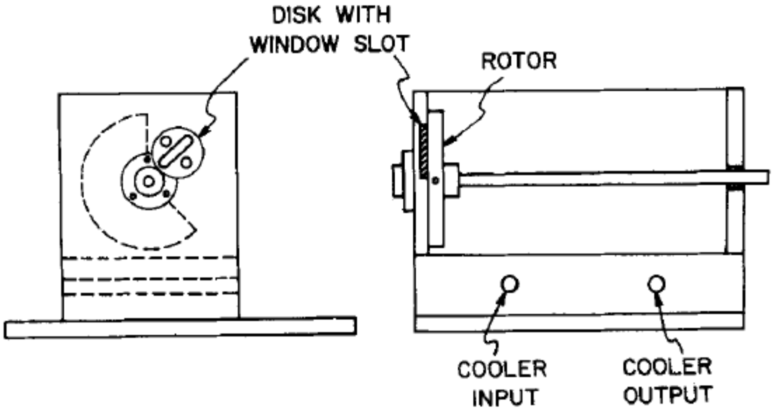


Figure 1.2: Schematic figure of the wetted wheel. Liquid sample is stored inside the box. The rotating plate, which is usually made by glass or metal, is immersed into the liquid sample. When the rotor rotates, part of the liquid will be dragged then exposed to the window slot, which the size is around millimeter-scale. This entire setup can be temperature controlled by cooler, which its inlet and outlet are shown on the right of the figure. This figure is reprinted from Fenn's paper.[33]

The technique that has been used for these experiment to carry low vapor pressure liquids into the vacuum chambers is so-called wetted wheel.[33, 34] This robust technique is developed by Fenn and co-workers. It contains a rotating wheel which is immersed into an enclosed liquid sample container and only a millimeter size of window opens to vacuum. When the wheel rotates, part of the liquid will be grabbed by the wheel and exposed to vacuum when the wheel rotates to the opening window. A schematic figure of the wetted wheel setup is shown in Figure 1.2.

This wetted wheel technique provides great temperature control for the liquid sample and can be used for thermal roughness of the liquid surface. With careful handling of the wetted wheel, the vapor pressure limit for the liquid sample has been pushed to $< 10^{-3}$ Torr. Liquid samples that are suitable for wetted wheel include long-chain hydrocarbons,[35] glycerol,[36] ionic liquids,[37] salty water at 212 K,[38] and perfluoropolyether.[39]

Cylindrical Liquid Microjet

In order to study scattering experiments on more volatile liquids, such as room temperature water, Faubel and co-workers have developed the liquid microjet technique.[40, 41] This device generates a micrometer thin cylindrical liquid jet by pumping the liquid through a micrometer sized orifice. The curved geometry of the cross-section of a cylindrical jet provides better vapor expansion than a flat surface. This leads to a better decreasing of overall vapor pressure P_{vap} above the surface and dramatically lower the collision number N_{coll} of the scattered gas particle with the liquid vapor. In Section 1.3, more details of how to estimate N_{coll} for flat and curvature surfaces will be described.

Another benefit for using a cylindrical jet is that it has strong evaporative cooling effect when the jet exposed to vacuum. Combining the fact that cylindrical jet is a flowing system and will keep pushing fresh liquid throughout the jet orifice, one can use this liquid source to study supercooled liquid by evaporative cooling the jet. The jet will not freeze up and remain as liquid form until it reaches surfaces with nuclei to condense. This is something that wetted wheel can never achieves.

With all the benefits of the microjet, Nathanson and co-workers were the first group to apply cylindrical jet in scattering experiment.[42] Liquids like pure water and other volatile liquids can now be brought into vacuum. Examples of the experiments include collisions of DCl molecules with salty water at 238 K,[43] scattering of Ne and O_2 from liquid dodecane,[42] collisions of Cl_2 and N_2O_5 with surfactant-coated Br^--glycerol solutions,[44, 45] and collisions of organic molecules with salty water.[46]

Although the cylindrical microjet provides accessibility to study volatile liquids, its geometry is not suitable for scattering experiments due to the curvature of its surface. This decreases the angular specificity in a scattering experiment. Another downside of the microjet is its small scattering area. A typical jet diameter is 5–50 µm which is considerably smaller than the molecular beam diameter (1–2 mm). This small scattering target therefore increases the data accumulation time. Background from the scattering chamber could also have a relatively larger influence on the data. Therefore, a new liquid source for volatile samples is desired.

Flat Liquid Jet

A new liquid source suitable for high vapor pressure liquids has been proposed, which is the so-called flat liquid jet. Flat jets are usually generated by two colliding cylindrical jets or a slit nozzle.[47, 48] At the jets' colliding center, a millimeter-sized flat liquid sheet is formed. After the first sheet, several sheets perpendicular to each other are then formed downstream.

This type of liquid source seems to be suitable for scattering experiment because of its comparable size to a molecular beam, and therefore acting as a great scattering target. Besides, the relatively flat surface on the jet provides angular specificity similar to the wetted wheel. Last but not least, the flat jet is a continuously flowing system and will keep refreshing the scattering surface. Plus, the evaporating cooling effect can further lower the jet to a supercooled temperature and further lower the vapor pressure.

This kind of liquid source has been used inside vacuum and shows great potential to explore liquid system.[48, 49, 50, 51, 52] To our understanding, however, no one has applied it to molecular beam scattering experiments. With all the benefits of the flat jet mentioned above, our group proposes to use flat jet as a new liquid source for scattering experiments. The goal of the work in this dissertation is to extend the capabilities of gas-liquid scattering experiments to the surfaces of volatile liquid flat jets, enabling both the speed and angular distributions of escaping molecules to be determined.

1.3 Cylindrical Jet vs. Flat Jet

To compare the differences between cylindrical jet and flat jet, two aspects can be discussed. One is the evaporative cooling of a liquid jet. This phenomenon explains the liquid temperature being cooled down under vacuum along the jet axis. The second aspect is the number of collisions for a outgoing particle leaving the liquid surface then reaches to the detector. In the later Subsections, both aspects are described by theory and the hope is to use it to estimate the experimental condition and to give a characteristic view of both jets.

Evaporative Cooling

For liquid placing inside vacuum, it is continuously evaporating due to the low pressure of the surrounding. The temperature of the liquid is therefore decrease and this phenomenon is so-called evaporative cooling. To estimate the temperature gradient along the jet axis (z-axis), dT/dz, the Faubel's group suggested that it can be written as[40]

$$\frac{dT}{dz} = -2\frac{\frac{dr}{dt}}{v_{jet}}\frac{\Delta H_{vap}}{C_p}\frac{1}{r_{jet}} \tag{1.2}$$

where r and dr/dt are the jet radius and the radial evaporative ablation rate. The later one can be transformed from time dependence into z-axis dependence by dividing it with the jet velocity along the z-axis $dz/dt = v_{jet}$, which is why v_{jet} is included in Equation 1.2.

ΔH_{vap} and C_P are enthalpy of vaporization and heat capacity of the liquid at local liquid temperature T. Since jet radius is shrinking due to evaporation, the form of dr/dt can be theoretically defined based on evaporation theory $(dr/dt)_{evap}$. The Faubel's group has defined $(dr/dt)_{evap}$ as the product of mean velocity of the evaporating molecules and the density ratio of gas and liquid molecules. The total equation of dr/dt can be written as

$$\left(\frac{dr}{dt}\right)_{evap,Faubel} = \bar{u}\frac{\rho_{gas}}{\rho_{liq}} = \sqrt{\frac{8k_BT}{\pi m_{gas}}}\frac{\rho_{gas}}{\rho_{liq}} \tag{1.3}$$

where $\bar{u}$ is the mean velocity of the evaporating molecules. This value is assumed to be Maxwellian distribution at temperature T with equation $\bar{u} = \sqrt{(8k_BT)/(\pi m_{gas})}$, where k_B is the Boltzmann constant and m_{gas} is the mass of a single gas particle. ρ_{gas} and ρ_{liq} are the densities of gas and liquid phase. Applying Equation 1.3 back into 1.2, the final equation suggested by the Faubel's group is now shown as

$$\left(\frac{dT}{dz}\right)_{evap,Faubel} = -\sqrt{\frac{32k_BT}{\pi m_{gas}}}\frac{\rho_{gas}\Delta H_{vap}}{\rho_{liq}r_{jet}C_p}\frac{1}{v_{jet}} = -\sqrt{\frac{32m_{gas}}{\pi k_BT}}\frac{P_{vap}\Delta H_{vap}}{\rho_{liq}r_{jet}C_p}\frac{1}{v_{jet}} \tag{1.4}$$

where ρ_{gas} is transferred into the vapor pressure of the gas particles P_{vap} by ideal gas law $\rho_{gas} = (P_{vap}m_{gas})/(k_BT)$, where the unit of ρ and P_{vap} are weight per unit volume (commonly being $\mathrm{kg\,m^{-3}}$) and . With this liquid jet temperature estimating equation, the Nathanson's group also follow the same form proposed by Faubel's group.[53] One should note that the Saykally's group also claimed in their text that they also use the same temperature and jet radius gradient in their research.[54] However, the radial ablation rate in Equation 2 in this paper does not use $\sqrt{(8k_BT)/(\pi m_{gas})}$ for $\bar{u}$. Instead, they use $\sqrt{(k_BT)/(2\pi m_{gas})}$, which is unclear why or if this is just a typo.

Nevertheless, the Saykally's[17] and the Cohen's[55] group later proposed a different form for the radial evaporative rate using the Hertz–Knudsen equation[56, 57]

$$\left(\frac{dr}{dt}\right)_{evap,Saykally} = \frac{J_{e,obs}}{\rho_{liq}} = \frac{\phi_e J_{e,max}}{\rho_{liq}} = \phi_e\frac{P_{vap}}{\sqrt{2\pi m_{gas}k_BT}}\frac{1}{\rho_{liq}} \tag{1.5}$$

where $J_{e,max}$ is the theoretical maximum evaporation rate defined by the Hertz–Knudsen equation $J_{e,max} = P_{vap}/\sqrt{2\pi m_{gas}k_BT}$ with unit of number of molecules per unit area (m^2) per unit time (s). However, a calibrated evaporation rate $J_{e,obs}$ is required in order to show the actual experimental value. The conversion is made by having an empirical constant ϕ_e with relation of $J_{e,obs} = \phi_e J_{e,max}$. With the newly defined dr/dt by Equation 1.5, we can apply it back to Equation 1.2 and get

$$\left(\frac{dT}{dz}\right)_{evap,Saykally} = -\sqrt{\frac{2\phi_e^2}{\pi m_{gas}k_BT}}\frac{P_{vap}\Delta H_{vap}}{\frac{\rho_{liq}}{m_{gas}}r_{jet}C_p}\frac{1}{v_{jet}} \tag{1.6}$$

where m_{gas} is added in Equation 1.6, since the unit of ρ_{liq} is usually $\mathrm{kg\,m^{-3}}$ and therefore requires this extra term to transform the unit to $\#\,\mathrm{m^{-3}}$. One should note that the same equation at Equation 9 in Cohen's paper has a typo on term v_{jet}, where it should be at the denominator instead of the numerator.[55]

Now if we compare Faubel's and Saykally's equations for dT/dz, it can be easily seen that the two equations give different estimated results. System of a water cylindrical jet with diameter of 27.6 µm and nozzle temperature T_{nozzle} at 285 K are tested. The jet velocity v_{jet} is estimated to be $13.9\,\mathrm{m\,s^{-1}}$. The jet conditions mentioned above are not randomly picked, instead, they are quite common in our current setup. The jet temperature curves are shown in Figure 1.3. Mathematica codes that used to calculate T_{jet} are shown in Appendix A.2.

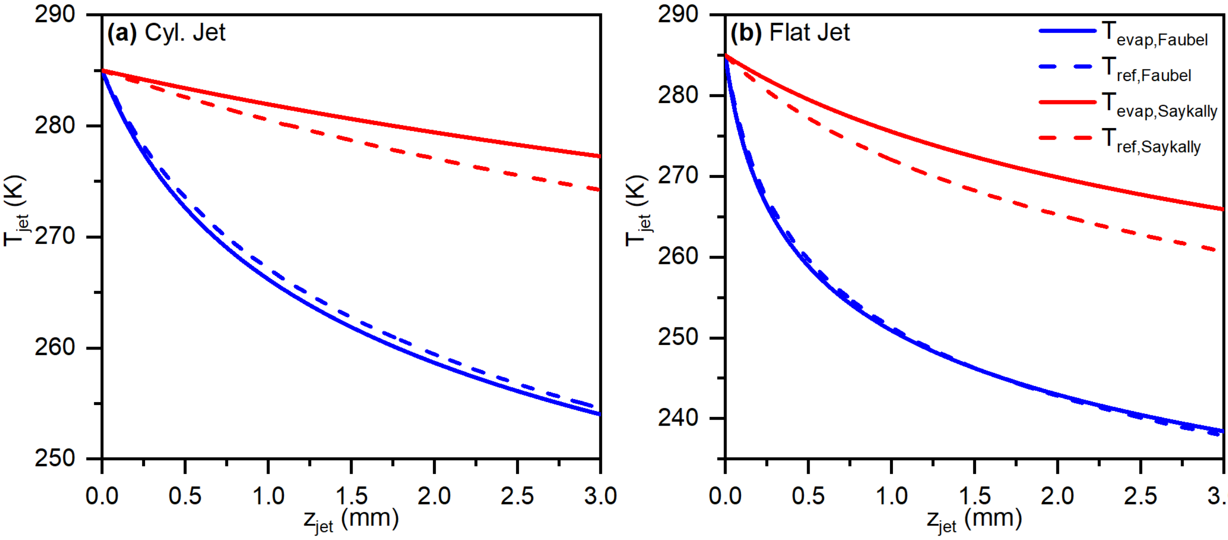


Figure 1.3: Water (a) cylindrical jet's (r_{jet} =13.8 µm, flow rate=$0.5\,\mathrm{mL\,min^{-1}}$, $v_{jet} = 13.9\,\mathrm{m\,s^{-1}}$) and (b) flat jet's (two cyl. jets with $r_{jet} = 27.6$ µm, flow rate=$2.5\,\mathrm{mL\,min^{-1}}$, $v_{jet} = 8.8\,\mathrm{m\,s^{-1}}$) temperature T_{jet} along the jet axis z_{jet}. Faubel's (blue curve) and Saykally's (red curve) are shown. dr/dt are estimated by evaporation theory (evap) and reference value (ref). In both estimations, parameters of liquid's enthalpy of vaporization ΔH_{vap} and heat capacity C_P are set to be constant at $2.27\,\mathrm{kJ\,mol^{-1}}$ and $4.20\,\mathrm{J\,g^{-1}\,K^{-1}}$, respectively. Jet temperature at the jet nozzle ($z_{jet} = 0$) is set to be 285 K.

Both T_{Faubel} and $T_{Saykally}$ are calculated by assuming ΔH_{vap} and C_P are constant, which the values are $2.27\,\mathrm{kJ\,mol^{-1}}$ and $4.20\,\mathrm{J\,g^{-1}\,K^{-1}}$, respectively. However, P_{vap} and ρ_{liq} vary with different temperatures, especially for P_{vap}. Therefore, it is important to set P_{vap} and ρ_{liq} to be functions with variable T_{jet}. A way to setup these functions is to use experimental values. Here we use $P_{vap}(T_{jet})$ and $\rho_{liq}(T_{jet})$ of supercooled water published from Sippola's group.[58] Supercooled water is chosen because water jets are often being evaporative cooled below 273 K under vacuum. ϕ_e in $T_{Saykally}$ is set to be 0.35 for cylindrical jets with r_{jet} <5 µm.[55]

Although this number may not be suitable for our 13.8 µm jet, it is still being used as a reference.

With all the parameters mentioned, the T_{jet} curves caused by evaporative cooling are now calculated numerically again by the Mathematica code shown in Appendix A.2. A Mathematica code to calculate water P_{vap} at different temperature using both Sippola's equation and Clausius-Clapeyron relation are also shown in the Appendix A.1 for reference. As the results shown in Figure 1.3, it is clear to see that Faubel's equation predicts a larger temperature gradient and therefore a lower jet temperature. The deviation not only could be coming from the inherent definition of dr/dt for both equations, but also could be that the empirical value of ϕ_e in Saykally's equation needs to be re-examined.

Besides using $(dr/dt)_{evap}$ based on evaporation theory with calculating values of P_{vap} and ρ_{liq} as a function of T, one can assume gas to be ideal and then use Clausius-Clapeyron (C.-C.) relation to extrapolate P_{vap} with reference point at T_{ref}. The radial ablation rate using reference value $(dr/dt)_{ref}$ can now be rewritten as

$$\left(\frac{dr}{dt}\right)_{ref} = \left(\frac{dr}{dt}\right)_{T_{ref}} \sqrt{\frac{T}{T_{ref}}} exp\left[-\frac{\Delta H_{vap}}{k_B T_{ref}} \frac{T_{ref}-T}{T}\right] \tag{1.7}$$

where $(dr/dt)_{T_{ref}}$ is the radial ablation rate at reference temperature T_{ref}. Plugging Equation 1.7 back into Equation 1.2, we get

$$\left(\frac{dT}{dz}\right)_{ref} = -2\left(\frac{dr}{dt}\right)_{T_{ref}} \sqrt{\frac{T}{T_{ref}}} exp\left[-\frac{\Delta H_{vap}}{k_B T_{ref}} \frac{T_{ref}-T}{T}\right] \frac{\Delta H_{vap}}{r_{jet} C_p} \frac{1}{v_{jet}}. \tag{1.8}$$

This new form of $(dT/dz)_{ref}$ requires obtaining value of $(dr/dt)_{T_{ref}}$ at T_{ref}. Two sets of values are provided, again, by the Faubel's group[40] and Saykally's group.[54] T_{ref} are set at 277 K, while $(dr/dt)_{T_{ref}}$ are 0.36 and 0.055 cm s^{-1}, respectively. The later one is said to be used for r_{jet} >5 µm. The calculated T_{jet} are again shown in Figure 1.3 with dash lines for Faubel's (blue) and Saykally's (red) parameters. These curves also have the same trend which Saykally predicts higher T_{jet} than Faubel. The deviation between two different P_{vap} estimating methods, however, is small, which indicates that it is fine to use ideal gas to predict water jet evaporating phenomenon. Estimated T_{jet} using different methods are listed in Table 1.1 at jet distance $z_{jet} = 1.5$ and 3 mm.

Besides cylindrical jet temperature, temperature of water droplets dripping under vacuum has also been studied.[17, 59, 60] Temperature of the flat jet, however, remains mostly undiscovered until 2022 by Yin's group.[61] They provide both experimental and simulated data for water and ethanol flat jets. One can search their paper for more detail.

A simple way to estimate the flat jet temperature is by adding a converting factor into the cylindrical jet equation. This factor is mainly based on the geometry differences between the jets. Considering using same liquid for both types of jets, which means ΔH_{vap} and C_p are the same, the rate of the evaporating cooling is than governed by the geometry of the jet, which is the surface area verses the volume of the liquid. This can be described in the

Jet Type	***dr/dt* Method**	**Group**	**T_{jet} at z_{jet} =1.5 mm (K)**	**T_{jet} at z_{jet} =3 mm (K)**
Cyl.	evap	Faubel	261.9	254.0
		Saykally	280.6	277.3
	ref	Faubel	262.7	254.6
		Saykally	278.7	274.2
Flat	evap	Faubel	246.2	238.4
		Saykally	272.4	265.9
	ref	Faubel	246.3	237.9
		Saykally	268.3	260.7

Table 1.1: Cylindrical and flat jets' T_{jet} predicted by Faubel's and Saykally's group. Results of different dr/dt predicting methods by evaporation theory (evap) and reference values using Clausius-Clapeyron relation (ref) are also listed. Jet conditions for cylindrical jet are r_{jet} =13.8 µm, flow rate=0.5 mL min^{-1} and v_{jet} = 13.9 m s^{-1}. While for flat jet are r_{jet} = 27.6 µm, flow rate=2.5 mL min^{-1} and v_{jet} = 8.8 m s^{-1}. T_{nozzle} is set to be 285 K for both types of jets.

differential point of view along the jet axis, z, and can be described as the ratio of perimeter and its cross-sectional area.[40] This ratio is called α and can be written as

$$\alpha = \frac{(Surface\ Area)}{(Volume)} = \frac{(Perimeter)dz}{(Cross - Sect.\ Area)dz} = \frac{(Perimeter)}{(Cross - Sect.\ Area)}. \tag{1.9}$$

Therefore, the converting factor between the cylindrical jet and flat jet can be simply written as the ratio of their α, which is defined as R_α and shown as

$$R_\alpha = \frac{\alpha_{flat}}{\alpha_{cyl}} = \frac{(Perimeter)_{flat}}{(Cross - Sect.\ Area)_{flat}} \frac{(Cross - Sect.\ Area)_{cyl}}{(Perimeter)_{cyl}}. \tag{1.10}$$

One should note that this is a simple conversion, which assume the flat jet has the same width along the z-axis and shape is no longer a leaf shape. In this dissertation, the width of the jet has been assumed to be the widest part of the jet, which is around 0.6 mm for running a water flat jet at 2.5 mL min^{-1} flow rate located at 0.5 mm below the tip of the nozzle. Therefore, the resulting temperature should be considered as a low temperature end. After multiplying this factor term R_α to the cylindrical jet temperature gradient in Equation 1.2, the flat jet temperatures can then be calculated by using both evaporation theory and reference data method to estimate T_{jet}. The results are shown in Figure 1.3 (b) and jet temperatures at 1.5 and 3 mm along the jet axis are listed in Table 1.1.

It is clear that with same starting T_{nozzle}, flat jet has much lower T_{jet}. Take z_{jet} =1.5 mm as an example, using both Faubel's and Saykally's methods predicts a 10–16 K lower jet temperature for flat jet. This is mainly due to the value of R_α being around 5, which means the flat jet has 5 times higher of cooling rate than the cylindrical jet with the same flow rate. This further shows great potential of the flat jet to explore liquid at lower temperature where cylindrical jet cannot reach.

Number of Collisions

Evaporating or scattered gases out from the jet surface need to pass through the vapor cloud surrounding the jet. These outgoing gases can collide with the vapor and then be perturbed, therefore, losing their initial properties. Thus, it is important to calculate the pressure gradient P_R and number of collisions N_{coll} along the axis of jet surface normal (R-axis). A way to estimate them can be done by using an electrostatic analogy.[53] For example, the cylindrical jet is often being simulated by a charged cylinder, while a flat jet is by a uniformly charged disk electrode. P_R can then be viewed analogous to the density of the electric field. As for N_{coll}, it is simply an integration of P_R starting from the liquid surface to the detector, which is analogous to electric potential.

With the electrostatic analogy, three types of jet can be estimated, the cylindrical jet with infinite length (cyl,inf), cylindrical jet with a certain length (cyl,L), and flat jet assumed to be a disc shape (flat). By setting the jet surface temperature T_z, P_z at position z along the jet axis is then calculated. The equations of P_R along R-axis for these different types of jets can be written as[40, 42, 53]

$$P_{cyl,inf} = \frac{P_z}{2}\frac{r_{jet}}{R}, \tag{1.11}$$

$$P_{cyl,L} = \frac{L_{jet} P_z r_{jet}}{2R\sqrt{{L_{jet}}^2 + R^2}}, \tag{1.12}$$

$$P_{flat} = \frac{P_z}{2}\left(1 - \frac{R}{\sqrt{R^2 + {r_{disc}}^2}}\right) \tag{1.13}$$

where r_{jet} and L_{jet} is the radius and the exposed length of a cylindrical jet. The 1/2 factors are shown in all the pressure equation because the vapor particle only contributes to the collision numbers if its flying direction is towards the liquid surface. If it is evaporating to the vacuum, it will be pumped away.[53] r_{disc} is the radius of a disc, which is used to estimate the flat jet geometry. Using the above equations, we can now try to calculate the pressure decreasing trends for different jets. Assuming the water jet has been evaporative cooled down to 265 K and the vapor pressure is 2.5 Torr, the results are shown in Figure 1.4. This is again calculated by using the Mathematica code shown in Appendix A.3. The r_{jet} and L_{jet} for the cylindrical jet has been set at 13.8 µm and 3 mm, which is again mimicking the cylindrical jet size of similar to the experimental condition. r_{disc} is around 0.37 mm, which should give the same area as a leaf shape flat jet with flow rate at 2.5 mL min^{-1}.

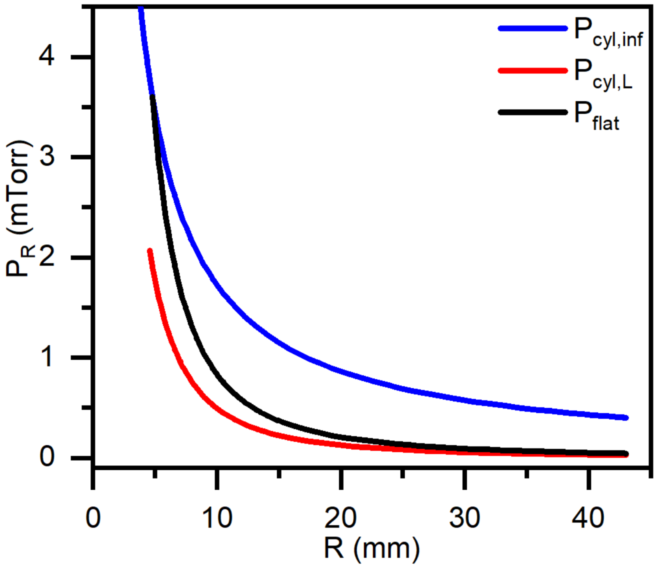


Figure 1.4: Pressure decreasing along the water jet surface normal. Three different types of jet are compared here. They are cylindrical jet with infinite length (blue curve), cylindrical jet with a certain length (red curve), and flat jet assumed to be a disc shape (black curve). The jet surface temperature is set at 265 K and its vapor pressure is 2.5 Torr.

In Figure 1.4, cylindrical jet with infinite length shows the worst pressure decreasing curve. This is easy to understand, since an infinite jet could contribute a lot more vapor pressure while the other two are finite in size. Therefore, it is more reasonable to compare cylindrical jet with finite length and flat jet. The infinite jet data is shown just for reference. Comparing $P_{cyl,L}$ and P_{flat}, Equation 1.12 predicts a better pressure decreasing for a finite length of cylindrical jet than Equation 1.13 for flat jet. For example, at $R = 43\,\mathrm{mm}$, which is the distance from the jet to the entrance of the detector, $P_{cyl,L}$ is 2.8×10^{-5} Torr while P_{flat} is 4.5×10^{-5} Torr.

After converting the P_z into molecular density n, one can calculate number of collisions by integrating the inversion of mean free path λ, which is defined as particle's flight length before it collides with the other particles. The equation can be written as[53]

$$N_{coll} = \int_{r_{jet}}^{R_{set}} \frac{dR}{\lambda(R)} = c\sigma \int_{r_{jet}}^{R_{set}} n(R)dR \tag{1.14}$$

where R_{set} is a set distance along the surface normal that one wish to observe, which is usually being set at the distance to the detector entrance $R_{set} = 43\,\mathrm{mm}$. $\lambda(R)$ is defined to be $(c\sigma n(R))^{-1}$, where $c = \langle v_{rel}/v \rangle$ is the ratio of particle's relative velocity v_{rel} and velocity v.[62] This parameter is set to be $3/4$ for evaporating solvent particles colliding with each other. σ is the cross-section of the outgoing particles with the surrounding particles. $n(R)$ is

Jet Type	$\boldsymbol{P_R}$ **(Torr)**	$\boldsymbol{\lambda}$ **(m)**	$\boldsymbol{N_{coll}}$ **(#)**	$\boldsymbol{Prob_{non-coll}}$ **(%)**
Cyl$_{inf}$	4.0×10^{-4}	0.3	1.1	32
Cyl$_{L}$	2.8×10^{-5}	4.4	0.8	43
Flat	4.5×10^{-5}	2.7	3.7	2

Table 1.2: Number of collisions for different type of water jets, including cylindrical jet with infinite length, cylindrical jet with finite length and flat jet, are listed. Parameters related to number of collisions, such as pressure P_R, mean free path λ and non-collision probability $Prob_{non-coll}$, are also included. The numbers listed are calculated at $R = 43\,\text{mm}$. Liquid surface temperature and its vapor pressure are set to be 265 K and 2.5 Torr. Cross-section between the water gas-phase particles is $30\,\text{Å}^2$. Jet radius and length for cylindrical jet are 13.8 µm and 3 mm. Radius of a circle disc, which is used to estimate flat jet, is 0.37 mm.

the number density of vapor particles at vacuum, which should be half of P_{vap} and are defined as Equation 1.11, 1.12, and 1.13. After applying the $n(R)$ equations into Equation 1.14, the number of collisions at R_{set} for different type of jets can be written as[40, 42, 53]

$$N_{cyl,inf} = \frac{r_{jet}}{\lambda} \ln \left[\frac{R_{set}}{r_{jet}} \right], \tag{1.15}$$

$$N_{cyl,L} = \frac{r_{jet}}{\lambda} \ln \left[\frac{R_{set}}{r_{jet}} \frac{L_{jet} + \sqrt{{r_{jet}}^2 + {L_{jet}}^2}}{L_{jet} + \sqrt{{R_{set}}^2 + {L_{jet}}^2}} \right], \tag{1.16}$$

$$N_{flat} = \frac{r_{dis}}{\lambda} \left(1 + \frac{R}{r_{disc}} - \sqrt{1 + \left(\frac{R}{r_{disc}} \right)^2} \right). \tag{1.17}$$

Using the above equations with the same parameters for different types of P_z mentioned above, the results of water jets are shown in Figure 1.5. σ is set to be 30 Å in this calculation.[53] The number of collisions is much higher for the flat jet than the cylindrical jet. Take $R_{set} = 43\,\text{mm}$ for example, $N_{cyl,L}$ is estimated to be 0.8, while N_{flat} is 3.7. We can then calculate the ratio of non-collision particles with total outgoing particles $R_{non-coll}$, in other words, the fraction of unperturbed evaporating molecules, by Beer's law $R_{non-coll} = \exp\left[-N_{coll}\right]$. The probabilities of observing non-collision particles $Prob_{non-coll}$, which is the percentage form of $R_{non-coll}$, can then be compared between cylindrical jet and flat jet, where the values are 42% and 2%. This large difference shows the drawback of using a flat surface. The pressure is much harder to be dissipated above the flat surface. Therefore, it is very hard to observe nascent outgoing particles using a flat liquid source, such as wetted wheel. Table 1.2 shows a summary of calculated parameters P_R, λ, N_{coll} and $Prob_{non-coll}$ at $R = 43\,\text{mm}$.

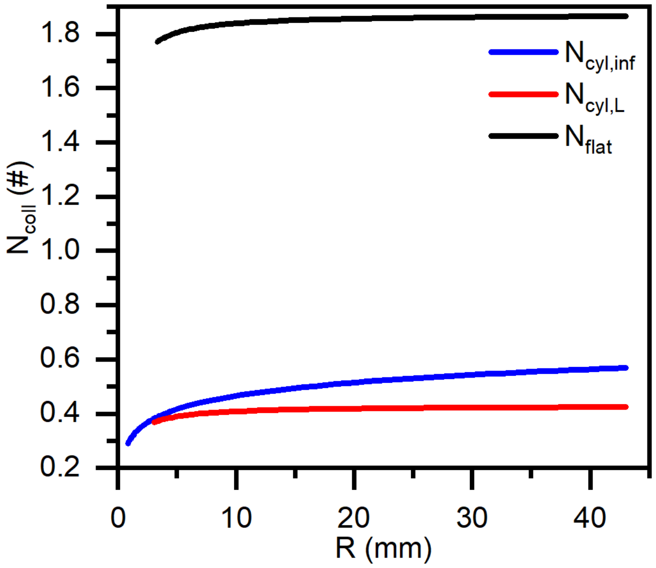


Figure 1.5: Number of collisions along the water jet surface normal. Three different types of jet are compared here, which are cylindrical jet with infinite length (blue curve), cylindrical jet with a certain length (red curve), and flat jet assumed to be a disc shape (black curve). The jet surface temperature is set at 265 K and its vapor pressure is 2.5 Torr.

Thoughts on Flat Jets

Due to the larger ratio of surface area to volume, flat jet can be used to investigate lower temperature of liquids that cylindrical jet cannot do so. Plus, the lower temperature further lowers the vapor pressure inside the vacuum chamber. To achieve to lower temperature, it would be best if one can generate a very thin flat jet in order to maximize the fraction of α. However, the thickness of flat jet generated by colliding two cylindrical jets has been reported to be around sub-micron.[48] Nanometer scale of thickness is not available unless using a gas dynamic jet, which uses gas to squeeze the liquid jet down to 20 nm.[63] This type of flat jet is apparently not suitable for scattering experiments because its continuously gas flow above the surface will perturb the outgoing particles.

Another way to lower down the liquid temperature is to use the lower stream of the jet sheet. The lower the sheet, the longer it has been exposed to the vacuum chamber. More evaporative cooling applies to these lower sheets. The sheet size, however, is smaller for lower sheet and thus scattering signal becomes smaller. One should also be aware of the shape of the sheet. Similar as cylindrical jets, the lower sheets will breakup and turn into a stream of jets at a certain length.[47] This distance depends on the flow rate and liquid itself. For example, a dodecane flat jet running at 3.5 mL min^{-1} starts spraying at its second sheet, while running at 2.5 mL min^{-1}, the jet sheet can be form to the third one.

The pressure decreasing gradient is intrinsically worse for flat jet due to its geometry. By using the electrostatic analogy to make pressure estimation, one should note that this formula requires knowing the cross-section of the outgoing particle. In the previous Subsection, σ for water has been set to be $30\,\text{Å}^2$ following the values reported by the Nathanson's group.[53] This value, estimated by hard sphere collision, might be underestimated due to neglecting the dipole moment between water molecules.[64] This is an issue to predict if the liquid system will work or not. Therefore, other ways to confirm the if the number of collisions affects the outgoing signal or not is required.

Nevertheless, the key point of the success for a flat jet is to balance between better cooling effect and slower pressure decreasing gradient due to its geometry. Besides using the above equations and codes, one might find a more sophisticated theory to do the estimation or try to elaborate the issues with several testing experiments. Chapter 3 shows a series of test with different experimental setup to verify if the detected signal is nascent or not. Whereas chapter 4 use the best setup verified in Chapter 3 then use neon-dodecane flat jet as a model system to experimentally justify if the vapor affects the outgoing particles or not.

Chapter 2

Experimental Setup

In this chapter, the apparatus for the liquid jet scattering experiment is described. It consists of five parts, the scattering apparatus, molecular beam generation, liquid flat jet, electronic setups and data collection method. The main purpose for this chapter is to provide detail on the scattering apparatus and introduce new setup.

2.1 The Crossed Beam Machine

Understanding how materials transform has always been a central topic for chemist, especially in the field of chemical kinetics and dynamics. Fundamental questions such as what reactants are included, how much energy is required for the reaction to occur and how does it dissipate, how long does the reaction takes, is the reaction stereo specific, and last but not least, what are the products, are key pursuits. To answer these questions, the crossed molecular beam, or crossed beam, technique has been developed.

A typical crossed beam machine includes two molecule sources the generate two molecular beams that will then collide under high vacuum. A detector viewing the collision intersection is dedicated to detecting the scattering signal and is able to give the angular distribution. The ability to observe single collision reaction make this technique especially useful to study elementary reactions with nascent signals.

History of Crossed Beam and B Machine

One of the early crossed beam setups was developed by Broadway[65] in 1933 for elastic scattering of alkali metal atoms, sodium, and potassium, in vapor of *trans*-di-iodoethylene (*trans*-$C_2H_2I_2$). Reactive scattering, however, was first successfully demonstrated 20 years later by Bull[66] and closely followed by Taylor and Datz.[67] During this era, reactions that could be studied were confined to alkali species due to the limited detection methods. In 1969, Lee and Herschbach overcame this limitation by incorporating an electron ionization detector into the crossed beam apparatus.[68] This allows universal detection and opens up a new field of chemical dynamics.

Nowadays, there are several crossed beam machines operating around the world. Although they are slightly different from each other depending on the detection methods and geometries, they all share similar arrangement from the very first Yuan Tseh Lee type crossed beam apparatus, which was built in Harvard around 1967.[68] This machine is called *Hope* or the "20 in machine" based on the size of the o-ring for the rotating chamber. After serving for 30 more years, *Hope* is now retired and serves as an exhibit to honor the legacy of crossed beams at National Science and Technology Museum at Kaohsiung in Taiwan since 2000.

Besides *Hope*, Lee also built two other crossed beam machines in his group, not including the other crossed beam setups he assisted in building. One is the so-called "B Machine" and the other is the "35 in Machine". The later one was built in Berkeley around 1979 and was mainly designed by Randal Sparks.[69] The 35 in Machine was built with much larger rotating chamber in order to have better energy and angular resolution. This machine is now located at Institute of Atomic and Molecular Sciences Academia Sinica at Taipei in Taiwan.

In this dissertation, all the experiments are performed on B Machine, which is sometimes called "Machine B" or the "25 in Machine". The B Machine shares generally the same design idea with *Hope* and only has slightly larger rotational chamber. It was mainly designed by Lee's graduate student, Cheuk-Yiu Ng, around 1975 when Lee moved to University of California at Berkeley. Most of its chambers were built in Lawrence Berkeley National Laboratory (LBNL) while the main chamber was built at University of Chicago. Having a rich traveling history, B Machine was first move from LBNL down to the C level of Giaugue Hall on the Berkeley campus in 1986. Later on, B Machine was given to Lee's former graduate student, Daniel Neumark, and was then moved to D10 Latimer Hall by Dan's graduate student, Jason Robinson.[70]

Since the birth of B Machine in the 1970s, it has been modified for different type of experiments. Research direction has been moved from gas-phase collisions, in which the setup is crossing two molecular beams, to photofragment translational spectroscopy (PTS) studies, which is crossing a molecular beam and a laser beam. After several trials and modification, B Machine is now being optimized for studying gas-liquid interfacial reactions, which on the setup side is crossing a molecular beam with a liquid flat jet. In the following Subsection, the details of the apparatus and the new parts are introduced.

Apparatus

Despite all the modifications that have been made, B Machine always consists of three chambers: the source, main and detector chambers. All of them are made from non-magnetic stainless-steel 304L. A general layout is shown in Figure 2.1 with the right wall of the main chamber removed. These three-dimensional drawings are re-drawn into both AutoCAD and Fusion360 from the original blueprints. These blueprints can be found both in the Neumark group and LBNL. Main chamber blueprint is provided by Prof. H. Floyd Davis in Cornell University.

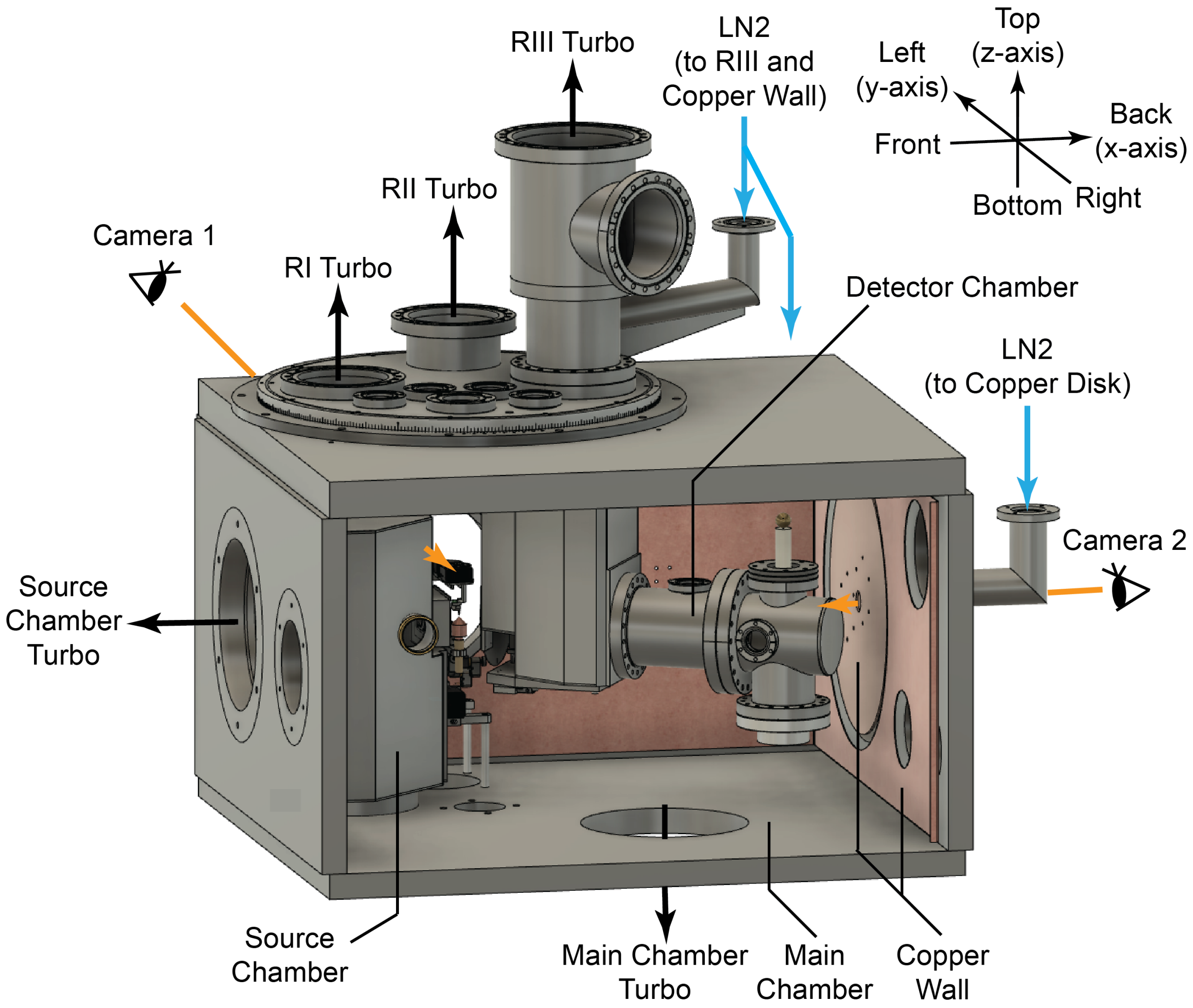


Figure 2.1: Layout of the liquid flat jet scattering setup. The axis system is shown on the top right. The wall on the right is removed in order to look into the chamber. The setup consists of three vacuum chambers, where the source chamber and the detector chambers are mounted on the front and top wall of the main chamber, respectively. The turbo pumping directions are shown in thick black arrows. Liquid nitrogen (light blue arrow) is fed into a double-wall chamber in RIII and acted as a cryo-pump. Two cameras are viewing at the x- and y-axis (orange arrows) in order to monitor the jet condition. Three chicken feeders, one of them located on the left of the wall is not shown, are used to fill LN_2 into the copper wall, copper disk and ionizer copper chamber inside RIII.

The source chamber is mounted on the front wall of the main chamber (see Figure 2.1). A pulsed valve is housed inside to generate a pulsed molecular beam. This supersonic beam is generated by a newly installed Amsterdam piezo valve (MassSpecpecD BV, ACPV2)[71] with proper temperature control system. It provides much shorter pulse width than the previous home-made pyrolysis source.[72, 73, 74] Further comparison between these two pulsed valves will be discussed in Section 2.2. The source chamber is pumped with a 2200 $\mathrm{L\,s^{-1}}$ turbo pump (Edwards Vacuum, STP-A2203C2) and backed with a 8.3 $\mathrm{L\,s^{-1}}$ mechanical pump (Welch, model DuolSeal 1397). A roots blower is installed between the turbo and mechanical pump to facilitate pumping capacity when the pulsed valve is on.

The main chamber is a large stainless-steel box with outer dimensions of $41.125 \times 42.5 \times 27\,\mathrm{in}^3$. The wall thicknesses are 1.625 in except for the top one (2.375 in), which is extra thick in order to hold the rotatable detector chamber and results in an internal volume of $0.56\,\mathrm{m}^3$. The chamber is so large mainly due to providing space for the detector chamber to be rotated inside. The detector chamber is attached to the main chamber from the top sealed by two 25 in spring-loaded gaskets. A rotary vane pump (Edwards Vacuum, E2M18) with pumping speed around 5.7 $\mathrm{L\,s^{-1}}$ is pumping the region between these two gaskets, so-called Tec-ring, to maintain main chamber's vacuum when rotating the detector. Typical pressure in the tec-ring region is around mid-10^{-1} Torr.

Plenty of ports are located around the main chamber for possible future extension and application. Most of the ports are sealed with a Viton o-ring with chemically inert vacuum grease (Apiezon, Apiezon L). For example, two glass windows are mounted on the left and the back wall of the main chamber. Two cameras (Basler, DMK 72BUC02) with adjustable focusing lenses are newly installed to look through these windows and monitor the liquid jet condition at the reaction center. Glass windows are selected instead of acrylic windows to reduce image shifting on the camera. Further details of the liquid jet holder, which is also located inside the main chamber, and its controlling program can be found in Section 2.3 and Appendix B.1. A pressure monitoring LabVIEW program is also written in order to continuously collecting pressures data of the chamber. The code is shown in Appendix B.2.

The same turbo and mechanical pump models as in the source chamber are used to pump down the main chamber. However, performing a liquid jet experiment requires much larger pumping speed in order to overcome the high gas load from the jet. A large liquid nitrogen cooled copper shield, consisting of a L-shaped copper wall and a circle copper disk, has been designed and installed inside the main chamber working as a cryogenic pump. The pumping speed S can be calculated by[75, 76]

$$S = \frac{\alpha A_c \overline{v}_g}{4} \tag{2.1}$$

where α is the sticking coefficient, sometimes called condensation coefficient, and depends on the temperature of the cryogenic surface, T_c. A_c is the area of the cryogenic surface. $\overline{v}_g = (8RT_g/\pi m)^{1/2}$ is the mean velocity of the gas molecules, which depends on gas temperature, T_g, and molecular weight of the gas, m. To estimate the pumping speed of our copper

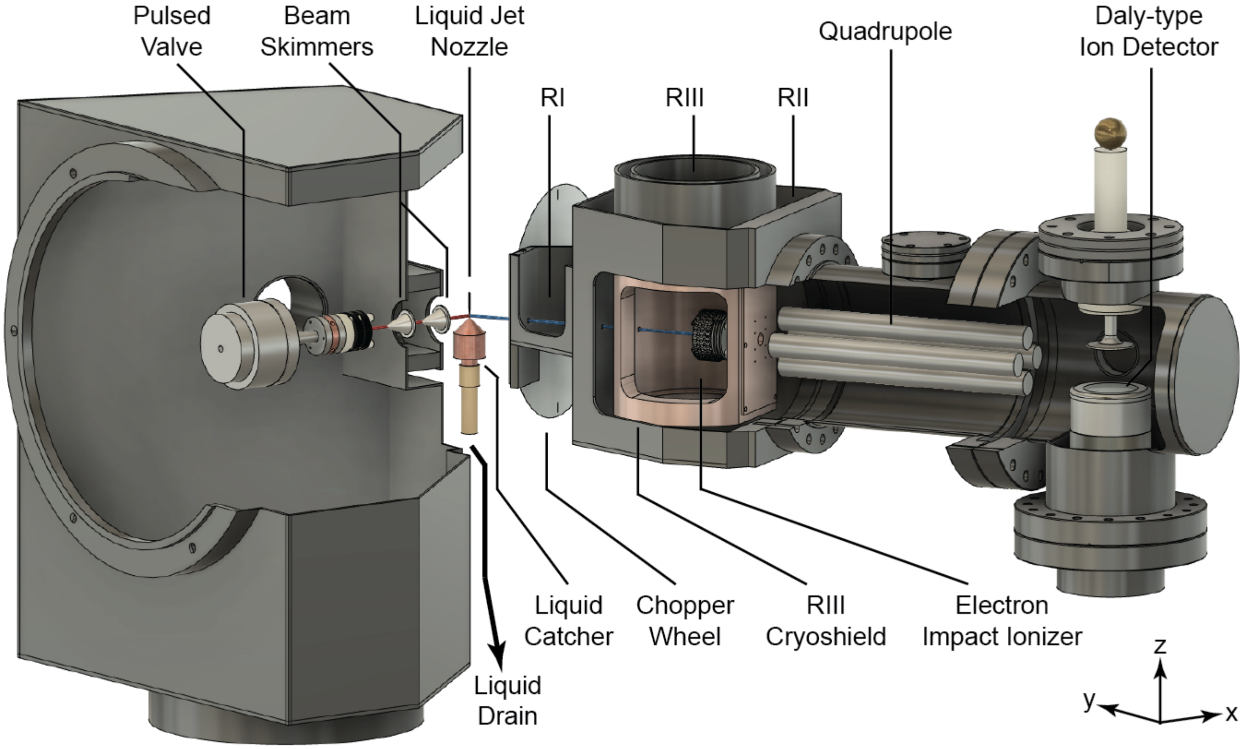


Figure 2.2: Schematic of the detail setup inside the vacuum chambers. Although the main chamber is removed, all the parts shown in the figure are under vacuum. Both source and detector chambers are partially cut for readers to see the electronic arrangement inside the chambers. A red line starting from the pulsed valve represent the molecular beam direction, while a blue line represents the scattering particles.

shield, we assume T_c around 150 K with liquid nitrogen cooling and T_g at room temperature 300 K. α is approximated to unity, which suitable for molecules staying in liquid phase at room temperature. A_c is approximated to be $1\,\mathrm{m}^2$. With these numbers, we estimate the copper shield's equivalent pumping speed for dodecane is $48\,000\,\mathrm{L\,s}^{-1}$ and for water is $148\,000\,\mathrm{L\,s}^{-1}$. The pressure reduction due to this extra cryo-shield is huge. It brings the main chamber pressure from mid-10^{-5} to mid-10^{-6} Torr.

Finally, the detector chamber is located inside the main chamber. A clearer view showing the components inside is shown in Figure 2.2. Details about the arrangement can be found elsewhere[68] and are briefly mentioned here. The detector is isolated from the main chamber through a home-made gate valve, which can slide from sealed mode to a 100 µm circular aperture (small hole) or a $3 \times 3\,\mathrm{mm}^2$ square aperture (big hole). The detector comprises three differentially pumped regions, region I, II and III (RI, RII and RIII). RI is the first differential pumping chamber and helps lower the incoming gas flux. RII connects with RI by a $3 \times 3\,\mathrm{mm}^2$ square aperture aligned with the detector inlet. RIII sits inside RII separated by a $3 \times 3\,\mathrm{mm}^2$ square aperture again. It has a liquid nitrogen cooled copper cage with an axial ionizer using Thoriated Iridium filament (Extrel, Pittsburgh).[77, 78, 74] This cryogenic cooled cage is crucial in order to have lower background generated by ionizing the stray gas and pursues better a signal-to-noise ratio (S/N).

When the particles get ionized by the ionizer, they exit RIII through a 0.4 in circular aperture and arrive in RII again. A quadrupole with rod diameter 3/4 in is used to perform mass selection. The quadrupole is controlled by either a 1.2 or 2.1 MHz oscillator with 300 W power (Extrel) depending on the target ion mass range and ideal mass resolution. For reference, the 1.2 MHz oscillator gives worse mass resolution (±0.5 amu), but larger mass range (1–500 amu), while the 2.1 MHz one is better mass resolved (±0.25 amu) but only reaches to (120 amu).

After mass selection, ions are detected by a Daly-type ion detector.[79] To further describe it, cations first strike to a $-30\,\mathrm{kV}$ metal surface, which is commonly called the "doorknob" due to its shape. Secondary electrons are released and strike to a scintillator. Photons are then generated by the scintillator and detected by a photon multiplier tube (PMT) (Burle, defunct, model 8850), which includes a 12-stage dynode. A typical experimental PMT voltage used in this thesis is set to be 1.55 kV. Signal from the PMT is collected by a constant-fraction discriminator (Ortec, model 584) to eliminate noise with low intensity. Besides not using the constant fraction mode, this discriminator is very sensitive that the discriminating threshold can be set as low as $-5\,\mathrm{mV}$. A typical threshold is set at 25 mV. There is no high voltage protection for the input of this discriminator, which means if PMT signals are too strong, it could burn it. This in fact has happened three times in 2021 in January, March, and April due to mis-handling the liquid jet and causing high vapor flux pours into the detector. Therefore, one should always keep an eye on the jet condition when collecting data.

The discriminated signals are then be counted by both MS controller (Extrel, model 5221) and an MCS-pci card (Ortec), where MCS stands for multichannel scaler. The Extrel controller provides mass spectra and is controlled by Extrel Merlin automation software. The MCS card, however, allows us to count the input signal with dwell times as short as 100 ns. Dwell times are set at 0.5 and 1 µs for molecular beam velocity measurements and time-of-flight (TOF) spectra collection, respectively. A LabVIEW control program for the MCS card was developed by Mark Shapero.[80]

The angular range of the detector is restricted from $-5°$ to $90°$ where $0°$ is the molecular beam axis. This is due to the spatial restriction of chopper wheel with the jet holder at negative angles and the detector chamber with the source chamber more positive angles. One can reach a larger scanning range from 90° to 115° by modifying the source chamber. The larger angular range, however, requires modifying the main chamber, which might not be desired for the time being.

Each differential pumping region in the detector is pumped by a 400 or $420\,\mathrm{L\,s^{-1}}$ turbo pump (a Leybold GmbH Turbovac 361 and two Seiko Seiki STP-400 respectively). All the turbos are backed by a $400\,\mathrm{L\,s^{-1}}$ Turbovac 360 foreline turbo pump (Leybold GmbH) and a mechanical pump (Welch, DuolSeal 1397). The extra foreline turbo not only increases the compression ratio for better pumping speed, but also serves as a protecting device in case the mechanical pump fails. Unfortunately, this did happen in January in 2018, which the detector mechanical pump failed and caused the RII and RIII turbos to fail. The foreline and RI turbos, or the Leybold turbos, somehow kept spinning. This further helps prevent the oil vapor in the mechanical pump back flow into the detector, even though there are molecular sieves inserted before the inlets of all our mechanical pumps.

A 2-slit chopper wheel is located in front of the detector entrance. Its purpose is to measure the molecular beam velocity and to set time zero *to* for both evaporation and scattering experiments. The entire chopper wheel setup, including its water-cooling system, is mounted at the bottom of the rotating detector chamber, so that the wheel can chop the incoming signal from the jet at any detector angle. The chopper can be moved along the axis perpendicular to the detector axis. This allows us to choose to keep the chopper on or off axis. Its opening time is shown in Figure 4.1 and its temporal resolution will be discussed. Its timing related to other electronics will be shown more in detail in Section 2.4.

2.2 Molecular Beam Generation

What makes the crossed-beam method being so powerful is not only the development of vacuum and detector technologies, but also the huge improvement of molecular beam generation.[81, 82] In the later Subsections, a history of molecular beam is introduced. Then, the testing information of the newly installed Amsterdam valve will be shown.

History of Molecular Beams

A brief history of molecular beam technique starts in 1911 where Dunoyer first demonstrated molecular rays of alkali atoms,[83] an early prototype of a molecular beam. Quickly followed by Stern in the 1920s,[84, 85] effusive beams generated by a heated oven were systematically studied and bloomed in the famous Stern-Gerlach experiment.[86]

An effusive beam,[87] sometimes called an oven beam, has been successfully assisted gas-phase research. Its velocity distribution is same as a Maxwell-Boltzmann distribution, which can be written as[88, 89]

$$N_{eff.\ beam}(v) = N_{MB,flux}(v) \propto v^3 \exp\left[-\frac{mv^2}{2RT_0}\right] \tag{2.2}$$

where $N_{eff.\ beam}(v)$ is the number density of gas molecule in an effusive beam with speed v and mass m. R is the gas constant and T_0 is the temperature of the oven or gas reservoir. It is worth to mentioning that this v^3 distribution is the $flux$ distribution of Maxwell-Boltzmann distribution. It is different from the traditional form of Maxwell-Boltzmann distribution, which is often called the *volume*-form and has v^2 instead of v^3 as a factor. The idea of v^3-expression was pointed out by Einstein in 1920.[85] He explained that the v^2-expression

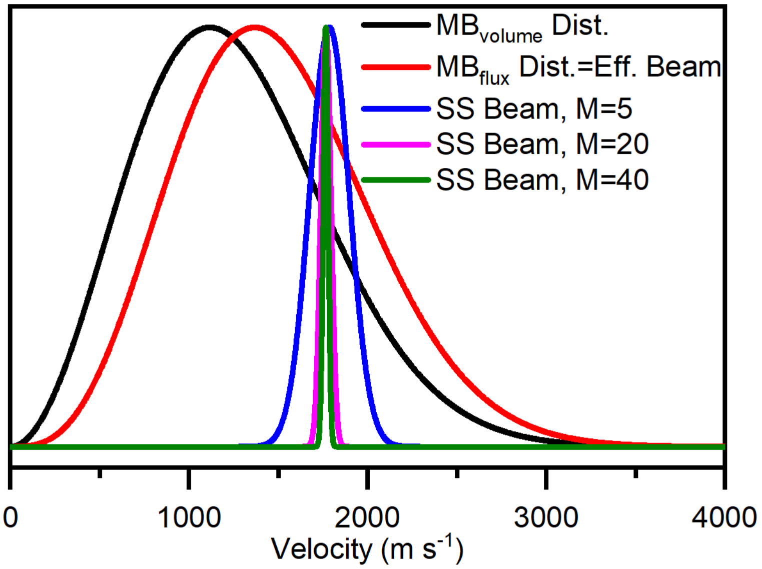


Figure 2.3: Helium beam velocity distribution. *volume*-form of Maxwell-Boltzmann distribution (black line) and effusive beam (red line) at temperature 300 K are calculated and shown. Supersonic distributions are also shown with Mach numbers at 5 (blue line), 20 (magenta line) and 40 (green line) with flow velocity equals to 1765 m s^{-1}.

claimed by Stern is not applicable to molecular beam with a Knudsen source orifice. Details of this criticism can be found in Ramsey's[90] and Comsa's[10] papers. Comparison between the *volume*-form (v^2-expression) and the *flux*-form (v^3-expression) can be found in Figure 2.3.

Now let's focus back to the velocity distribution of the effusive beam. $N_{eff.\ beam}(v)$ shows a broad distribution and requires beam velocity selector to narrow down.[91, 92, 93] The beam intensity is also limited by the gas effusion rate through the oven orifice. A rule of thumb is that when λ_0, the mean free path of the gas molecules in the reservoir, is far smaller than the diameter of the reservoir orifice, D, the beam intensity will be dramatically reduced due to many collisions during the expansion process.[89]

In 1951, Grey proposed a new type of beam called a nozzle beam that provides a much narrower beam and became a game changer.[88] A cone-shape orifice was acting as a beam outlet, and a conical nozzle is placed downstream of the beam to extract the central part of the beam in order to provide free molecular flow. The expansion of the gas flow can therefore be treated as an adiabatic and isentropic expansion, where the enthalpy. $H(T_0)$ can be transferred to kinetic energy of the beam with flow velocity u. The ultimate flow velocity, u_{max}, is defined as[89, 94]

$$u_{max} = \sqrt{\frac{2H(T_0)}{m}} = \sqrt{\frac{2C_pT_0}{m}} \tag{2.3}$$

where C_p is the heat capacity of the gas molecule. Equation 2.3 is essentially from energy conservation $mu_{max}^2/2 = H(T_0)$. A useful practice is to calculate u_{max} for monoatomic beam of helium, neon and argon, for which the results should be 1765, 786 and $559\,\mathrm{m\,s^{-1}}$ at $T_0 = 300\,\mathrm{K}$ with $C_p = 5k_B/2$ for monatomic gas. The velocity of the free jet, or more often called supersonic source (SS), is now being defined as[89]

$$N_{SS\ beam}(v) \propto v^3 \exp\left[-\frac{m(v-u(x))^2}{2RT(x)}\right] \tag{2.4}$$

where $u(x)$ and $T(x)$ is the local flow velocity and temperature of the beam during expansion along the center of the beam axis at distance x. A comparison between the effusive and supersonic beam is shown in Figure 2.3. The supersonic beam distributions are shown with different Mach numbers M, which is defined as

$$M = \frac{u}{a} = \frac{u}{\sqrt{\frac{\gamma RT}{m}}} \tag{2.5}$$

where a is speed of sound and γ is heat capacity ratio C_P/C_V. Common numbers for γ are 5/3 for ideal monoatomic gas and 7/5 for ideal diatomic gas.

Up to this point, the mentioned molecular beams are continuous beams. High pumping speed is required for vacuum environments to ensure free expansion. Therefore, shortly after the idea of supersonic beam, researchers put efforts on developing pulsed beam with microsecond pulsed width in the 1960s through the 1980s.[95, 96, 97, 98, 99]

Nowadays, the technique of pulsed valves is pretty mature, and the pulse width has been shortened down to several microseconds. Several common pulsed valves are listed in Table 2.1 for reference, such as General valve,[100] Even-Lavie valve[101] and Amsterdam valve.[71] A home-made pyrolysis valve,[72] which also use a piezo crystal from Physik Instrumente to actuate the valve, was installed and developed by former student Scott Goncher in this project.[102, 103] Its purpose was to generate radicals and to study their photodissociation dynamics. An Amsterdam valve was newly installed in B Machine in year 2021 for liquid jet scattering purpose due to its shorter pulse width. Comparison between the pyrolysis source and Amsterdam valve will be introduced in the later Subsection.

Amsterdam Valve

An Amsterdam piezo valve ACPV2 (MassSpecpecD BV, Enschede) with nozzle diameter 500 µm and its electronic driver EDU5-200V were recently installed on B Machine. Although this valve is great for producing strong beams with tens of microseconds pulse width, there are some limitations using it. First, the gas compounds need to be compatible with the valve o-ring to prevent leaking from the corroded o-ring. This issue, however, can be solved by changing the nozzle o-ring. Suggested by Steve Saric and Erin Sullivan,[104] graduate students in the fast beam project in the Neumark group, the o-ring inside the valve is

Valve Name	**Actuating Mechanism**	**Pulse Width (µs)**	**Repetition Rate (Hz)**	**Temperature Range (K)**
General Valve	solenoid	300	0–250	269–378
Even-Lavie Valve	solenoid	25	0–600	10–500
Amsterdam Valve	piezo	<20	0–5000	300

Table 2.1: Comparison between several types of common pulsed valve, which includes General valve, Even-Lavie valve and the Amsterdam valve.

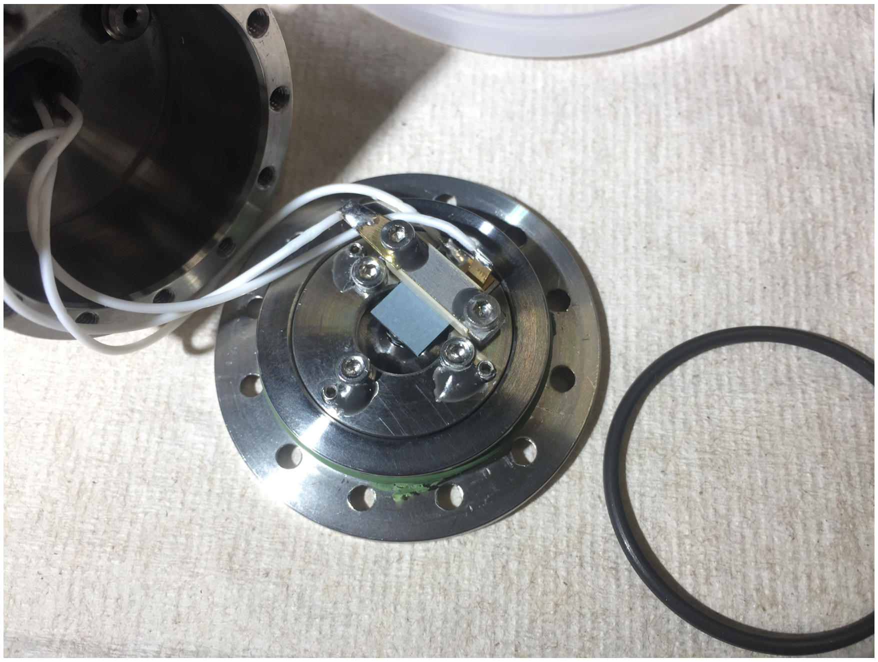


Figure 2.4: Inside the Amsterdam valve. The piezo crystal (blue rectangle piece) is located around the center of this figure and connected with the wires. Between the crystal and the outlet of the valve, a Kalrez o-ring is used to seal in between. The green o-ring used to seal the valve body is broken and have been swapped to a Viton o-ring. One can change the o-ring to other materials, such as Kalrez, based on their chemical compatibility of the carrying gas.

specifically changed into a KALREZ o-ring to be chemically resistant to possible corrosive gas sample in the future. If the gas molecules deposit inside the valve, one can also clean it by removing the faceplate and sonicating the entire valve in ethanol.[104]

Secondly, the valve is not recommended to be heated up. This is specifically mentioned in the manufacturer's manual,[105] even though the piezo crystal PZT507 was rated for maximum operating temperature at 80 °C. This could be due to the extra heating inside the

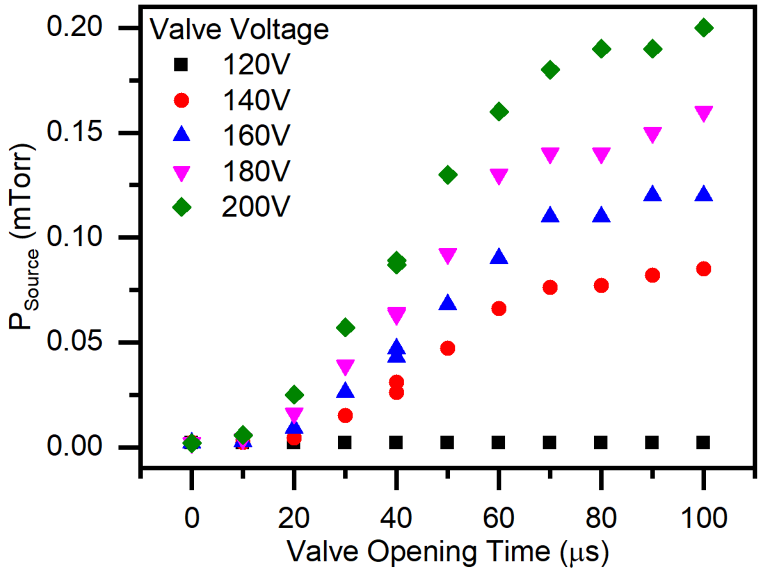


Figure 2.5: Leak test of a properly sealed Amsterdam vale. The valve is pumped in the source chamber and backed with 4 bar pure neon. Different valve settings of voltage and opening time are tested.

valve when pulsing the valve which leads to an even higher temperature at the piezo crystal. In this thesis, the valve has been constantly cooled by the building cooling water system at the temperature of 12 °C. A further sophisticated valve temperature control holder was installed in July of 2022 and the valve's temperature vs. valve operation was tested. It is observed that the valve does not pulse at valve temperature above 43 °C. Therefore, a heating temperature below 35 °C is suggested, which is again consistent with the company's warning.

Last but not least, the valve requires careful tuning of the position of the piezo crystal when the crystal is replaced. This process is done by tuning the three set screws inside the valve (see Figure 2.4) after unscrewing the lock screws. Minor adjustment around 1/8 turns for one set screw at once is suggested. The goal of the adjusting process is to make sure the valve is sealing properly.

The valve company provides a systematic tuning process using a leak detector (Leybold, Phoenix XL300) to test the valve. In the XBeam project, however, a more informal way is used due to absence of the leak detector. The leak test is done by immersing the valve into ethanol then observing the bubble formation rate when the valve is backed with 4 bar of neon. A desired leak rate is that the valve is partially sealed when there's no voltage applied, fully sealed when applied with 140 V and strongly pulsed with expelling sprayed ethanol at the orifice. For example, the bubble forming rate of 20 s per bubble at 0 V and 50 s per bubble at 140 V is considered a fine valve condition. For a well-sealed condition, which is also tuned by the company, the valve is not leaking either applying voltage on or

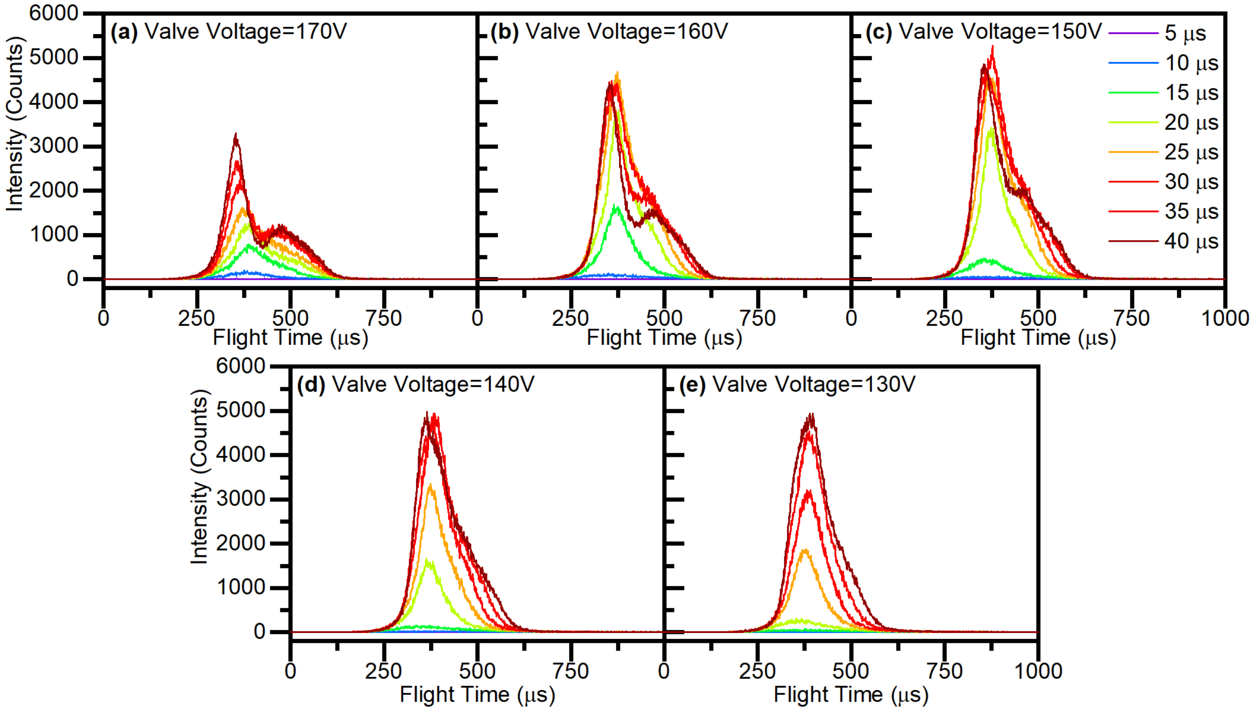


Figure 2.6: Beam profile of the Amsterdam valve at different valve settings. Valve open time varying from 5 to 40 µs are tested with valve voltage at (a) 170 V, (b) 160 V, (c) 150 V, (d) 140 V and (e) 130 V. The valve is backed with 4 bar pure neon. All the data are taken with 5000 sweeps on the same day and therefore the intensities are comparable.

off. A pressure test for this valve condition is recorded by attaching the valve to the isolated source chamber and pumped by its 2200 L s^{-1} turbo and pressure measured by an ion gauge. The pressure results are shown in Figure 2.5 for reference.

Besides checking the leak rate of the valve, it is also crucial to find the best valve controlling conditions for generating a proper beam profile, which is important when doing data analysis. By changing the valve's operating voltage and opening time, the best valve condition can be found, which needs to simultaneously fulfill the symmetric temporal beam shape and sufficient beam intensity. Figure 2.6 shows testing of the valve performance running at 200 Hz and backed with 4 bar pure neon. Bimodal peaks are appeared at high valve voltage which suggests that the valve is not opened correctly. While lowering the voltage, the bimodal peak merges back into one and a single symmetric peak appears. It is noted that this kind of testing should be done with different gases as the gas samples can have different heat capacities and provide different cooling conditions inside the valve. For example, typical valve settings are valve voltage at 160 V with 10 µs opening time for argon and 130 V and 30 µs for neon.

Finally, with the fine-tuned Amsterdam valve, we can compare the beam profile of the

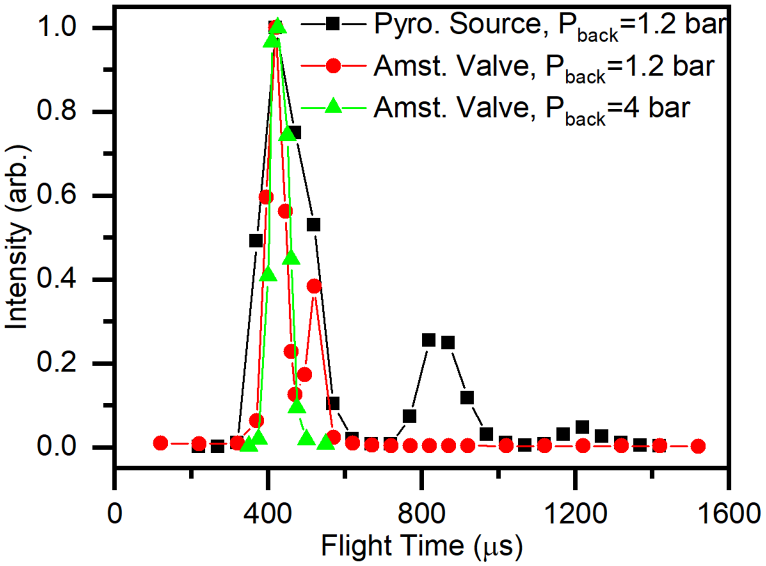


Figure 2.7: Pure helium molecular beam profiles of the pyrolysis source and the Amsterdam valve. The intensity is normalized for easier comparison.

pyrolysis source and the Amsterdam valve using pure helium. The beam profile is measured with the chopper wheel in the molecular beam axis and provides an opening time of 15 μs for the collected TOF. The plot of the maximum value in each TOF with varying valve delay times can be used as the beam profile. The results are shown in Figure 2.7. Damping is clearly seen in the pyrolysis source where the second and third pulses are shown at 870 and 1220 μs. This also affects the source chamber pressure, which is 3×10^{-5} Torr when the valve is running.

As for the Amsterdam valve with the same backing pressure at 1.2 bar and opening time at 100 μs, the source chamber pressure is 2×10^{-5} Torr. If we only look at the first pulse, the pyrolysis source also shows a broader pulse duration with full width at half maximum (FWHM) around 150 μs. While for a well-tuned Amsterdam valve, FWHM is smaller than 50 μs. The tiny damping occurs in the Amsterdam valve at 520 μs can be eliminated by increasing the backing pressure to 4 bar shown as the green curve. The exact intensity differences between the two valves, unfortunately, is not applicable due to the different detector ionizing settings. Nevertheless, the Amsterdam valve often requires lower ionizer emission current, which suggests a larger beam intensity in the Amsterdam valve. All the properties mentioned above suggest that the Amsterdam valve generates a better molecular beam profile and is recommended to use for scattering experiments.

2.3 Liquid Jet Operation

Throughout this dissertation, both cylindrical and flat jets are used. The entire liquid line setup is briefly described here, and the schematic is shown in Figure 2.8. The liquid line used in this experiment is 1/16 in PEEK line with total length around 5 m. Liquid samples are stored in a Duran glass amber bottle to prevent any photochemical reaction. The screw caps with tubing connection system (Duran, #1129814) are specifically selected in order to directly use the liquid sample after degassing without venting the bottle. To perform this, three different fluid lines are required to attach to the bottle. One is for sparging gas into the liquid where a sparger is located at the tip of this gas line and immersed into the liquid. Another is a vacuum line for pumping down the liquid. The other is the input liquid line to the HPLC pump with a 2 µm glass filter to prevent clogging inside the pump.

Degassing is generally performed using helium due in its low solubility to the liquid and non-interesting mass range. Liquids are first filtered through a glass filter with 2 µm porous then poured into the bottle. After sealing and isolating the bottle, the vacuum line is opened, and the bottle is pumped down until pressure reaches 20 Torr. Then the vacuum line is closed, and helium is sparged into the liquid until the bottle pressure goes up to 850 Torr. This pump-down and pressure-up process is repeated five times before the liquid is fed into the HPLC pump.

The HPLC pump is a G7110B isocratic pump from Agilent. This model only has a single inlet and therefore swapping the liquid sample requires taking out the liquid filter and purging the pump head thoroughly. An HPLC pump is chosen over the syringe pump due to the larger liquid volume and the ability to refill it during experiment. This is particularly important for a flat jet owing to its much higher flow rate than cylindrical jet.

The downside of HPLC pump, however, is that the pulsing condition occurs in the liquid line generated by the two pistons in the pump head. Although there's a built-in damping system in the pump, the jet "breathing" phenomenon is still observable. To eliminate this pulsing condition happens, a 35 mL flexible stainless-steel ballast reservoir (Swagelok, #SS-FM4SL4SL4-75CM) acting as a pulse damper is installed sideways in the liquid line.[106] Following Dan DePonte's suggestion, the damper is filled with the sample liquid with around 1 cm of air left unfilled. This air section acts as an air damper and would theoretically solve the pulsing issue more efficiently.

Before the liquid flows through the heat exchanger, it passes through an inline filter with 2 µm PEEK frits. This filter is used to prevent any particles from reaching the chip nozzle, which will be described in detail in the following Subsection and causes blocking. 0.5 µm PEEK frits can also be used, though the HPLC pumping pressure will be slightly higher. The heat exchanger is a double-wall tubing that passes through the main chamber wall. It is temperature controlled by a chiller (Neslab, RTE-7) that can lower the liquid temperature from 25 °C to −5 °C for a dodecane flat jet. The cooling power of the heat exchanger is limited by the length of the double-wall tubing and the cooling capacity of the chiller, where temperature at −25 °C is reported. For lower liquid temperatures, it is easier to use the cryogenic cooling wall to cool the jet, especially for salty water.

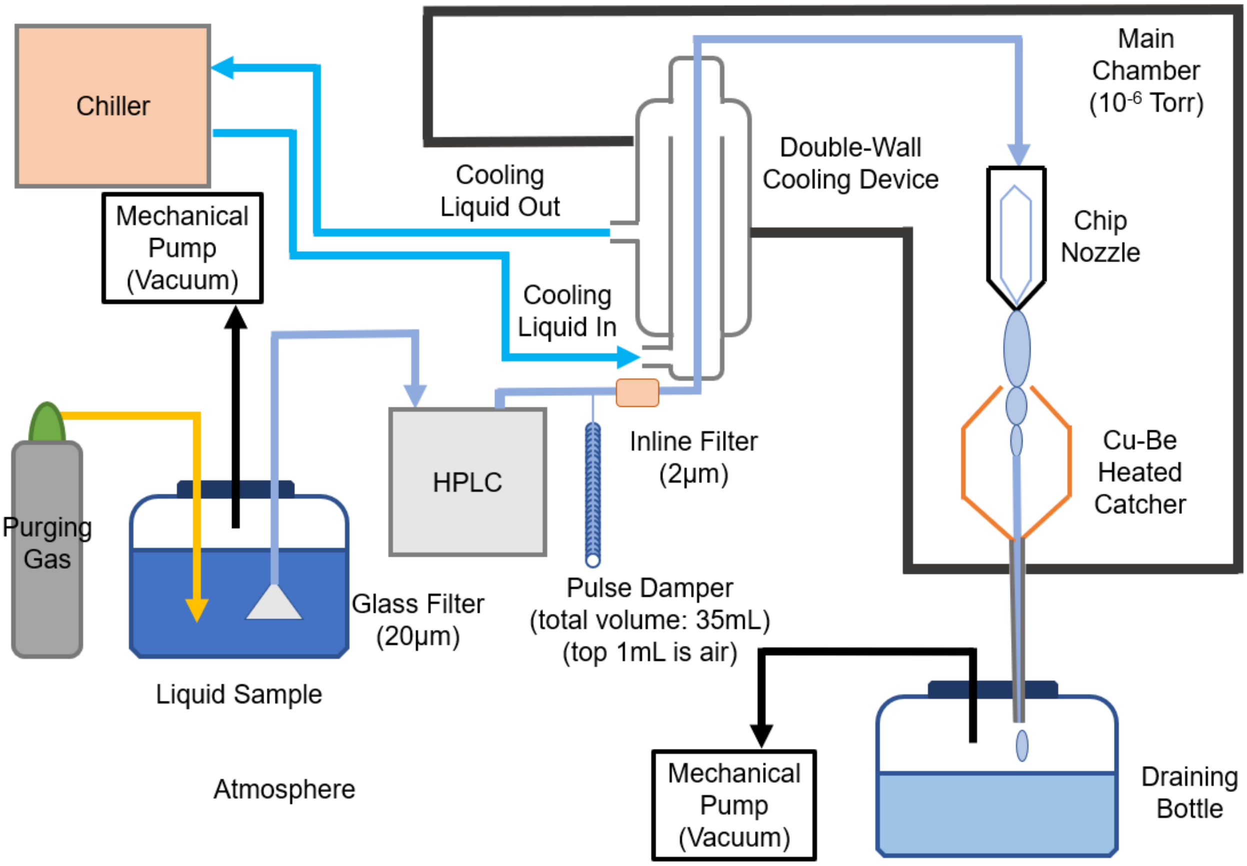


Figure 2.8: Schematic of the the liquid line system. The liquid samples are first being prepared in a degassing system. Then be pumped by a HPLC pump followed by double-wall cooling device passing through the main chamber wall. The liquid is then pushed through the chip nozzle and a flat jet forms. The wasting liquid from the jet is collected by a catcher system. Liquid is collected in a draining bottle pumped by a mechanical pump. Liquid sample are recycled and re-filtered before reuse.

After temperature control, the liquid flows through the chip nozzle interface then generated a flat sheet through a commercially available microfluidic nozzle. The wasted liquid is caught and disposed of by a liquid catcher. This entire jet holder system will be described in detail in a later Subsection, as is the chip nozzle.

Flat Liquid Jet Generation

There are several ways to generate flat jets that can be characterized into three groups. The first method is pushing the liquid through a tapered rectangle slit-shaped orifice. The resulting liquid will therefore be squeezed into a flat sheet. This method has been used in some dye laser system (Radiant Dyes Chemie, model RDSN06) in order to perform windowless dye pumping.[32] A more sophisticated, tens-of-micrometer-sized slit nozzle has been developed in the recent years.[107, 108] Our collaborator, Bernd Abel at the University of Leipzig, has developed a sapphire slit nozzle with slit dimension of $50 \times 10\,\mu\mathrm{m}^2$ in 2018. With flow rate at $2.5\,\mathrm{mL\,min^{-1}}$ and backing pressure around 20 bar, a water flat jet is generated with sheet size of $0.5 \times 1.5\,\mathrm{mm}^2$. In the end we did not use this type of slit nozzle, unfortunately, due to clogging and alignment issues.

The second type of flat jet is the colliding jet. Essentially, the flat sheet is generated by colliding two cylindrical jets.[47, 48] Second sheets and so on can be formed perpendicularly after the first sheet, but their sheet sizes are much smaller and therefore are usually not used. By changing the collision angle of the two jets, one can change the shape and the size of the liquid sheet. Operating a colliding jet requires great stability of the liquid flow. If one jet is too strong or is clogged, it will cause liquid spraying inside the vacuum chamber. This will send out a huge pressure spike and could damage the electronics in the detector. In order to control the jet precisely, its jet holder is often designed in a way that the jet angle can be changed freely either by picomotors or other manual mechanism. The flow rate of each cylindrical jet can also be tuned freely. All these issues make the development of colliding jets time consuming, and the holder is usually bulkier.

The final type of flat jets are gas dynamic jets. The idea is instead of crossing two liquid jets, two gas beams are shot onto the liquid jet symmetrically and therefore compress the jet into a flat sheet. The main advantage of using such type of jet is that the liquid sheet thickness can be as low as nanometer range, which is suitable for transmission spectroscopy for infrared, X-ray and even electrons. Koralek and DePonte developed a microfluidic chip nozzle (Micronit, sheet nozzle) to generate ultrathin liquid sheets with reported thickness below 20 nm,[63] whereas both the slit nozzle and colliding jet can only go down to tens of µm. However, this mode of operation is unsuitable for scattering experiments.

This microfluidic nozzle is shown in Figure 2.9. It consists of three channels, where the liquid is fed through the lower inlet and goes through the central channel, while the gas goes through the outer two channels. This type of gas dynamic jet, however, is not suitable for molecular beam scattering experiments. Because the gas flow squeezing the liquid will perturb any evaporated or scattered gas particles from the jet and ruin their nascent properties.

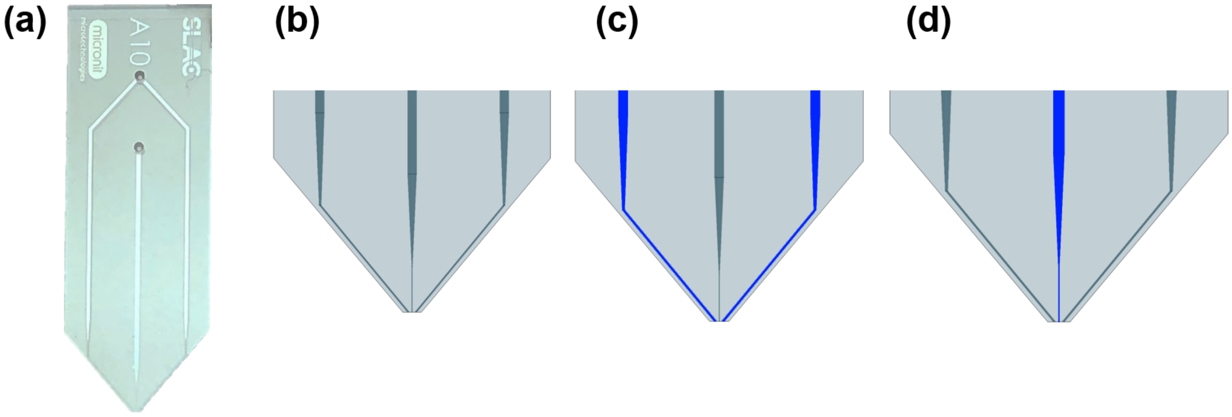


Figure 2.9: (a) Photo of the chip nozzle. (b) Zoom-in view at the tip of the nozzle. Chip nozzle running at (c) colliding jet and (d) cylindrical jet mode with blue channels representing the liquid flow.

Eventually, our group decided to use the microfluidic chip nozzle, but operates it in the colliding jet mode. How this is performed is that the liquid is flowing through the two outer channels. With a fixed colliding angle 80°, the resulting flat jet is acting similar to a colliding jet with two 60 µm diameter cylindrical jets, since the outlets are rectangular orifices with size $50 \times 60\,\mu\text{m}^2$. Although there's no ability to change the colliding angle and orifice size from this commercial nozzle, the machining precision is so nice that all the chips produce the same size of jets under the same HPLC pump condition, and therefore there is no need to worry about misalignment as long as no particles clogging the channels.

Another plus for using this chip nozzle is that it can be easily swapped to the cylindrical jet mode by flowing liquid through the center channel. The rectangular orifice of $20 \times 30\,\mu\text{m}^2$ generates a cylindrical jet with 28 µm diameter. It is also important to block the two outer channels to prevent liquid climbing up due to the capillary action. This is also true for the colliding jet mode, where the center channel needs to be blocked.

To operate the chip nozzle, a nozzle interface is required. Neptune Fluid Flow Systems provides a compact custom-made nozzle interface made by PEEK for this chip nozzle. Since no bio-sensitive samples are used in the current project, we used a home-made stainless-steel nozzle interface instead. The stainless-steel interface was designed by Sebastian Malerz in Bernd Winter's group at Fritz Haber Institute (FHI) in Berlin, Germany, and manufactured in the machine shop there. In the design shown in Figure 2.10, IDEX's fittings (Super Flangeless Fittings, M-653x) with o-rings(Viton, AS568-001) are used to seal both the liquid inlet and the chip interface. Since the nozzle interface is only sealed by o-rings, it is recommended that the pressure of the HPLC pump does not exceed 70 bar.

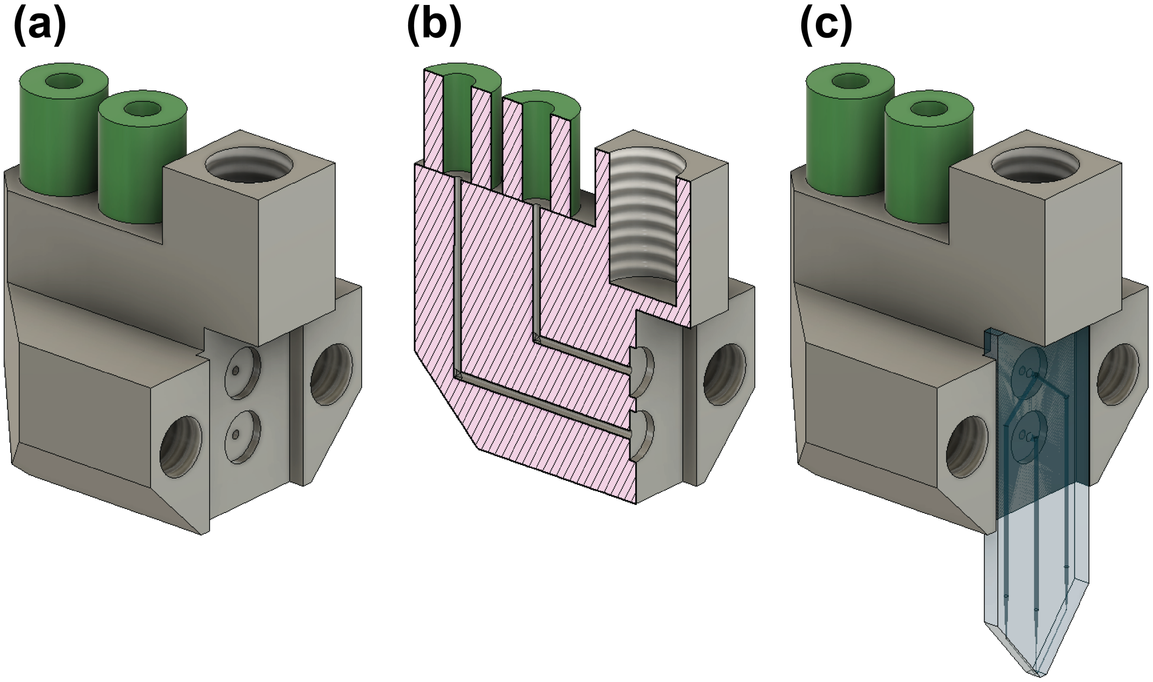


Figure 2.10: 3D drawings of nozzle interface using Fusion 360. The part is shown as (a) the interface only, (b) its sectional view, and (c) combining with the chip nozzle where the clamper to hold the nozzle is not shown for clarity. Two o-ring grooves are designed to fit with size AS568-001 o-rings in order to seal the nozzle.

Jet Holder System

Operating a liquid jet under vacuum requires sophisticated position control of the jet. This is particularly important for scattering experiments because the jet needs to be well aligned at the reaction center. The holder system is designed by Marvin Pohl and manufactured in the machine shop in chemistry department at University of California at Berkeley. A 3D model of the jet holder system is shown in Figure 2.11.

The holder is designed to be a seven-axis system, which includes three axes (x, y, and z) of freedom for the catcher relative to the jet, three axes for the jet and catcher relative to the molecular beam, and the rotational axis for the jet. To be more specific, first, the relative position between the jet and the catcher needs to be carefully adjusted to be located at the node between the sheets. In this dissertation, the catcher is always located at the first node which is shown in Figure 2.12. Doing so dramatically reduces the exposed liquid inside vacuum, prevents backflow from the draining bottle, and therefore lowers the chamber pressure.[109] This type of catcher system brings the waste liquid out of the vacuum chamber immediately without worrying about exceeding the liquid storage volume. This method is particularly suitable for flat jets since their flow rates are much higher. To obtain this

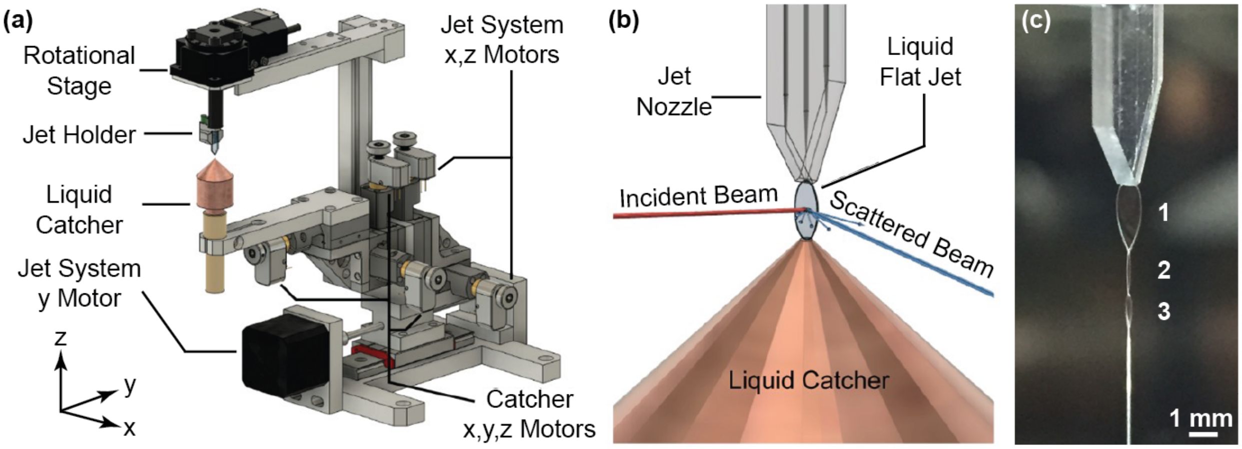


Figure 2.11: (a) Total setup of the jet holder. (b) Closer look at the jet region. The red line represents the impinging molecular beam. After hitting the jet surface, the scattered gas can shoot to many directions shown as blue lines. (c) Photo of a dodecane jet running at atmosphere. The flow rate is $2.5\,\mathrm{mL\,min^{-1}}$.

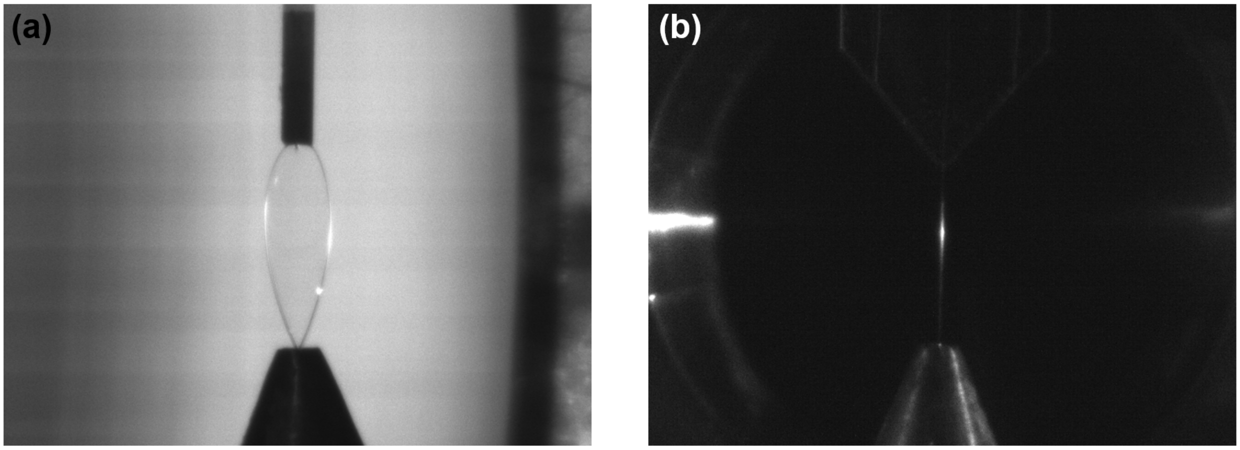


Figure 2.12: (a) Camera view of the jet on y-axis (perpendicular to molecular beam axis). (b) Camera view of the jet on x-axis (the molecular beam axis).

mechanism, 3 pairs of translational stages with picomotors (Newports, 8301NF) are used.

Secondly, the jet plus catcher system needs to be positioned at the reaction center, so that the molecular beam can correctly shoot onto the jet surface and the detector can receive the scattering signal starting from the rotational center. Two of the axes (x and z in Figure 2.11) are moved using the same picomotors as used for catcher alignment. The y axis is moved by a step motor with translational stage (Schneeberger, MNNL 24-110) having 11 cm of rail length. This much larger range of movement is designed in order to move the jet in and out of the molecular beam axis, so that the beam can be directly shot into the detector in order to measure the beam velocity.

Finally, a rotation stage is installed to rotate the jet and change the molecular beam incident angle. A piezoelectric miniature rotation stage (Physik Instrumente, Q-614) was first installed. Due to the high tension from the PEEK liquid line, this type of rotation stages suffer from low torch and the jet angle is limited. After we swapped it to a step motor-driven rotation stage (Standa, 8MR174(E)-11-20), this issue was resolved.

2.4 Electronic and Machine Parameters

Based on the basic translational energy equation $E = mv^2/2$, one can calculate particle's translational energy by knowing its mass, m, and velocity, v. The mass is measured by the quadruple mass spectrometer. The velocity, on the other hand, requires careful calibration of particle's flight time, t, and flight distance. In this thesis, we use time-of-flight (TOF) distributions to characterize the properties of the outgoing particles, where t is the x-axis in the figure instead of velocity. The main reason is for simplicity because the detector also reports flight times. In the following Subsections, the method to extract particle's neutral flight time and flight distance are shown. Then, the electronic triggering and delay systems that are set based on the flight time are then described.

Flight Time Determination

The outgoing neutral particle's velocity, v, is defined as $v = NFL/t$, where NFL is the neutral flight length and t is the neutral flight time. This t shows the properties of the measured particles and is then written as t_{true} in the later equations. However, the detector does not measure t_{true} directly. Instead, the measured flight time, t_{meas}, is a combination of t_{true}, ion flight times and several different time offsets. Although the actual t_{meas} varies from the machine geometry and electronics settings, it can be written in a general equation as

$$t_{meas} = t_{true} + IFT + E_o \pm M_o \tag{2.6}$$

where IFT is the ion flight time in the detector. E_o and M_o are the electronic and mechanical time offsets. We first discuss E_o. It is related to a photodiode (TT Electronics, OPB960N55) located on the chopper system to monitor rotating speed of the chopper. In our setup, the diode is located at angle 180° before the chopper slit reaches the detector entrance. When the slit passes through the diode, it will generate a pulse and it is used as a triggering source for electronic devices in this setup. There is a propagation delay time between the diode's receptor to receive the light and to form an electronic output. This time delay is rated as 5 µs on the manual and is the major contribution for E_o. Since this kind of electronic offset is always a delay phenomenon, E_o should always be a negative number in Equation 2.6. One should note that the definition of E_o mentioned above is slightly different from Douglas

Krajnovich's thesis.[110] Here, we follow the E_o description in Neil Cole-Filipiak's thesis[74] because the chopper setup are similar. $E_o = 6\,\mu s$ has been used in this thesis besides 5 µs in order to compensate any other possible electronic delays in our system.

M_o, the mechanical offset, is used to compensate the geometric displacement between the photodiode and the detector aperture. Ideally, the diode is placed exactly 180° before the aperture, which means the center of the diode, the center of the chopper wheel and the center of the detector entrance are on the same line. Time delay between a slit passing through the diode and then reaching the aperture is accordingly to be exactly 2500 µs when using a 200 Hz 2-slit chopper. However, this delay time is not exact in practical due to the mechanical displacement when installing the chopper. One can measure this displacement, which is M_o, by running a molecular beam and measure the delay time difference of the beam when the chopper is spinning counterclockwise (ccw) and clockwise (cw). In the case of chopper spinning ccw, which is also the condition performed in this thesis, M_o is defined as

$$M_{o,ccw} = \frac{VDT_{ccw} - VDT_{cw}}{2} \tag{2.7}$$

where VDT states for pulsed valve delay time. VDT_{ccw} and VDT_{cw} are VDT for chopper rotating ccw and cw. The sign for M_o in Equation 2.6 is plus for ccw and minus for cw condition. Here is an example of how we determine M_o. A pure helium beam is generated by the pulsed valve with certain VDT, which gives the highest intensity on the TOF spectrum. Tuning VDT with time steps of 10 µs, VDT_{ccw} and VDT_{cw} are found to be 1010 µs and 970 µs. VDT_{ccw} is larger than VDT_{cw} means that the pulsed valve needs to wait longer time before it fires, or in other words, the chopper needs to rotate more degrees for the slit to align with the detector aperture. Using Equation 2.7, we found that $M_o = 20\,\mu s$. A more precise value can be extracted if we align the peak positions in both ccw and cw TOF spectra. M_o is then found to be 19.5 µs.

To check the chopper offset with the detector, which is also a great practice, we convert M_o into a chopper angle. This can be easily calculated, and the offset angle is equal to $180° \times 20\,\mu s/2500\,\mu s = 1.4°$. Therefore, the actual angle difference between the diode and detector entrance is $180 - 1.4 = 178.6°$. One should note that the M_o measurement needs to be done after centering the MCS delay time with the photodiode pulse. This is done by an extra delaying device, which is named as MCS Alignment in Figure 2.14 in the later Subsection.

The second term ion flight time, IFT, is the flight time of the ionized ion starting from the ionizer, flying through the quadrupole, then finally reaching the doornob. IFT can be written in an equation of $IFT = \alpha_{ion}\sqrt{m}$, where α_{ion} is the ion flight constant and m is molecular mass. This equation is based on our detector acting as a Wiley-McLaren TOF mass spectrometer, where all the ions with same charges receive same potential energy from the electric field that is then transferred into kinetic energy.[111]

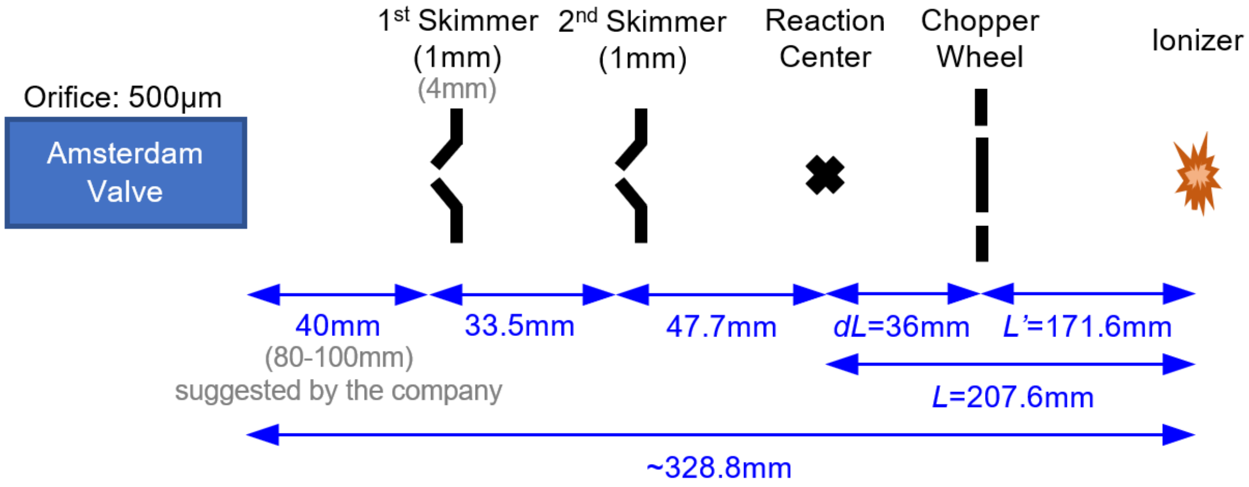


Figure 2.13: Schematic of flight length in the current setup and has been used in this entire thesis, where neutral flight length for evaporation and scattering experiments are $L' = 17.16\,\mathrm{cm}$ and $L = 20.76\,\mathrm{cm}$. One should recalculate the numbers if the chopper is entirely dismounted. Numbers written in gray are suggested dimensions by the Amsterdam valve company to increase beam intensity.

α_{ion} is determined by measuring the arrival times of different masses of ionized fragments in a molecular beam. The beam molecule is chosen to undergo dissociative ionization in the ionizer and the masses of its daughter ions span through the mass range that will be used in the experiments. For example, anisole has fragments with masses 15, 65, 78, 93 and 108 and used in this setup. The α_{ion} is measured to be $4.8\,\mu\mathrm{s}\,\mathrm{amu}^{1/2}$.

Neutral Flight Length

Since there are two types of experimental arrangement in this dissertation, when the chopper is in and out the detector axis, two neutral flight length NFL are used. One is NFL starting from the chopper to the ionizer, we call it L', and the other one is from the reaction center to the ionizer, L. A schematic is shown in Figure 2.13 to show the differences of L' and L based on the apparatus arrangement.

Let's first discuss L'. Its value is obtained by measuring the arrival time of the molecular beam using different noble gases with the chopper wheel chopping the beam. Assuming the gas pulses are fully supersonic expanded, their beam velocities can be defined as in Equation 2.3 with C_p equals to $5R/2$ for monoatomic ideal gas. Equation 2.6 can then be re-written as

$$
\begin{aligned}
t_{meas} &= t_{true} + IFT + E_o \pm M_o \\
&= \frac{NFL}{\sqrt{\frac{5RT_o}{m}}} + \alpha_{ion}\sqrt{m} + E_o \pm M_o \\
&= (\frac{L'}{\sqrt{5RT_o}} + \alpha_{ion})\sqrt{m} + E_o \pm M_o.
\end{aligned}
\tag{2.8}
$$

In practical, noble gases, such as helium, neon, argon and krypton, are used to extract L' from Equation 2.8. By plotting the arrival time of each molecular beam at its maximum intensity, which is t_{meas} in this case, as a function of square root of mass, $\sqrt{m}$, the data shows a nearly linear relation. By using a linear regression, the slope can then be used to obtain L' and we measure that $L' = 17.16\,\text{cm}$. This L' is used for evaporation and beam velocity measurement because the chopper wheel is used in both cases.

As for scattering experiment, the NFL is now defined to be the distance from scattering origin to the detector. This means that the time zero is now located at the reaction center[1]. The distance between the chopper and reaction center, dL, is measured to be 3.6 cm by a ruler. Therefore, L is $L = L' + dL$ and is equal to 20.76 cm.

Electronic Triggering and Delay Time

Based on the time offsets and required triggering times mentioned above, a timing diagram for the electronics are shown in Figure 2.14. To set up the time delays, we first use an oscilloscope (Tektronix, TDS-3032) to monitor the photodioide output, which shows a 15 µs square pulse with repetition rate at 400 Hz. Triggering for the electronics starts when the photodiode generates a pulse. This 400 Hz signal will then pass through a rate divider and output square pulses with the same shape as the input, but with lower rate at 200 Hz. The reason for this is to perform beam-on and beam-off subtraction for scattering experiments.

After rate dividing, this triggering signal is then fed into several digital delay generators (Stanford Research, DG535) to set up delay times for different electronics. First, a delay time is required to align the MCS start time to the center of the chopper slit. The timing for this delay generator is set as 1250 µs plus an extra time delay xx, which is determined by the diode and MCS triggering signal on the oscilloscope. Then, this new triggering signal is used to trigger delay time generator for MCS and pulsed valve. For evaporation experiment, MCS delay time is set to be 1250 µs and therefore added up to a 2500 µs delay for the slit to reaches the detector. This same MCS setting is also used for beam velocity measurement. Time zero, t_o, for both experiments are set at the location of chopper wheel.

[1] Time zero for liquid jet scattering experiment should be located at the jet surface. Besides using this location, we use the rotational center of the detector as time zero for simplicity. Since the jet thickness for both cylindrical and flat jets are around 10 µm, it is fine to neglect jet thickness from NFL, which is at the scale of tens of centimeters.

Exp. **Mode**	**Number of** **Ch. (#)**	**Subt.** **Mode**	**Start Ch.** **for Subt. (#)**	**Dwell** **(µs)**	**NFL** **(cm)**	**MCS Delay** **Time (µs)**
Evap.	5000	Unsubt.	N/A	1	17.16	1250
Beam Vel.	1000	Unsubt.	N/A	0.5	17.16	1250
Scat.	5000	Subt.	2500	1	20.76	1250-yy

Table 2.2: Common parameters used in DataAcq.vi for different types of experiments. E_o, M_o and α_{ion} are set to be $-6\,\mu s$, $19.5\,\mu s$ and $4.8\,\mu s\,amu^{1/2}$ throughout this thesis. yy is the used to compensate the time for beam to fly through dL.

As for the scattering experiment, the chopper is moved away so that the detector always collects the full scattering profile. t_o is now set at the reaction center and the MCS delay time needs an extra parameter, yy, to compensate this extra distance dL. yy is defined to be dL/u_{beam} where dL is measured to be 3.6 cm and u_{beam} is the beam velocity measured using Equation 2.9. Delay time for the pulsed valve is set for the most intense beam to pass through the chopper when chopper is aligned with the detector.

2.5 Data Collection and Analysis

Two types of raw data can be measured by this setup. One is the mass spectrum and the other is the time-of-flight spectrum. The mass spectra are taken using the mass spectrometry system from Extrel with its own Merlin software. The TOF data, on the other hand, are collected through a home-built LabVIEW program, DataAcq.vi, written by Mark Shapero.[80] Details of how the program works have been described in Mark's thesis, so only a brief introduction is given here. The program essentially consists of two parts, TOF collection and the beam velocity measurement. For the later part, different interpretations are made compared to Mark's and previous theses. Therefore, a Subsection is written to make statement of this deviation. New machine parameters are also used in DataAcq.vi, since the chopper position has been changed from the front of the second skimmer to the front of the detector. These parameters are summarized in Table 2.2.

Since liquid jet scattering experiments are newly performed on this project, data analysis for evaporation and scattering experiments were recently built. The codes are written in Python 3.0 and shown in Appendix C. Details of the TOF analysis procedure will be described in Chapter 4, while a brief description is described in the later Subsection. The angular distributions can also be obtained by integrating TOF data at different detection angles through the same analysis codes.

Besides evaporation and scattering experiment, a special type of experiment to extract residence time of the scattered molecule in the liquid can also be carried by this setup. The chopper needs to be used and located at the detector axis, while t_o is still defined at reaction center. This special case of how to extract residence time requires understanding

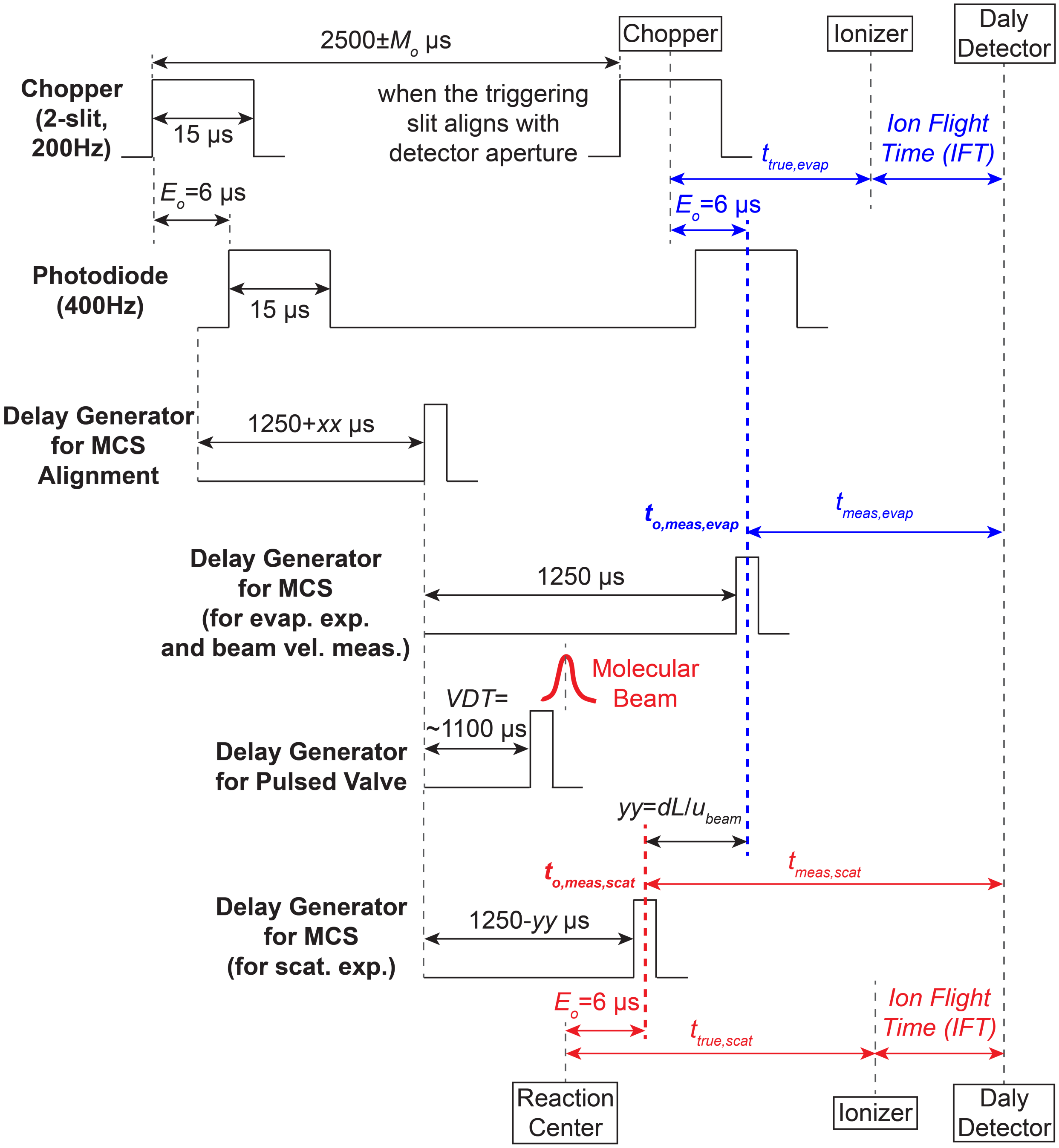


Figure 2.14: Timing diagram for different types of experiments. Blue notations show time settings related to evaporation experiment and beam velocity measurement. While for scattering experiment, the timings are colored in red.

the desorbing process of gas from the liquid surface. Bradley Ringeisen, a former student in the Nathanson's group at University of Wisconsin-Madison, provides great introduction on how to do so in his thesis.[112] A preliminary discussion for performing such experiment is shown in Chapter 5.

Beam Velocity Measurement

The molecular beam velocity characterization is done by using a LabVIEW program written by Mark Shapero.[80] The interpretation of the fitting function that will be described below is slightly different from Mark's and previous theses, which are basically following Douglas Krajnovich's derivation.[110] In their molecular beam expression, the velocity distribution is expressed as *volume*-form, which is same as Equation 2.4 but with v^3 changes into v^2. Here, I decided to follow Moorse's and Comsa's statement and use the $flux$-form, which has v^3 and should be the same as Equation 2.4. Since the expression of $N_{SS\ beam}(v)$ is a theoretical result. This format is commonly rewritten from Equation 2.4 into

$$N_{SS\ beam}(v) \propto v^3 \exp\left[-\frac{m(v-u)^2}{2RT}\right] = v^3 \exp\left[-s^2(\frac{v-u_{beam}}{u_{beam}})^2\right] \tag{2.9}$$

where u_{beam} and T are the beam velocity and beam temperature. s is so-called the speed ratio, which is essentially a ratio of the beam velocity and the thermal spread of the beam.[94] With this new molecular beam velocity expression, the speed ratio is commonly used to represent how well the beam is velocity-compressed. For example, a narrower velocity beam shows larger s and larger Mach number M, where M can be transferred from s by using the definition of Mach number in Equation 2.5. The relation is shown below for reference

$$s = \frac{u_{beam}}{\sqrt{\frac{2RT}{m}}} = \frac{u_{beam}}{\sqrt{\frac{\gamma RT}{m}}}\sqrt{\frac{\gamma}{2}} = M\sqrt{\frac{\gamma}{2}}. \tag{2.10}$$

Now let's focus back to $N_{SS\ beam}(v)$. Since this Equation 2.9 is the theoretical expression, we need to convert it to what is seen by the detector. First, the detector is an electron impact ionizer and has different ionization efficiency, k, to different speed of particles. k is inversely proportional to v and therefore a factor of $1/v$ needs to be considered.[77] Secondly, the detector measures the flight time instead of directly measuring the velocity. Thus, a Jacobian transformation is required to transform from velocity space to time space. Since $N(v)dv = N(t)dt$, the time space equation $N(t)$ requires a transforming factor of dv/dt. It is defined as

$$\frac{dv}{dt} = \frac{d}{dt}\left(\frac{NFL}{t}\right) = -\frac{NFL}{t^2} \propto \frac{1}{t^2}. \tag{2.11}$$

Therefore, the transformation factor is $1/t^2$ or v^2. Now, combining the two factors mentioned above, we can now derive the flight time distribution of the beam being measured by the detector as

$$\begin{aligned} N_{SS\ beam,meas}(t) &= N_{SS\ beam,meas}(v) \times \frac{dv}{dt} \\ &= N_{SS\ beam}(v) \times k \times \frac{dv}{dt} \\ &\propto v^3 \exp\left[-s^2(\frac{v-u}{u})^2\right] \times \frac{1}{v} \times v^2 \\ &\propto v^4 \exp\left[-s^2(\frac{\frac{L'}{t}-u}{u})^2\right]. \end{aligned} \tag{2.12}$$

In conclusion, Equation 2.12 is used to fit the beam velocity and speed ratio from the detector signal. Although the equation form is same as written in the previous data acquisition program, it is important to go through the derivation step-by-step.

Tim-of-Flight Spectra

TOF spectra are taken for both evaporation and scattering experiments. Both require transferring t_{meas} into t_{true} as shown in Equation 2.6. If one wishes to convert TOF spectra into velocity spectra, it is important to mention that the neutral flight length, NFL, are different for evaporation and scattering experiment, where NFL should be L' and L, respectively. This is mainly because time zero for both experiments are different. One can review Figure 2.13 in Section 2.4 for more information.

Two types of equations are often used to fit the TOF spectra. One is the Maxwell-Boltzmann distribution and the other is the supersonic distribution. The equations detected by the detector can be written as

$$N_{MB,meas}(t) \propto \frac{1}{t^4} \exp\left[-\frac{m(\frac{NFL}{t})^2}{2RT}\right] \tag{2.13}$$

and

$$N_{SS,meas}(t) \propto \frac{1}{t^4} \exp\left[-\frac{m(\frac{NFL}{t}-u)^2}{2RT}\right]. \tag{2.14}$$

where Equation 2.13 is basically re-written from Equation 2.2 after considering transforming factors mentioned in the previous Subsection. Equation 2.14 is same as Equation 2.12, but with v converted into t. Details on the actual numbers for the parameters and which equation should be use for fitting will be specified and discussed in each different experiment and shown in the analysis code in Appendix C.

Angular Distribution

Angular distributions are obtained through TOF spectra at different outgoing angles. Fittings curves of the TOF distribution, which are usually either using Equation 2.13 or 2.14, are used instead of raw TOF data in order to prevent high fluctuation at flight time closer to 0. Each data point on the angular plot is transformed by one TOF spectrum, where the intensities of the TOF fitting curve are first multiplied by a factor of $1/t$ element-wised then summed up. This extra factor is used to compensate the ionization efficiency. Details on how to transfer TOF into angular distribution is shown in the later parts of the Python code in Appendix C.

Since this kind of data requires a set of TOF spectra and usually takes longer time to collect, experimental condition might be slightly changed. To avoid errors from signal dropping through time, the TOF spectra are taken by scanning the outgoing angles back and forth several times then calibrating the intensity for both evaporation and scattering experiments. Take evaporation experiment as an example, the outgoing angles are scanned between 0° (viewing the jet surface normal) and 90° (viewing the rim of the jet) for four times, then each TOF is calibrated linearly with the data collected at 0°. In practical, the data collecting sequence of the outgoing angle is 0°, 30°, 60°, 90°, 0°, 80°, 50°, 20°, 0°, 10°, 40°, 70°, 0°, 75°, 45°, 15°, 0°, then finally 90°.

Chapter 3

Possible Background in Flat Jet Experiments

Before performing evaporation and scattering experiments on a flat jet, it is crucial to check and justify the feasibility of the current experimental setup. This is especially important for flat jets, since their vapor loads are much larger than cylindrical jets and can create unwanted background. Possible background sources were tested to evaluate how they affect the evaporation TOF. Some mistakes that we made are also discussed and should be avoided in the future. These are summarized in five topics in the later sections, which includes topic about gas leaking from the atmosphere, outgassing from porous material, stray gas in the main chamber, vapor backflow from the catcher, and extra cold trap around the jet.

3.1 Experimental

Most of the experimental setup are the same as mentioned in Chapter 2. If different setup is being used, it will be pointed out in each testing condition. Since neon evaporation and scattering from dodecane liquid jet has been chosen to be the research topic, the liquid sample is selected to be dodecane dissolved with neon in this chapter. Evaporation of both neon and dodecane are performed with jet surface normal aligned with the y-axis, which is indicated in Figure 2.2 and 2.11. All the evaporation data are collected at outgoing angle at 0° unless specified. The cylindrical jet and flat jet are both used and are generated by the microfluidic chips.

Typical mass spectra are shown in Figure 3.1 with different jet samples. No mass peak appears at 20 amu in both detector background and helium degassed dodecane flat jet. 20 amu only occurs when the jet is bubbled with neon. This suggested that our main chamber is clean and background free at mass 20 amu. Therefore, neon is a great evaporating target to use. All the dodecane data are collected through its dissociative ionization fragment $C_4H_9{}^+$ at mass 57 amu instead of parent ion at 170 amu. This is mainly due to the fact that the dirty quads create unstable signal at higher masses.

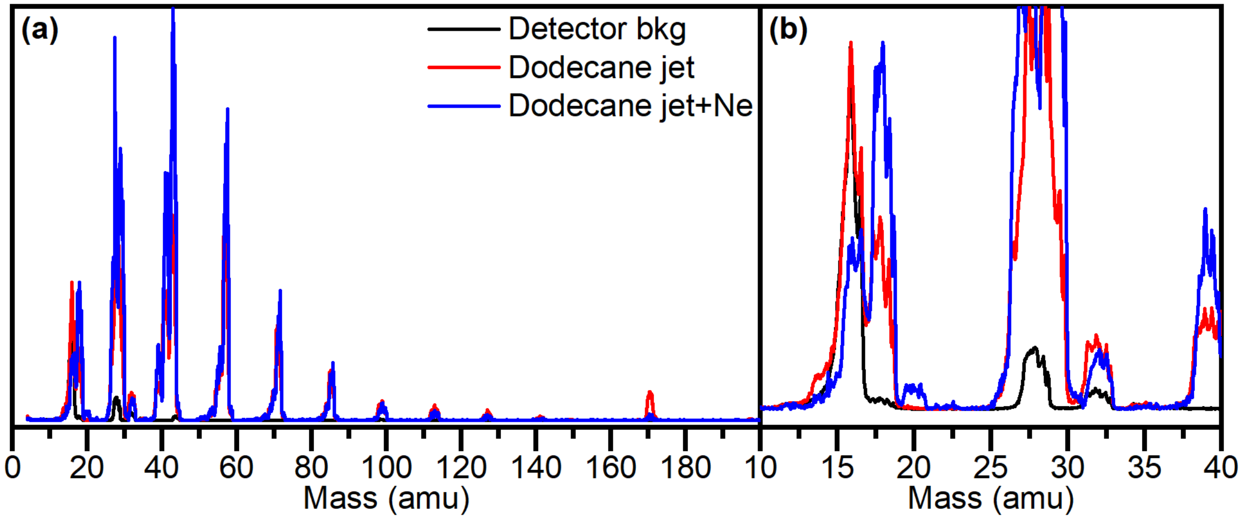


Figure 3.1: Mass spectrum of dodecane flat jets with (a) full mass range (4 to 200 amu) and (b) zoom-in range (10 to 40 amu). The intensities are normalized for easy comparison between detector background (black curve), dodecane jet (red curve), and dodecane degassing with neon (blue curve).

3.2 Leaking from the Atmosphere

Making sure that the vacuum chambers are sealed well might be an obvious thing to do. However, if the chambers are not sealed well, the TOF spectra will be strongly affected, and the data might not be analyzed correctly. A dodecane cylindrical jet with dissolved neon is tested with the main chamber at leaking and no-leaking condition, where the corresponding main chamber pressures are 4×10^{-5} and 4×10^{-6} Torr, respectively. Both neon and dodecane evaporation are taken at mass 20 and 57 amu, respectively. The results are shown in Figure 3.2.

Let's first look at Figure 3.2 (b), dodecane TOFs look the same regardless to leaking or no-leaking condition. This indicates that the leaking gases do not perturb the evaporating particles. Otherwise, the TOF should show a different distribution either due to mass contamination from atmosphere or extra number of collisions by the leaking-gases.

Neon TOFs, on the other hand, show clear differences. From no-leaking to leaking condition, the TOF shifts to shorter flight time, which indicates the evaporating molecules are either being accelerated or there are extra components contribute to the TOF. The acceleration is essentially a super-Maxwellian effect, which will be described more in the next chapter. Since the dodecane TOFs show no peak shifting, we believe that the mass contaminating explanation is more convincing. Therefore, we try to perform a simple bimodal fitting to fit the leaking data, which consists of a distribution having the same shape of the no-leaking data and a Maxwell-Boltzmann distribution with mass at 20 amu. For the first

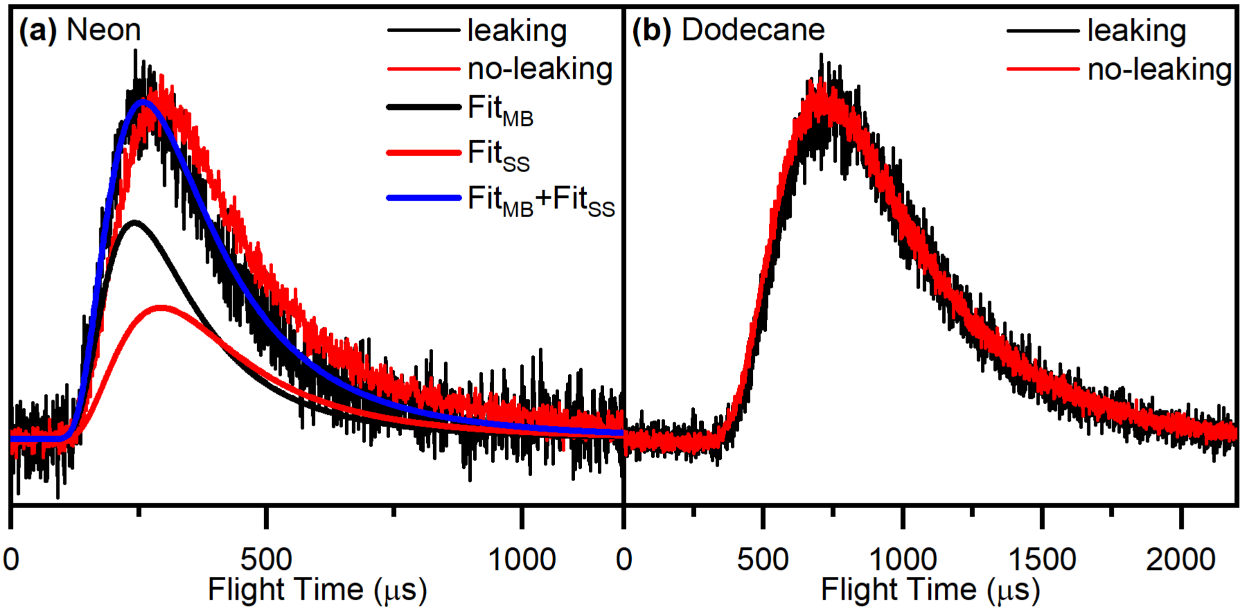


Figure 3.2: TOF spectra of evaporation of (a) neon (b) dodecane from a dodecane cylindrical jet under main chamber leaking (black curve) and no-leaking (red curve) condition. Fittings for leaking data in neon evaporation are calculated using a Maxwell-Boltzmann distribution at $m = 20\,\text{amu}$ and $T = 300\,\text{K}$ (thick black curve) and a supersonic distribution (thick red curve), which has the same fitting parameters as fitting for no-leaking data. The total fitting curve is shown in thick blue curve.

part, it is found that $N_{SS,meas}$ in Equation 2.14 captures the shape of no-leaking data very well. As for the second distribution, we choose to use $N_{MB,meas}$ in Equation 2.13 to fit it. With unfixed intensity for both components, the fitting results are shown in Figure 3.2 (a). The total fitting fits well to the leaking data and is shown in blue curve. The faster fitting component ($N_{MB,meas}$) has fitting parameters of $m = 20\,\text{amu}$ and $T = 300\,\text{K}$, which could be from deuterated room temperature water in the atmosphere.

3.3 Outgassing from Porous Materials

Before using a 2-slit chopper wheel, a 4-slit chopper wheel with two slits blocked by Kapton tape, a type of polyimide film, was used. The evaporation of a dodecane flat jet running at flow rate $2.5\,\text{mL}\,\text{min}^{-1}$ is studied. The TOF distribution at outgoing angle $\theta_f = 0°$ is shown as black curve in Figure 3.3. Besides the major peak at flight time around 750 µs, it seems that there is an extra slower feature peaking at 1750 µs. This type of bimodal feature where two peaks are separated clearly is something we never seen before, and therefore we suspect the extra slower feature is from unwanted background.

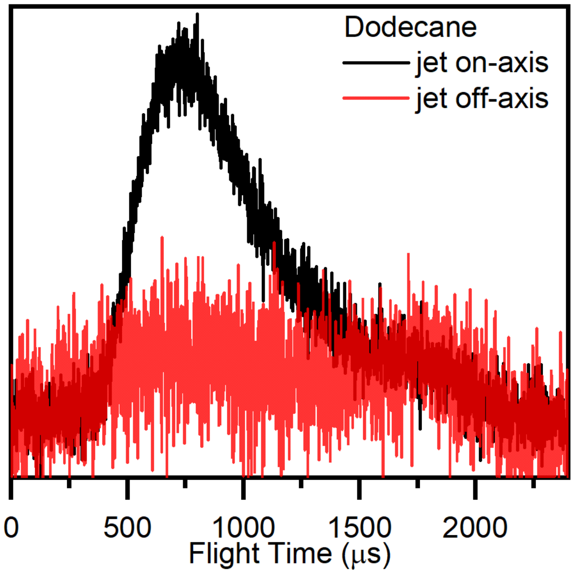


Figure 3.3: TOF spectrum dodecane evaporating from a dodecane flat jet using a Kapton tape blocked chopper wheel. Conditions of jet on-axis (black curve) and off-axis (red curve) to the detector are shown. The intensities are normalized for easy comparison.

To verify it, we tried to move the jet away from the detector. To be more specific, we move the jet 4 mm away from the reaction center on the axis perpendicular to the detector axis. The resulting TOF is shown as red curve in Figure 3.3. Large intensity decreases for both peaks. However, the portion of the slower feature increases that it is now having the same intensity as the faster one. This suggests us that the origin of slow component is different from the origin of the faster peak, which is from evaporation from the jet.

We then replace the taped 4-slit chopper with a 2-slit chopper. The slow peak disappears after doing so. We therefore conclude that the Kapton tape is the source of this kind of slow background, where dodecane particles can be absorb by the tape and then desorb. This suggest that one should not use any material that has strong outgassing properties around the jet and detector region. These outgassing materials are usually porous, and metals should be considered to replace them instead.

3.4 Stray Gas in the Main Chamber

In Chapter 2, it is mentioned that a newly designed copper wall is used as a cryo-pump inside the main chamber. The extra pumping capacity strongly affects the evaporation TOF especially for volatile liquids. Neon and dodecane evaporation are performed with a dodecane flat jet running at 3.5 mL min^{-1}. The TOF result at $\theta_f = 0°$ is shown in Figure 3.4 for both no cryo-pump and with cryo-pump condition.

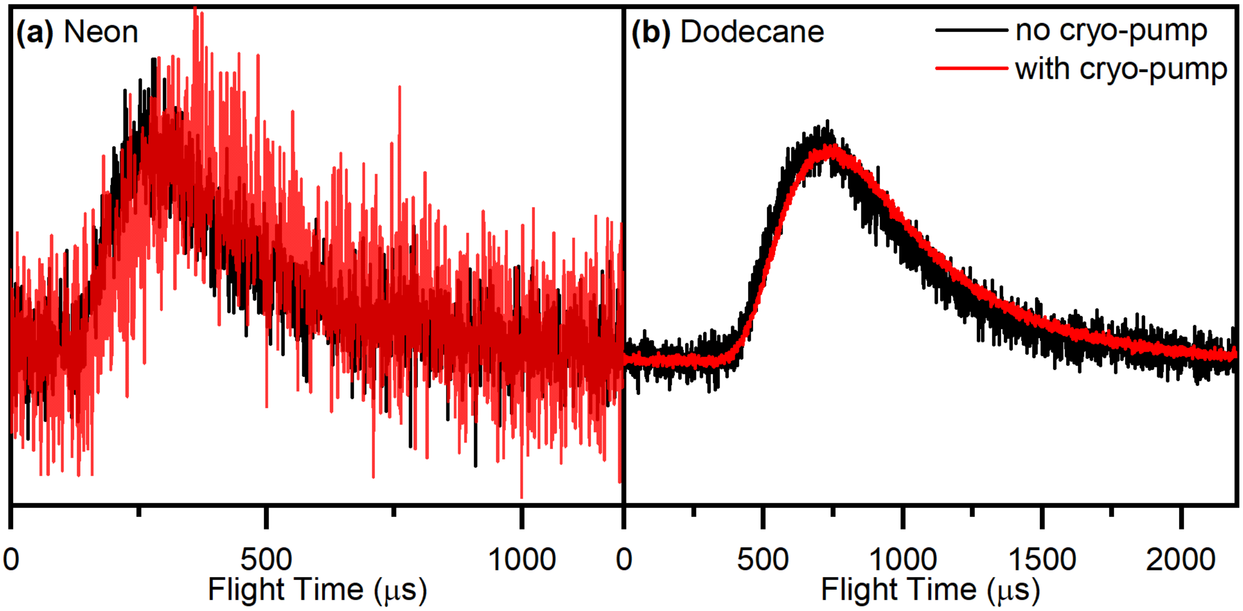


Figure 3.4: TOF spectra of evaporation of (a) neon (b) dodecane from a dodecane flat jet with cryo-pump on (black curve) and off (red curve). The intensities are normalized for easy comparison.

For both neon and dodecane, TOF distributions are much faster when the cryo-pump is not functioning. The main chamber pressure under such condition is as high as 9×10^{-5} Torr. While with extra pumping speed from cooling the copper wall, the main chamber pressure goes down to 5×10^{-6} Torr and the TOF distributions are slower for both cases. In the neon evaporation, the slower distribution could be due to deuterated water with mass 20 amu being trapped on the cooled copper wall. As for dodecane, the stray dodecane vapor can be dramatically pumped down by the copper wall and therefore decrease the vapor interference effect. This leads to a more Maxwellian distribution and will be discussed more in Chapter 4.

3.5 Backflow from the Catcher

A catcher is chosen to be used in this flat jet setup due to several advantages, such as lowering the evaporation gas load from the jet into the vacuum chamber. However, one must be careful about backflow from the catcher.[109] In our setup, two types of catchers are tested. One is a stainless-steel catcher with 1 mm orifice, which is made from a beam skimmer that is used to collimate the molecular beam in this system. The other catcher is a heated copper–beryllium catcher with a 0.5 mm orifice. This catcher is borrowed from Bernd Winter's group at FHI, who originally bought it from Microliquids GmbH (defunct, similar in design to the catcher produced by Advanced Microfluidic Systems GmbH, Göttingen).

Figure 3.5 shows the two catchers functioning under vacuum.

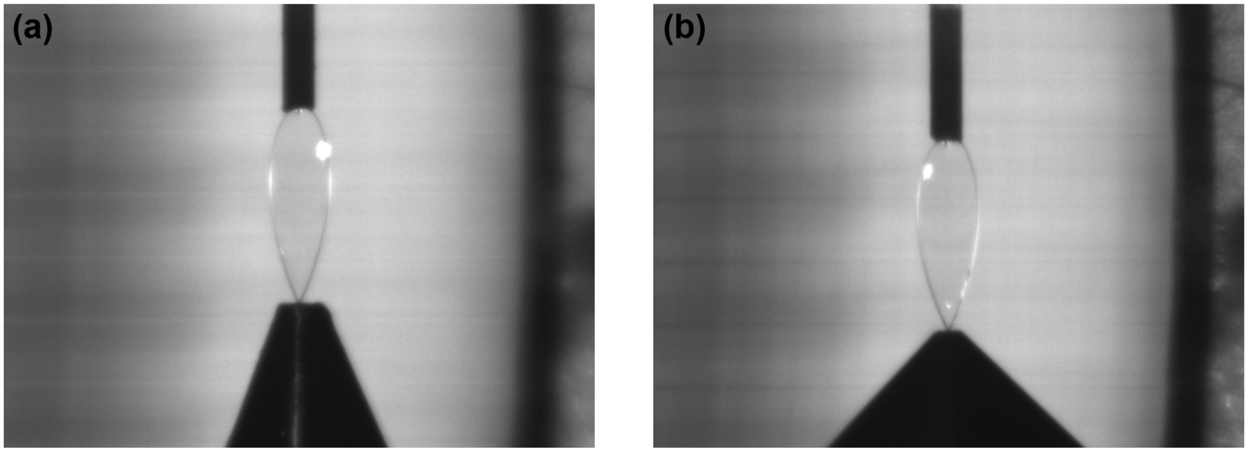


Figure 3.5: Photo of jet-catcher system viewing under vacuum. The catcher with (a) 1 mm and (b) 0.5 mm orifices are shown with dodecane flat jets running at 3.5 and 3.3 mL min^{-1}, respectively.

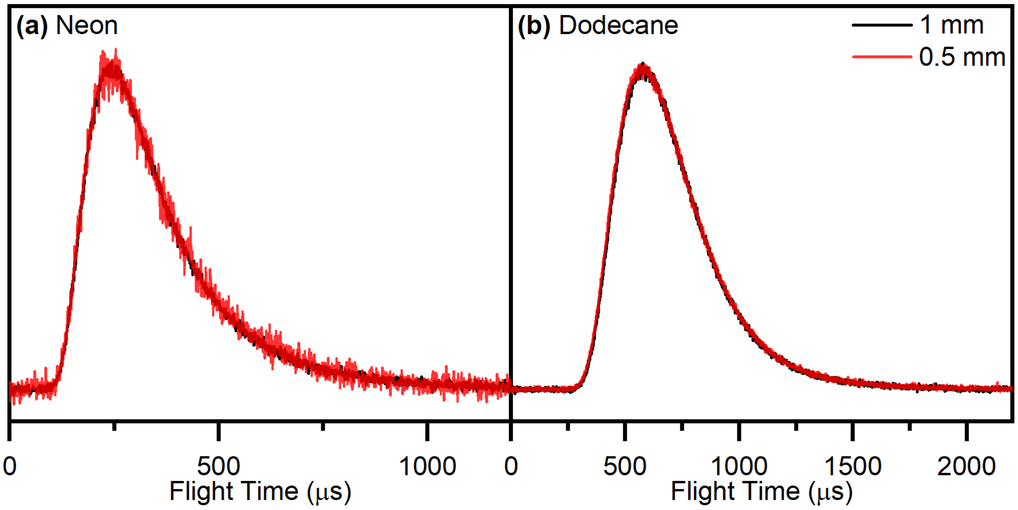


Figure 3.6: TOF spectra of evaporation of (a) neon (b) dodecane from a dodecane flat jet using a 1 mm (black curve) or a 0.5 mm (red curve) catcher. The intensities are normalized for easy comparison.

TOF distributions of neon and dodecane evaporation using different catchers are the same (see Figure 3.6). This is a strong proof that backflow from the catcher does not affect the dodecane evaporation. Otherwise, one would expect to see TOF shifts to more super

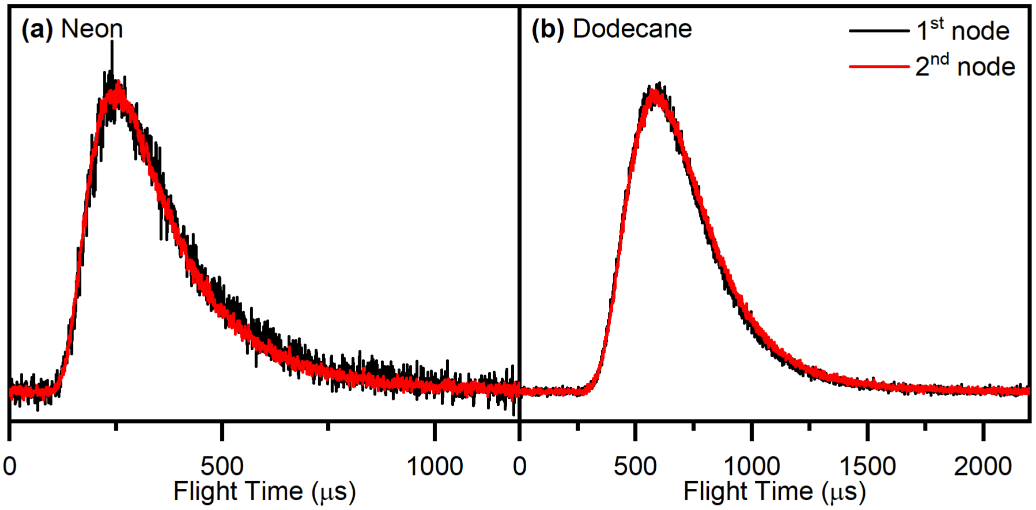


Figure 3.7: TOF spectra of evaporation of (a) neon (b) dodecane from a dodecane flat jet with catcher catching at the first (black curve) and second (red curve) node of the jet. The jet is caught by a 1 mm catcher. The intensities are normalized for easy comparison.

Maxwellian for 1 mm catcher than 0.5 mm catcher. Since there are no differences in the TOF, the 1 mm catcher is chosen to be used in Chapter 4 due to its advantage of easier handling. The larger orifice provides more tolerance if the jet is misaligned. Besides, it also gives more room for the second sheet to not touching the catcher's inner wall, which will also induce a vapor burst into the main chamber.

A comparison between catching the jet's first and second nodes is also tested and shown in Figure 3.7. No obvious difference is found in both neon and dodecane evaporation. This result indicates two things: first, the backflow from the catcher does not increase when catching the jet at the second node. Second, exposing the extra second sheet does not sufficiently increase the vapor load to affect the TOF distribution. This justifies the feasibility of using the second sheet for lower liquid temperatures in the future.

3.6 Cold Cage Around the Jet

An effective way to pump down the gases is to place the pump to the gas source as close as possible. Following this idea, it is straightforward to expect that the evaporating particles from the jet can be pumped away effectively if a cryo-pump is placed right next to it. In order to test this idea, a copper cage, designed by Marvin Pohl, attached to a cooling device was developed and shown in Figure 3.8.

The copper cage is designed to be compact due to geometric restrictions in our setup. The main cooling part is a 2 in diameter copper tube with 0.3 in wide slit and orifices. This slit-type of opening window allows the detector to view through it and get an angular

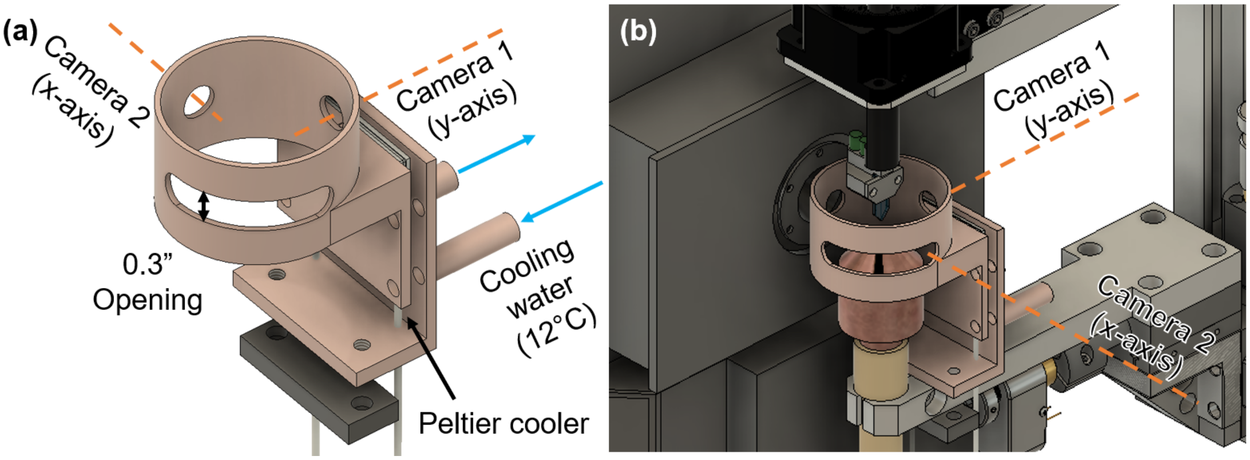


Figure 3.8: (a) 3D drawing of the copper cage. (b) 3D drawing of the copper cage installing into the chamber with liquid jet setup around the interaction region.

distribution of the evaporating signal from the flat jet. A Peltier cooler is used to cool down the cage to −14 °C, which is lower than the freezing point of dodecane and allows evaporating dodecane condenses onto the cage. Circulating cooling water at 12 °C is attached to the other side of the Peltier cooler in order to dissipate the heat from the cooler.

The TOF results of using the copper cage is shown in Figure 3.9. Conditions of no cage, cold cage at −14 °C and warm cage at room temperature 16 °C are shown. For neon evaporation, the conditions of no cage and cold cage show the same TOF distributions. For warm cage, however, the TOF is apparently slower instead of being super-Maxwellian. This indicates that there are slow components making the distribution looks decelerated. One possible reason could be that the evaporating particle undergoes secondary scattering from the cage and therefore performs a slower distribution. This process has been found in many crossed beam experiments and people has used cold fingers to trap the secondary bouncing gases.

Results of dodecane evaporation show similar trends as in neon, where the warm cage condition has slower distribution. The cold cage data even has slower distribution at its falling edge. Both of the neon and dodecane data indicate TOF spectra being affected by the cage. Nevertheless, we decided that we do not use the copper cage in this setup.

3.7 Conclusions

In this chapter, several possible sources that cause background in the TOF spectra are discussed and eliminated. All the experiments in the later chapters are carefully done with a well-sealed main chamber to prevent any background contribution from the atmosphere. All the possible outgassing sources, such as tape around the interaction region, are removed.

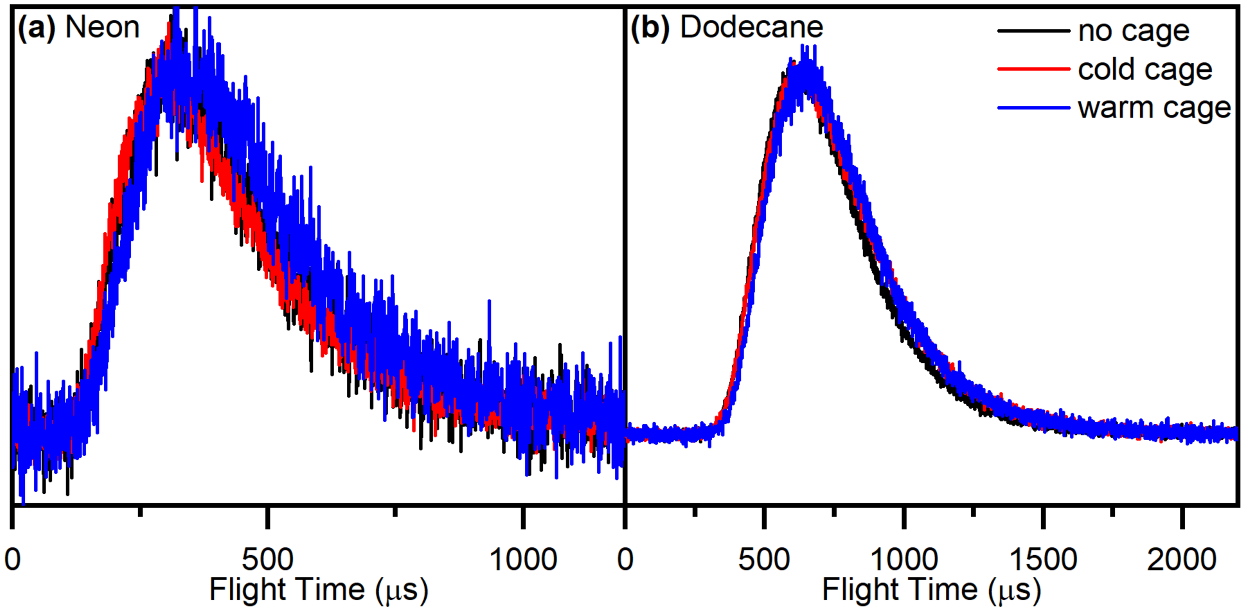


Figure 3.9: TOF spectra of evaporation of (a) neon (b) dodecane from a dodecane flat jet with no copper cage (black curve), cryo-cooled copper cage (red curve) and warm copper cage (blue curve). The intensities are normalized for easy comparison.

It is also shown that extra pumping speed is required even for low vapor pressure liquid like dodecane. Therefore, the cryo-pump is always functioning even when running a cylindrical jet no matter what liquid sample being used. The feasibility of using a catcher is also tested and was confirmed that there is no observable effect for the catcher backflow. The 1 mm catcher is then chosen to be used in the following experiments due to the larger alignment tolerance. Finally, it is concluded that it is best to not use a cryo-trap surrounding the jet. Doing so might confine the vapor expansion from the liquid jet itself and therefore increase probability of secondary collisions.

Chapter 4

Neon Evaporating and Scattering From a Dodecane Flat Jet

The content and figures of this chapter are reprinted or adapted with permission from C. Lee, M. N. Pohl, I. A. Ramphal, W. Yang, B. Winter, B. Abel, D. M. Neumark "Evaporation and molecular beam scattering from a flat liquid jet" *J. Phys. Chem. A*, **126**, *3373* (2022).

Abstract

In this chapter, two parts of experiments are performed. First, dodecane and neon evaporation from a flat dodecane liquid jet are studied. A super-Maxwellian effect is observed for dodecane due to the large collision number between the evaporating particles and the vapor above the liquid surface. For neon evaporation, however, Maxwellian distributions in the TOF spectra indicate no vapor interfering with evaporation after careful background subtraction. The angular distribution is also well fit by a cosine distribution. These phenomena suggest that neon is a good model system for evaporation experiments and can be further used as a scattering source.

In the second part, we further studied neon scattering from a flat dodecane liquid jet. Varying incident beam energy and incident angle are discussed. Two mechanisms are observed in the scattering process: one is the impulsive scattering (IS) and the other is the thermal desorption (TD). In the TOF spectra, the former one is described using a supersonic distribution while the later one is a Maxwell-Boltzmann distribution. Angular distributions are also analyzed, and the IS process shows maximum intensity at specular angle. Finally, hard- and soft-sphere models are used to give a kinematic picture to describe the energy transfer between the gas particles and liquid surface. It is found that the soft-sphere model fits the best for all cases, which indicates part of the energy from the impinging gas is transferred into internal excitation of liquid molecules when collision occurs.

4.1 Introduction

Evaporation is one of the most fundamental phenomena that can happen at an interface. It describes how particles interact with the liquid molecules then desorb. Understanding how particle desorbs is extremely helpful for understanding the scattering process. This is because evaporation considers only how particles escape the liquid surface. While for scattering experiments, the process roughly consists of three parts: gas absorbs into the liquid, gas dissolves into the bulk, and then gas desorbs from liquid to the gas-phase or vacuum. Not to mention that we are only considering non-reactive gas-liquid reaction. If reactions occur, more processes are needed to be discussed.

With knowledge learned from evaporation studies, mechanism in scattering experiments can be extrapolated. Experiments start from low vapor pressure ($P_{vap} = 10^{-7}$ Torr) liquids such as squalane and perfluoropolyether (PFPE) by Nathanson's group.[32] Atoms (neon and xenon) and molecules (SF_6) have been used to scattered onto these liquid surfaces carried by a wetted wheel setup,[33] which is only feasible for low vapor pressure liquid ($P_{vap} = 10^{-3}$ Torr). The most extreme case for the wetted wheel is probably salty water of 12 mol% LiBr solution at 212 K with vapor pressure 5 mTorr.[38] For more volatile liquids, the wetted wheel technique is no longer suitable because the liquid will freeze onto the wheel.

With these pioneering gas-liquid studies,[32, 31, 38] two possible mechanisms are revealed: impulsive scattering (IS) and thermal desorption (TD). These mechanisms show fast and slow velocity components in TOF spectra, respectively. The TD signal represents gas particles that undergo surface trapping and then thermalized prior to desorption. IS results from gas particles that do not penetrate the liquid and experience negligible surface residence times (with respect to thermalization). These IS particles then recoiling elastically or losing only a fraction of their initial energy. They are mostly scatted into the specular angle that depend sensitively on the initial energy and impact angle. (27) Note that IS and TD are limiting rather than clear-cut regimes for the scattering molecules from liquids. Nonetheless, these two mechanisms provide useful frameworks for the interpretation of our results, and we use it as such throughout this work.

To study volatile liquids, Faubel has developed liquid microjet technique to carry pure water at freezing point 273 K ($P_{vap} = 4.6$ Torr) into vacuum.[40] The flowing jet creates a special condition for the liquid being at its super cooled state without forming ice before it reaches any surfaces to nucleate. The first cylindrical microjet scattering experiment has been carried again by the Nathanson's group with neon and oxygen scattering off a 20 µm dodecane cylindrical jet at 295 K ($P_{vap} = 0.1$ Torr).[42] However, due to the curvature surface of the cylindrical jet, this setup is not suitable for obtaining angular distribution.

In this chapter, we aim to show the feasibility of the using a flat jet in a molecular beam scattering setup, where the flat jet combines benefits of both angular specificity of the wetted wheel and continuously refreshing liquid system like microjet. Therefore, we focused on noble gas evaporation and scattering. The liquid is selected to be dodecane, a key component of jet fuel surrogates.[113] Its vapor pressure is 1.5×10^{-2} Torr at temperature at 275 K). Under this pressure condition, it is considered to be more volatile than some long chain hydrocarbons, such as squalane. In the meantime of being semi-volatile, dodecane will not froze inside vacuum due to its low freezing point around 264 K. This allows us to turn on the jet even under vacuum and without worrying freezing it. This makes the experiment much easier to handle compared to pure water flat jet and is very convenient for our purpose of testing the flat jet system. As for the gas particle, noble gas neon has been chosen to be in the molecular beam to keep the scattering model simple.

4.2 Experimental

For both evaporation and scattering experiments, the liquid sample *n*-dodecane is purchased from TCI America with purity of 99.0% (TCD0968-500ML). After pre-filtering dodecane through a 2 µm porous of filter paper using a vacuum filter system, the liquid is then poured into the degassing system, which details are mentioned in Section 2.3. Sample liquid is then degassed with neon and helium for five times before doing the evaporation and scattering experiments, respectively. Since dodecane will not freeze under vacuum, the jet is only started after the main chamber reaches 2×10^{-6} Torr with copper wall cooled. Both flat and cylindrical jets are formed using the microfluidic chips.[63]

The flat jet is running at flow rate of $3.5\,\mathrm{mL\,min^{-1}}$ and cooled down by the double wall heat exchanger to jet temperature between −6 to 10 °C depends on experimental condition on that day. The HPLC pump pressure is around 35 bar under this range of liquid temperature and the jet size is around $1.7 \times 5.1\,\mathrm{mm^2}$ at vacuum. As for the cylindrical jet, because the flow rate is only $0.5\,\mathrm{mL\,min^{-1}}$, the cooling power is not so great and the liquid temperature can only go down to 16 °C.

A stainless-steel skimmer with 1 mm orifice size is used as a liquid catcher for both types of jets. The tip positions, however, are different. For flat jet, the tip is caught at the first node, which is roughly 5 mm below the tip of the chip nozzle. While for cylindrical jet, the catcher position is 3 mm below the tip. The height of the flat jet is set so that the detector is viewing the widest part of the flat sheet, which is around 2 mm below the nozzle tip. As for the cylindrical jet, the detector is viewing at the center of the exposed jet, which is 1.5 mm below the tip.

In evaporation experiments, the chopper wheel is always sitting between the detector and jet. The purpose of it is to set a time zero for the TOF spectra. The detector aperture is opened to big hole. This 3 mm big aperture and the 1.65 mm chopper slit are considered to provide a time opening of 29 and 15 µs by themselves, which are estimated using the slit center at diameter 17 cm and the chopper running at 200 Hz. The combination of the aperture and chopper gives a total opening of 45 µs. The time windows are shown in Figure 4.1 (a).

This 45 µs of opening window is the time resolution that limits the TOF data for evaporation experiment. However, it doesn't make a huge change to the TOF as long as the original distribution is as broad as hundreds of microseconds. This is true for Maxwellian distribution, which is often occurs in the evaporation process. Figure 4.1 (b) shows an example of a Maxwellian distribution of particles of 20 amu at 275 K calculated using Equation 2.13.

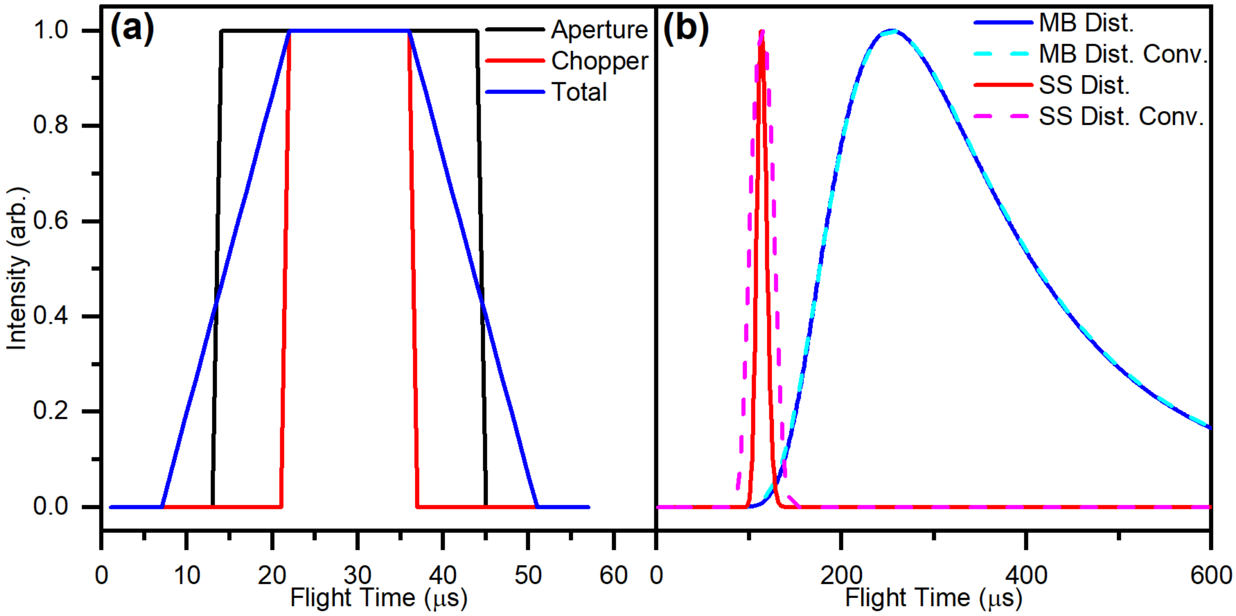


Figure 4.1: (a) Opening time of the detector aperture at big hole and chopper slit. Convolving both distributions gives a total opening time for the evaporation experiment. (b) TOF of both Maxwellian (blue) and supersonic (red) distributions are shown with (dashed line) and without (solid line) convolving with detector opening time.

No dramatic differences are seen after convolving it with the opening window. Therefore, no convolution is used in the analysis process. A supersonic distribution calculated using Equation 2.12 with $s = 15$ and $u = 1500\,\mathrm{m\,s^{-1}}$ is also shown in Figure 4.1 (b) for comparison. Its full width at half maximum (FWHM) is around 15 μs, which is comparable with the detector open window. The FWHM after convolution become 30 μs. This further suggests us that deconvolution from the data should be made with narrower distribution.

For scattering experiments, the detector aperture is also set at big hole so that the scattering signal is collected at its maximum intensity. Since the detector is opened to big hole with no chopper in front, the pressures in the detector chambers are quite high. The pressures in RI, RII and RIII can be as high as 3×10^{-7}, 3×10^{-10} and 2×10^{-10} Torr. This accelerate the speed of the quadrupole being contaminated by the liquid vapor. Fortunately, for neon with lower mass at 20 amu, we do not see strong intensity decreasing where the fluctuation in a set of scattering data is smaller than 5%. An intensity calibration is nevertheless being performed for scattering data.

Two kinds of gas beam are used for scattering experiments. One is pure neon (99.99% purity) and the other is 10% neon seeded in helium, which are often called slow and fast neon beam in this dissertation. The 10% neon is prepared in the laboratory by first pumped down the 48 L gas cylinder with mechanical pump for 1 hr. Then, 1 bar of pure neon is filled

Gas	$P_{backing}$ (bar)	Valve Vol. (V)	Valve Open Time (µs)	P_{source} (Torr)	Beam Vel. ($\mathrm{m\,s^{-1}}$)	FWHM (µs)
Slow Neon	4	200	15	8.6×10^{-6}	790	150
Fast Neon	4	200	15	7.7×10^{-6}	1460	60

Table 4.1: Amsterdam valve controller settings for slow and fast neon. Beam properties are also listed.

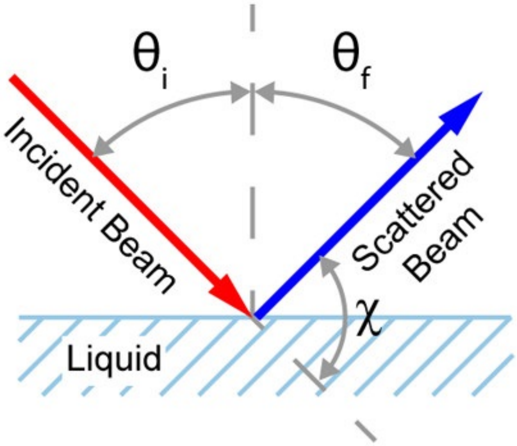


Figure 4.2: Schematic plot of the notation used in this dissertation. The incident angle θ_i, outgoing angle θ_f and deflection angle χ are shown.

into the cylinder and followed by 9 bar of pure helium topping the cylinder to 10 bar.

The molecular beam is generated by an Amsterdam valve with backing pressure of 4 bar for both slow and fast neon gases. Although the setting on the valve controller varies and depends on how the valve being tuned, the valve condition and relevant beam properties are listed in Table 4.1 for reference. One should note that the valve voltages listed here are much higher than mentioned in Section 2.2. This is mainly because that the condition below is tuned when we first received the valve with limited experiences. The piezo crystal broken several days after using this high voltage valve setting. This suggest that valve condition in Table 4.1 might be too tight for the piezo crystal to run for a long time and one should avoid pushing the valve to this critical condition.

For both evaporation and scattering experiments, they share the notations in this dissertation. The incident angle is set to be θ_i, while the outgoing angle is θ_f. Deflection angle χ is defined to be $180° - \theta_i - \theta_f$. Figure 4.2 shows the schematic plot for these notations.

4.3 Results and Analysis

Evaporation

The TOF spectra of dodecane and neon evaporation are collected and shown in Figure 4.3. Dodecane signal are detected by parking the quadrupole at dodecane's most intense ionization fragment $C_4H_9{}^+$ with mass 57 amu using ionization energy at 80 eV. This fragment is selected instead of the parent ion $C_{12}H_{26}{}^+$ with mass 170 amu due to the instability of larger mass on the mass spectra. The signal intensity often decreases dramatically when the detector is opened to big hole when the liquid jet is running. This phenomenon is unlikely due to intensity saturation, but because of the contaminated poles.

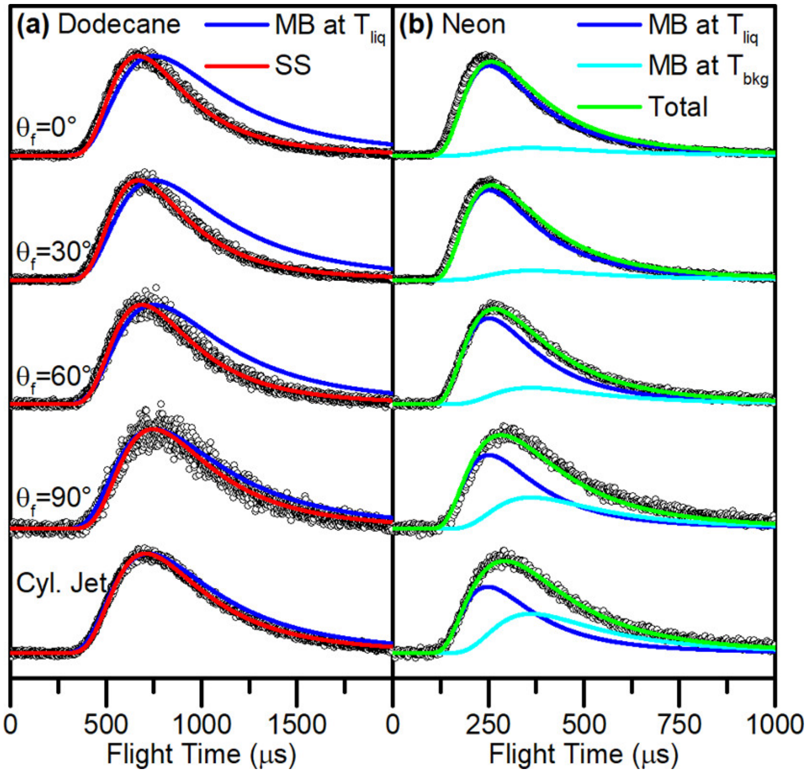


Figure 4.3: Normalized TOF spectra of evaporation of (a) dodecane (b) neon from a dodecane flat jet at liquid temperature 283 K. The bottom curves are evaporation from a dodecane cylindrical jet at liquid temperature 291 K. The data are fitted using a supersonic distribution (red traces) for the dodecane data. For neon, a sum of two Maxwellian distributions are fitted, where one is set at $T = T_{liq}$ (blue traces) the other is $T = T_{bkg}$ (light blue traces). The sums of the two contributions are shown by the green traces.

Dodecane evaporation from a cylindrical jet, shown as the bottom trace in Figure 4.3 (a), can be almost perfectly fitted with a Maxwell-Boltzmann distribution using Equation 2.13

with NFL and T being set as L' and liquid temperature $T_{liq} = 283\,\mathrm{K}$. The equation is written as

$$N_{MB,meas}(t) \propto \frac{1}{t^4} \exp\left[-\frac{m(\frac{L'}{t})^2}{2RT_{liq}}\right]. \tag{4.1}$$

This fit suggests that the evaporated molecules are fully thermalized with the surface liquid before they evaporate. Outgoing angles at 0°, 30°, 60° and 90° are shown in the upper traces in Figure 4.3. The 90° TOF spectrum is essentially reflecting the rim of the flat jet, which we estimated to be jet diameter around 50 µm. Despite the small liquid temperature difference between the flat and cylindrical jet (283 K and 291 K respectively), flat jet data strongly resembles the cylindrical jet data at the bottom where its jet diameter is around 25 µm. The intensity of evaporation from the rim is estimated to be less than 3% of the total flux. Thus, the contribution of evaporation from the rim is negligible at large outgoing angles.

Dodecane TOF shows a general trend of TOF shifting to shorter flight time when θ_f is closer to surface normal. These faster distributions indicate a super-Maxwellian velocity distribution similar to Equation 2.14, which can be described as

$$N_{SS,meas}(t) \propto \frac{1}{t^4} \exp\left[-\frac{m(\frac{L'}{t} - v_{SS})^2}{2RT_{SS}}\right]. \tag{4.2}$$

where v_{SS} and T_{SS} are set as fitting parameters. This TOF shifting trend is also shown in neon evaporation, where a narrowing distribution occurs when θ_f decrease. However, for neon, the Maxwell-Boltzmann distribution at T_{liq} agrees better with the 0° TOF spectrum instead of the 90° spectrum. This thus indicates that the angle-dependent deviating trends in dodecane and neon evaporation arise from different processes.

We first look at the super-Maxwellian effect in dodecane evaporation. This phenomenon is often being explained by vapor-phase collisions between the evaporating particles with the vapor particles above the liquid surface. Fittings are done by using a Python script shown in Appendix C.1. The fitting results for TOF spectra using Equation 4.2 are shown in Figure 4.3 as red curves. Fitting parameters of v_{SS} and T_{SS} are shown in Figure 4.4.

For cylindrical jet, the data are fit with $v_{SS} = 73\,\mathrm{m\,s^{-1}}$ and $T_{SS} = 210\,\mathrm{K}$ (open circle and square). The small deviation of these values from a Maxwell-Boltzmann distribution ($v_{SS} = 0\,\mathrm{m\,s^{-1}}$ and $T_{liq} = 289\,K$) indicates some collisions occur in cylindrical jet evaporation process. As for flat jet's parameters, v_{SS} increases from 90 to $135\,\mathrm{m\,s^{-1}}$ and T_{SS} decreases from 178 to 78 K with decreasing θ_f. This shows considerable thermal relaxation analogous to a supersonic beam expansion mentioned in Section 2.2.[40, 41] This analogy is most prominent at 0°.

The evolution of super-Maxwellian to Maxwellian distribution from 0° to 90° can be understood by the vapor cloud density around the jet. Higher vapor pressure, P_{vap}, leads to lower mean free path, λ, and therefore increases the number of collisions, N_{coll}, before

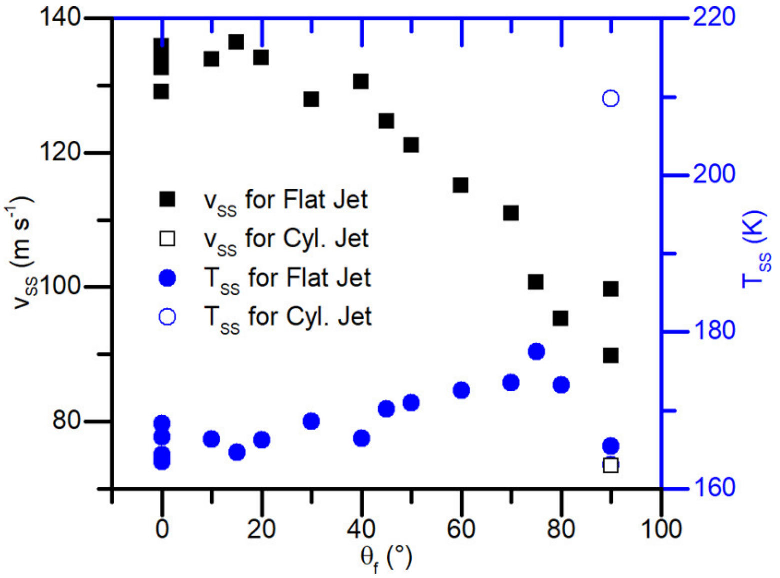


Figure 4.4: Supersonic fitting parameters of flow velocity v_{SS} (black squares) and local temperature T_{SS} (blue circles) for dodecane evaporation. Fittings for flat jet and cylindrical jet are shown in solid and open symbols.

the particles being detected by the detector. Method to estimate N_{coll} and its relative parameters can be found in Section 1.3 and the codes are provided in Appendix A.3. Only brief description about N_{coll} is mentioned here. Since P is estimated through analogy of electric field,[53] one can use the electric potential equations for a disk electric plate to calculate N_{coll}. The electric field around a finite disk distorts at the edges,[114, 115] which therefore predicts the evaporative flux decreases with increasing θ_f. Another way to describe the angular trend of N_{coll} is by using the cosine law of evaporation.[116, 9] It is stated that the directional distribution of the evaporating particles from a flat surface follows $\cos\theta_f$ distribution. Therefore, predicted by the cosine law, more collisions occur at angles closer to surface normal.

To reduce N_{coll}, one can either reduce P or the collision cross-section σ, which both are proportional to N_{coll}. For P, it is a function of temperature and is limited by the liquid freezing point. For example, it would be −9 °C for a dodecane liquid jet. σ, however, can be reduced by the choice of particles involved. The dodecane-dodecane hard-sphere cross-section has been estimated to be around 250 Å[42, 117] and is roughly two times smaller for cross-section of neon-dodecane. In fact, cross-section of neon-dodecane should be even smaller due to the weak interactions of noble gas with other species. This weak interaction can be estimated through the energy well depth ϵ of a Lennard Jones potential. Between dodecane-Ne dimer, ϵ has been estimated to be 0.9 kJ mol^{-1}, which is quite small compared to dodecane dimer 3 kJ mol^{-1} [1].

[1] Potential energy well of dodecane-Ne dimer, $\epsilon_{dodecane-Ne}$, is estimated by the potential energy well of

Due to the small potential energy well estimated above, it is unlikely that the evaporating neon is decelerated by dodecane liquid. Therefore, the slower distributions are considered to be originated from something else. Skinner and co-workers did simulations on noble gases evaporating from liquid water, and then discuss the evaporation energy and angular distribution of the evaporating particles.[116] One of the possible future works can be applying their model to our liquid dodecane system with noble gases.

After careful analyzing and system background check, the slower distribution has been explained by neon background wondering in the main chamber. Because neon has low boiling point (27 K), the cryo-shield inside the main chamber cannot trap neon efficiently. Therefore, the wondering neon inside the chamber will contribute an extra component in the TOF distribution. A bimodal fitting is proposed to analyze neon evaporation consisting of two Maxwell-Boltzmann distribution (Equation 2.13) at liquid and cryo-shield temperatures (T_{liq} and T_{bkg})

$$N_{MB,MB}(t) = \frac{c_{liq}}{t^4} \exp\left[-\frac{m(\frac{L'}{t})^2}{2RT_{liq}}\right] + \frac{c_{bkg}}{t^4} \exp\left[-\frac{m(\frac{L'}{t})^2}{2RT_{bkg}}\right] \tag{4.3}$$

where c_{liq} and c_{bkg} are the intensity fitting constants for the two distributions. Since the wondering neon is considered to be an isotropic contribution, the weighting of c_{bkg} is fixed to be constant over different outgoing angles. While for c_{liq}, it is set to be free fit. Fitting for neon evaporation uses Python script shown in Appendix C.1. T_{bkg}, which is obtained by simultaneously fitting all the outgoing angles, is fitted to be 137 K. This number is fairly close to the temperature measured on the cryo-shield (150 K). c_{bkg} is set to be the same for all outgoing angle since the wondering neon shouldn't have any angular specificity. To understand this, it is easier to investigate the angular distribution plot, first, which we can convert the "area" of the TOF spectra into angular distribution. The "area" is quoted because it requires conversion of ionization efficiency in order to get the correct intensity on the angular plot. One should check Section 2.5 for details on how to do so.

The angular plot for both T_{liq} and T_{bkg} neon are shown in Figure 4.5 with blue circles and light blue triangles. The T_{bkg} contribution is basically constant at all angles, which is set by the code. Whereas the T_{liq} contribution can be fit by the expected cosine distribution (dashed gray line), so for the dodecane evaporation shown as red squares. This highlights the unique ability of our experiment to measure angular distribution of evaporating particles from a volatile liquid flat jet.

To summarize the evaporation experiment, it is found that the background-adjusted neon component shows Maxwellian distribution in the TOF spectra and cosine distribution in the

dodecane dimer and neon dimer. For dodecane dimer, the value is extrapolated by Lennard-Jones potentials of hydrocarbons n-alkane from $n = 1$ to $n = 9$.[118] For dodecane where $n = 12$, the value is calibrated from alkane fluid system from $n = 1$ to $n = 16$, where the factor of $\epsilon_{dimer}/\epsilon_{fluid}$ is around 1.35. This brings to the value of $\epsilon_{dodecane\ dimer} = 1.35 \times \epsilon_{dodecane\ fluid} = 1.35 \times 2.26 \approx 3\,\mathrm{kJ\,mol^{-1}}$. With neon dimer reported as $\epsilon_{Ne\ dimer} = 0.27\,\mathrm{kJ\,mol^{-1}}$,[119] the well depth of dodecane-neon dimer is estimated as $\epsilon_{dodecane-Ne} = \sqrt{\epsilon_{dodecane\ dimer}\epsilon_{Ne\ dimer}} = \sqrt{3 \times 0.27} \approx 0.9\,\mathrm{kJ\,mol^{-1}}$.

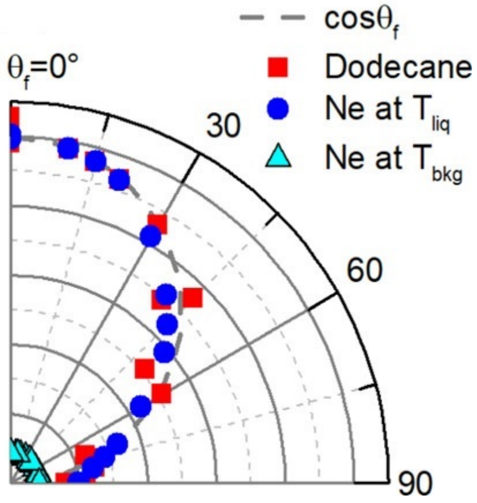


Figure 4.5: Angular distribution of the evaporating dodecane (red squares) and neon (blue circles) from a flat jet. The contribution of a constant background of wondering neon in the main chamber is also shown (light blue triangles). The dashed gray line represents cosine distribution.

angular plot. Both of them confirm that atoms with smaller van der Waals radii, like neon, can escape the dodecane vapor sheet without undergoing enough vapor phase collisions to make significant distortion to their velocity and angular distribution. This suggested that neon scattering from dodecane flat jet might as well not be largely perturbed by the dodecane vapor.

Scattering

Neon scattering from dodecane flat jet has never been studied, and the TOF spectra of slow and fast neon are first experimentally performed in the Neumark group and are shown in Figure 4.6. The translational energies of the slow and fast neon beam are measured to be 5.9 and 22 kJ mol^{-1}, respectively. The incident angles are both 60°. Jet temperatures are 278 and 277 K for slow and fast neon. Outgoing angles from 30° to 90° are taken with interval of 10°, while only 30°, 50°, 70° and 90° are shown. In each TOF spectrum, both slow and fast neon are collected with same number of sweeps of 10000. The worse signal-to-noise ratio for fast neon is simply due to the amount of neon in the fast neon beam is 10 times less than the pure slow neon beam. Note that in contrast to the evaporation studies, chopper wheel is not used in the scattering experiments and is not located along the detector axis.

Looking at the TOF spectra for both slow and fast neon, they both show trends of slower and broader TOF distribution at smaller outgoing angle, then become faster and narrower at larger outgoing angle. This trend is consistent with the expectation of thermal desorbed product following a cosine distribution. The data also shows bimodal features, with fast and

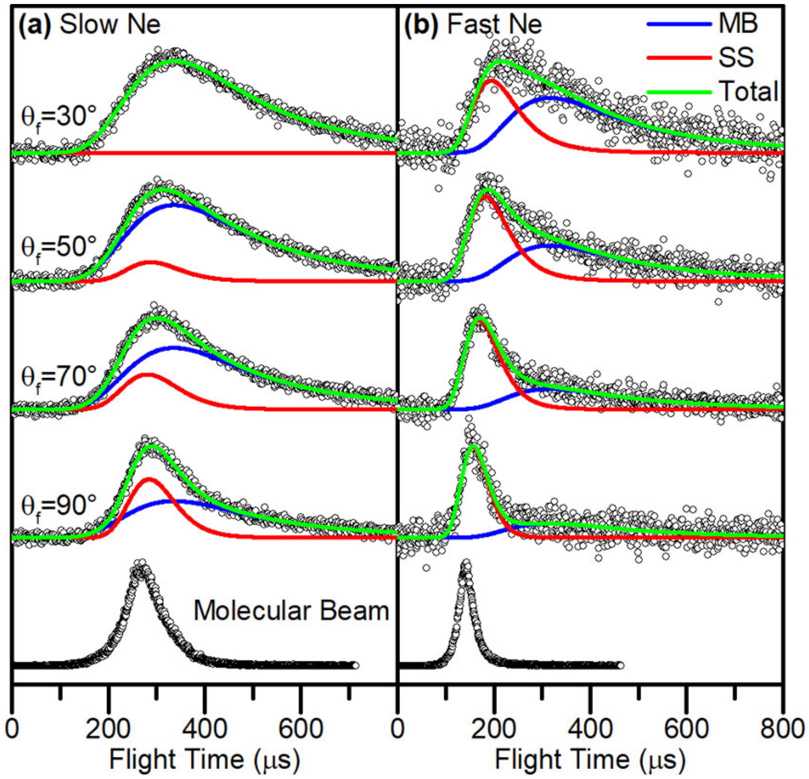


Figure 4.6: TOF spectra of (a) slow and (b) fast neon scattering off dodecane flat jet. The data are fitted using the sum of an Maxwellian distribution (blue traces) and a supersonic distribution (red traces). The sums of the two contributions are shown by the green traces. The bottom curves show the molecular beam temporal profiles.

slow components. It is then straightforward to use a two-component fit as a first try, where N_{MB} and N_{SS} are included and used to represent the IS and TD processes. The fitting function can be written as

$$N_{MB,SS}(t) = \frac{c_{MB}}{t^4} \exp\left[-\frac{m(\frac{L}{t})^2}{2RT_{liq}}\right] + \frac{c_{SS}}{t^4} \exp\left[-\frac{m(\frac{L}{t} - v_{SS})^2}{2RT_{SS}}\right] \tag{4.4}$$

where c_{MB} and c_{SS} are the fitting constants for the TD and IS processes. The fiting is done by Python code in Appendix C.2 and the fitting results are shown in Figure 4.6 where the blue and red traces represent the fittings of MB and SS distributions. The green traces are the sums of MB and SS traces, which shows great fit with the data. This hinted that this bimodal mechanism is a suitable fitting. It is also important to mention that fitting curve $N_{MB,SS}$ needs to be convolved with the molecular beam profile before fitting with the data. This is mainly because that the IS part, unlike TD, is usually narrower and could have FWHM as short as 50 µs. In case of having an IS process having comparable FWHM with the molecular beam, we decide to always fit the raw data with distributions convolved with beam profile.

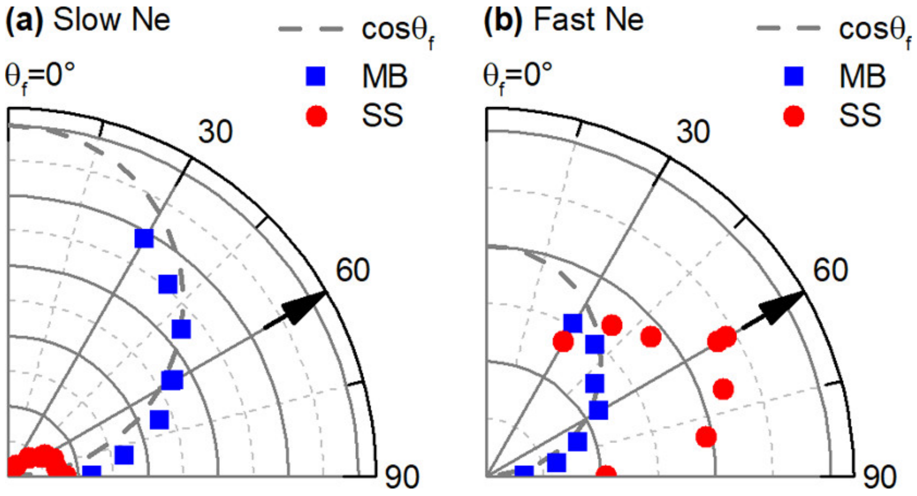


Figure 4.7: Angular distributions created from the integrated, non-normalized intensities of the neon scattering data with incident angle at 60°for (a) slow and (b) fast neon beams. Blue squares represent the TD contributions from fitting curves by MB distribution in the TOF spectra. Red circles are the IS contributions from the SS fitting distribution. The cosine function representing the expected angular distribution for evaporation is indicated by the dashed, gray line. Arrows indicate the specular angle.

Now using the fitted curves, angular distributions are shown in Figure 4.7 where blue squares and red circles represent the TD and IS contributions. Both slow and fast neon show great cosine fit for the Maxwell-Boltzmann component, same as what we see in evaporation studies. This indicates TD from the scattering data is evaporating from the surface with nearly completing thermal equilibration with the liquid temperature. While for IS parts, they both peaked at specular angles, which are similar to solid state scattering experiments.[120]

The IS/TD ratio, or the SS/MB ratio, also shows big differences between slow and fast neon where the former one has more TD component at all outgoing angles. This reflects a higher trapping probability of neon at the jet surface for lower collision energies, where the gas-surface interaction potential and thermal motion on the surface become more relevant. These trapped particles therefore undergo thermal equilibration with the liquid and become part of the TD. On the other hand, particles with higher translational energy can escape the surface potential much easily and then contributes to become IS.

Besides comparing different incident energies, we also carried out fast neon scattering at different incident angles 45° and 80° as shown in Figure 4.8. The overall trends remain similar where the spectra get narrower as θ_f increases. After using Equation 4.4 to fit IS and TD, the fitted results can be transferred into angular distribution and shown in Figure 4.9.

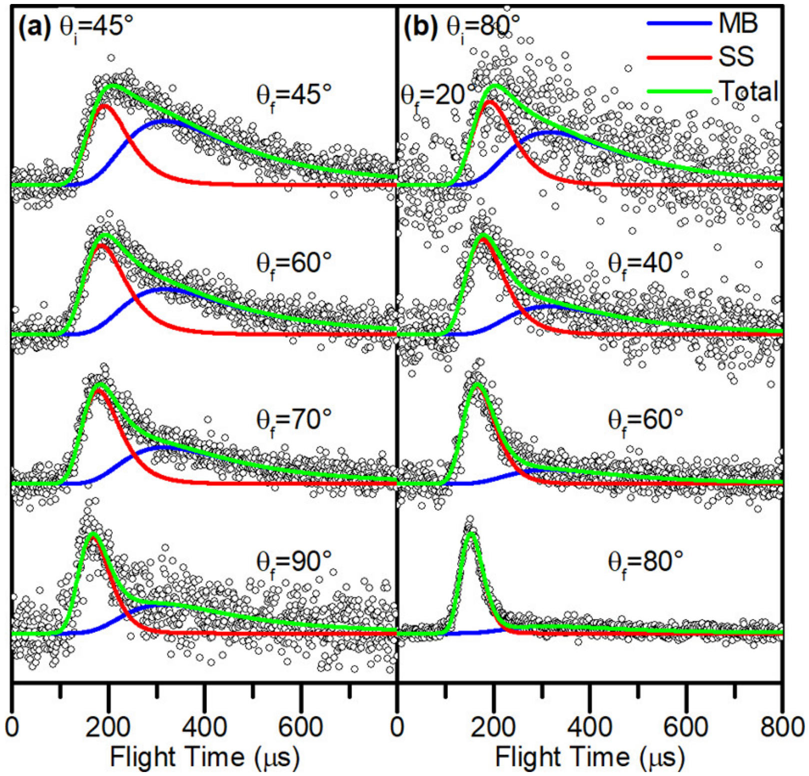


Figure 4.8: TOF spectra of neon scattering off dodecane flat jet with incident angle at (a) 45° and (b) 80°. The data are fitted using the sum of a Maxwellian distribution (blue traces) and a supersonic distribution (red traces). The sums of the two contributions are shown by the green traces.

For $\theta_i = 80°$, the IS part peaks around the specular angle. One should note that outgoing angle at 90° is not shown because of molecular beam leaking into the detector. As for $\theta_i = 45°$, outgoing angles can only be provided from 45° to 90°. This is mainly due to the geometric restriction of our scattering setup. Therefore, it is hard to state that if the outgoing angle is peaked at 45° or not. Another thought for this breakdown of not peaking at specular angle is that there could be other scattering mechanisms going on, which means the current bimodal fitting system is too simplified.

The IS/TD ratio increase with θ_i, which is consistent with existing literature of gas-solid scattering experiments.[120] These results show higher trapping probability of neon at smaller collision energy as the gas-liquid interfacial potential and thermal motion of the liquid surface become more relevant. These trapped particles may then thermal equilibrated with the liquid and eventually evaporate, which lead to the TD process. On the other hand, higher energy particles can escape the surface potential more easily.

Now if we focus on the TD contributions, they follow the cosine distribution for both incident angles, which again emphasize the MB component is from thermal equilibrium

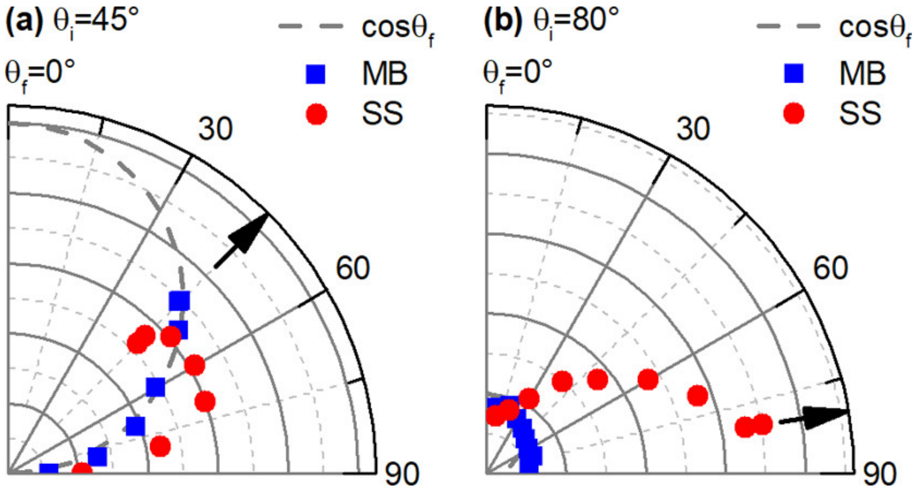


Figure 4.9: Angular distributions created from the integrated, non-normalized intensities of the fast neon beam scattering data with incident angle at (a) 45° and (b) 80°. Blue squares represent the TD contributions from fitting curves by MB distribution in the TOF spectra. Red circles are the IS contributions from the SS fitting distribution. The cosine function representing the expected angular distribution for evaporation is indicated by the dashed, gray line. Arrows indicate the specular angle.

processes. Another interesting thing to mention is that the TD contribution is dramatically smaller in the 80° case than 45°. This can be understood by having a more head-on collision leads to more energy transfer.[121]

We now investigate the energy transfer between the incident gas beam and the liquid surface by analyzing the SS component in the TOF spectra. Historically, researcher has been using a "soft-sphere" kinematic model to explain energy transfer of both gas-solid[122] and gas-liquid[123] scattering experiments. This model states that the average fractional energy loss in the IS channel can be described as a function of the deflection angle χ[31, 123, 124]

$$\frac{\Delta E}{E_i} \approx \frac{2\mu}{(1+\mu)^2}\left[1+\mu(\sin\chi)^2-\cos\chi\sqrt{1+\mu^2(\sin\chi)^2-\frac{E_{int}}{E_i}(\mu+1)}+\frac{E_{int}}{E_i}+\left(\frac{\mu+1}{2\mu}\right)\right] \left[1+\frac{V-2RT_{liq}}{E_i}\right] \tag{4.5}$$

where ΔE is the absolute change in translational energy and is defined as $\Delta E = E_i - \langle E_{IS}\rangle$, which E_i is the incident molecular beam translational energy and $\langle E_{IS}\rangle$ is the average energy in the IS channel. μ is the mass ratio of gas particle and effective surface mass and can be written as $\mu = m_{gas}/m_{eff}$. Deflection angle χ is equal to $180° - (\theta_i + \theta_f)$. E_{int} is the total internal excitation of the gas and liquid particles after collision. T_{liq} is the temperature of the liquid surface. V is the potential energy well, or the interaction potential of the gas-liquid surface. Since neon is a monoatomic gas, it has no internal degree of freedom for transferred energy to be deposited to. Therefore, E_{int} then only contains the internal excitation of the liquid. For V in this case, it is assumed to be 0 due to the low potential energy well in the neon-dodecane system (see footnote in the previous evaporation Subsection).

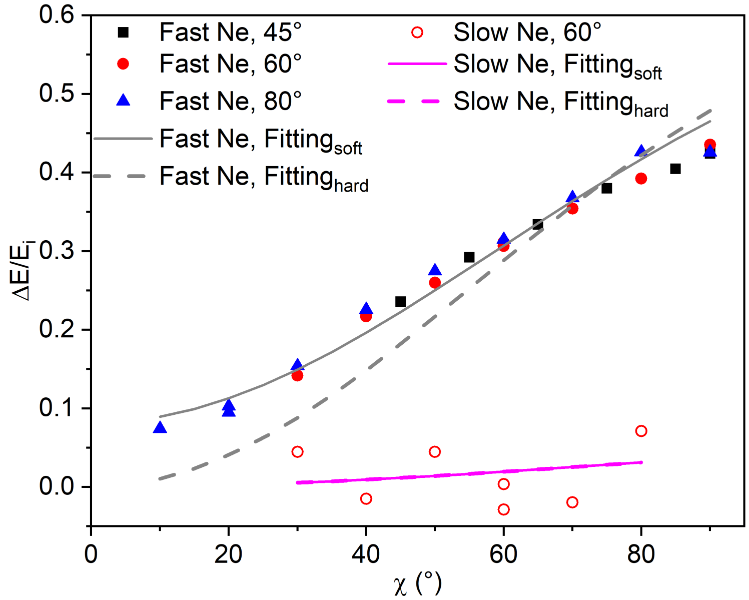


Figure 4.10: Average fractional energy loss as a function of the deflection angle for impulsively scattered slow and fast neon atoms from a dodecane flat jet, where the incident beam energies are 5.9 and 22 kJ mol^{-1}. The solid lines give the predictions of the soft-sphere model, where the incident atom interacts with a localized region of the surface with effective mass, m_{eff}, and this region may increase its internal energy, E_{int}, during collisions. The dashed lines are the predictions of the hard-sphere model where internal excitation has been set to zero. For the slow beam, the soft-sphere and hard-sphere models give indistinguishable predictions. The fitting results for slow and fast neon using the soft-sphere model are $m_{eff} = 211$ and 64 amu and $E_{int} = 0$ and 2.6 kJ mol^{-1}, respectively, whereas the hard-sphere model predicts $m_{eff} = 211$ and 46 amu, respectively.

To obtain $\Delta E/E_i$, E_i and $\langle E_{IS}\rangle$ are calculated from the average energy of the supersonic fitting results for the beam profile and IS component, respectively. The results are shown in Figure 4.10 where solid symbols represent fast neon and open symbols for slow neon. A Python script is written to fit the data with Equation 4.5 and the scripts are shown in Appendix C.3. Parameters μ and E_{int} are set to be free fitting for soft-sphere model. While for hard-sphere model, E_{int} is set to be 0, since the model assumes collision is an elastic scattering process and no internal energy is transferred.

Gas	$\boldsymbol{E_i}$ **(kJ mol^{-1})**	$\boldsymbol{m_{eff,soft}}$ **(amu)**	$\boldsymbol{E_{int,soft}}$ **(kJ mol^{-1})**	$\boldsymbol{m_{eff,hard}}$ **(amu)**	$\boldsymbol{E_{int,hard}}$ **(kJ mol^{-1})**
Slow Neon	5.9	211	0	211	0
Fast Neon	22	64	2.6	46	0

Table 4.2: Fitting parameters of soft- and hard-sphere kinematic models for slow and fast neon scattering from dodecane flat jet.

Looking at the data of fast neon scattering, results from all incident angles (45°, 60° and 80°) lie on a same curve. This suggests that the energy transfer is independent to the incident angle and, instead, depends on deflection angle. This aligns with the prediction of soft-sphere model shown in Equation 4.5. Fitted with all three incident angles, the fitting results for m_{eff} are 64 and 46 amu for soft- and hard-sphere model, while E_{int} are 2.6 and 0 kJ mol^{-1}. It is clear that the soft-sphere model is a better fitting method, and it captures the energy transfer dynamic much better than hard-sphere model at smaller deflection angle. This result suggests part of the energy is transferred into the liquid during the collision process and the amount is estimated to be 12% of E_i. The effective surface masses for both models are lighter than a single dodecane molecule (170 amu) and can be indicated that when the collision occurs, only part of the dodecane molecule joins the collision process.

The result of slow neon, however, fluctuates a lot. This could be due to the slow TOF distributions. For a slow distribution with broad beam profile, it is hard to distinguish IS and TD processes while they mostly merged together. For example, at outgoing angle 30° in slow neon scattering, a single Maxwell-Boltzmann distribution at the liquid temperature can fitted quite well. This leads to either a tiny contribution of IS, or the scattered beam has been decelerated by the liquid surface and being combined into the TD component.

Nevertheless, $\Delta E/E_i$ for slow neon is much smaller than fast neon. This leads to an increasing m_{eff} for slow neon and the value is fitted to be 211 amu for both soft- and hard-sphere models. Note that this effective surface mass decreases with increasing incident energy because the timescale of the collision is shorter and fewer surface atoms move cooperatively. A higher effective mass is usually explained by the higher "stiffness" of the surface which leads to higher recoil energies of the incident atoms and, consequently, less energy transfer to the surface. To summarize the fitting results, Table 4.2 lists all the fitting values for slow and fast neon with soft- and hard-sphere model.

Finally, a comparison between the flat and cylindrical jet is performed and shown in Figure 4.11. In this experiment, the molecular beam and detector axis are set to be perpendicular for both flat and cylindrical jets. The average incident angle for a cylindrical jet is calculated to be $33 \pm 22°$ and the outgoing angle is $57 \pm 22°$.[53] Therefore, to make a comparison, the flat jet is rotated so that the incident angle is 33° and the outgoing angle is 57°.

Since the accumulation time for both flat and cylindrical jets are around 13 min (which is equal to 150000 sweeps), the data quality directly shows huge differences on signal-to-noise ratio. For flat jet, the S/N ratio is 4.4 times better than the cylindrical jet, which represents around 20-fold improvement of the acquisition time t by scaling S/N with $\sqrt{t}$. This shows a great benefit of using a flat jet for shorter data acquisition time. Similar comparison has also been tested with argon scattering off squalane.[42, 53]

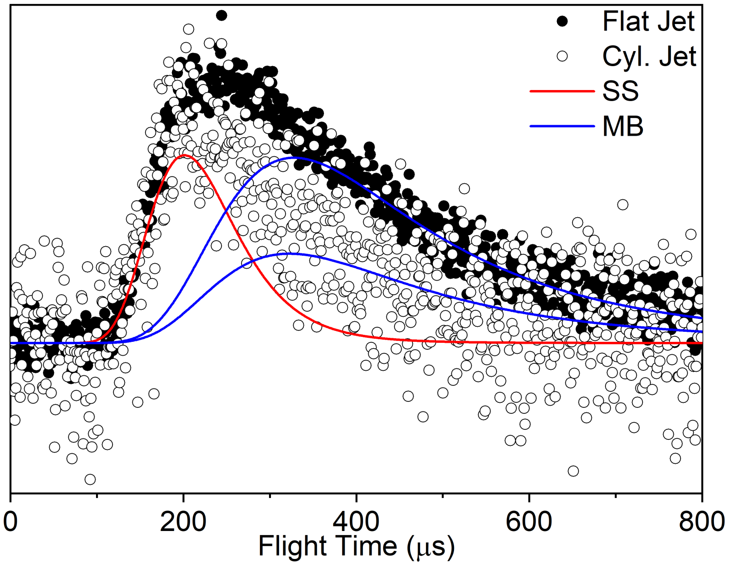


Figure 4.11: TOF spectra of neon scattering off dodecane flat (solid circles) and cylindrical (open circles) jet. The incident angle for flat jet is 33°. The angle between the molecular beam and detector axes is 90° for both jets. Solid lines show simulations of IS (red trace) and TD (blue trace) components. T_{liq} are 278 K and 283 K for flat and cylindrical jet, respectively.

A bimodal fitting with Equation 4.4 is again being used to fit the IS and TD components for both cylindrical and flat jets. The TOF distributions for both jets are first fitted individually. The IS channels show similar fitting parameters of v_{SS} and T_{SS} between both jets. Then, using the average value of these parameters, the IS channels are fitted again and so does the TD channels, which are shown in Figure 4.11 as red and blue curves. The similarity of the distribution for IS channels is also predicted by the kinematic model, as the energy loss depends only on deflection angles, instead of incident and outgoing angles.

Lastly, it is worth to mentioned that the TD process in our cylindrical jet data has more contribution than the same neon-dodecane system done by the Nathanson's group.[42] One can explain this by the energy difference of the incident beam, which are 22 and 50 kJ mol^{-1} in ours and their case. Predicted by the kinematic model (Equation 4.5), the smaller the E_i, the more energy loss for the IS component. Thus, the probability of the impinging gas to be trapped onto the liquid surface increases and therefore becomes undergoing TD.

Nathanson's group also studied the scattering differences between the flat and cylindrical jets using argon-squalane system.[42] They also found great improvement using a wetted wheel. The IS/TD ratio, however, is larger for flat jet, whereas we found less TD in the cylindrical jet condition. We are not sure what causes this difference, but one should note that there are some fundamental differences between our setup and theirs. First, of course, we are using a neon-dodecane system whereas they are using argon-squalane. Secondly, the incident beam energy in their system is $90\,\mathrm{kJ\,mol^{-1}}$, whereas ours is only $22\,\mathrm{kJ\,mol^{-1}}$. Finally, Nathanson's group uses a larger cylindrical jet with diameter of 50 µm larger than ours (28 µm). Therefore, performing a scattering experiment with similar conditions in order to compare ours result with theirs might not be a bad idea.

4.4 Conclusions

Dodecane and neon evaporating from dodecane flat jet have been used to show the feasibility of the flat jet system in our current molecular beam scattering apparatus. The TOF distribution of dodecane evaporation shows super-Maxwellian effect and indicates high collision numbers above the liquid surface. The angular distribution, however, follows the cosine law quite well. As for neon with much smaller collision cross-section than dodecane, the TOF spectra fits well with the Maxwell-Boltzmann distribution after excluding the background neon in the main chamber. The angular distribution again fits the cosine law very well. This shows that neon doesn't suffer from vapor collision at the flat jet surface and can be used as impinging particles for scattering experiments without worrying nascent signal being perturbed by liquid vapor.

Neon scattering off dodecane flat jet is therefore then performed. Two mechanisms are observed in the scattering process. The first one is the thermal desorption process, where the incident particles being fully equilibrated with the liquid molecules then desorb. The translational energy distribution of this mechanism can be illustrated by a Maxwell-Boltzmann distribution and its angular distribution follows a cosine distribution. The second process is the impulsive scattering. Its TOF distribution is well fitted by a supersonic distribution. By looking its angular plot, it shows more trapping probability at lower incident energy and smaller incident angle. The intensity of the impulsive scattering component peaks at specular angle, which is similar to a gas-solid scattering situation.

A soft-sphere kinematic model is being used to explain the energy transfer between the gas and liquid on the interface during the scattering process. It is found that the effective surface mass for dodecane in the fast neon-dodecane system is 64 amu while the number increases to 211 amu for slow neon. This shows a "stiffer" surface property for slow neon, which could be due to a longer recoiling time on the surface. Internal energy excitation of the liquid can also be studied and was suggested to be 12% and 0% for incident beam of fast and slow neon, respectively.

Comparison between flat and cylindrical jets shows great advantages of shorter accumulation time and better signal-to-noise ratio for flat jet. Both jets share the same impulsive scattering TOF distributions despite that cylindrical jet has larger angle uncertainty. This also confirms that the soft-sphere model is the suitable kinematic model to describe the gas-liquid dynamics, which predicts the energy loss of the impulsive scattering process only depends on deflection angles, instead of incident and outgoing angles.

Chapter 5

Prospective Experiment for Extracting Residence Time

This chapter mainly introduces a proposed experimental method to extract residence time of gas particles staying in the liquid. The way to perform this type of experiment is adding a chopper into the detector axis. One should note that the data and analyzing code used in this chapter are developed in a preliminary stage. More sophisticated and careful data processing are required in the future.

5.1 Residence Time

After molecular beam impinging the liquid surface, gases can be dissolved into the liquid then desorb. The time for the gas particle staying in the liquid is related to a parameter called residence time, τ. One should note that only non-dissociative gas desorption condition is discussed here. For more complicated condition, such as reaction happens in the liquid, Bradley Ringeisen's and Annabel Muenter's theses provide great descriptions about it.[112, 125, 126, 127] Equations written below are mainly based on derivation from the Nathanson's group and only the essential ones are shown here. We start explaining τ from the diffusion equation defined by Fick's second law of diffusion

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} \tag{5.1}$$

where $C(x,t)$ is the concentration of gas particle in the solution at liquid depth of x and time t. D is the diffusion constant of the gas in the liquid, which can be further defined as $D = k^2\tau$ with k is the desorption rate constant. Equation 5.1 can be analytically solved when the gas beam is continuously impinging the liquid surface with constant intensity. Details of the derivation process is shown in Bradley's thesis and only the resulting equation is shown below[112]

$$C(x,t) = erfc\left(\frac{x}{4Dt}\right) - exp\left[\frac{kx}{D}\right] exp\left[\frac{k^2t}{D}\right] erfc\left(k\sqrt{\frac{t}{D}} + \frac{x}{\sqrt{4Dt}}\right). \tag{5.2}$$

This analytical form, however, may not be convenient to use in our setup because we do not have a constant beam. Instead, a numerical form is more useful for more flexible beam application. Equation 5.1 is then being rewritten as

$$\frac{C_x^{t+1} - C_x^t}{\Delta t} = D\left[\frac{C_{x+1}^{t+1} - 2C_x^{t+1} + C_{x-1}^{t+1}}{(\Delta x)^2}\right] \tag{5.3}$$

where Δt and Δx is now set to be the step size for numerical calculation for desorption time and liquid depth. The number 1s in this equation are essentially equals to Δt and Δx but are written in numbers in order to reflect the matrix size in the later description. A schematic plot to describe the parameters are shown in Figure 5.1.

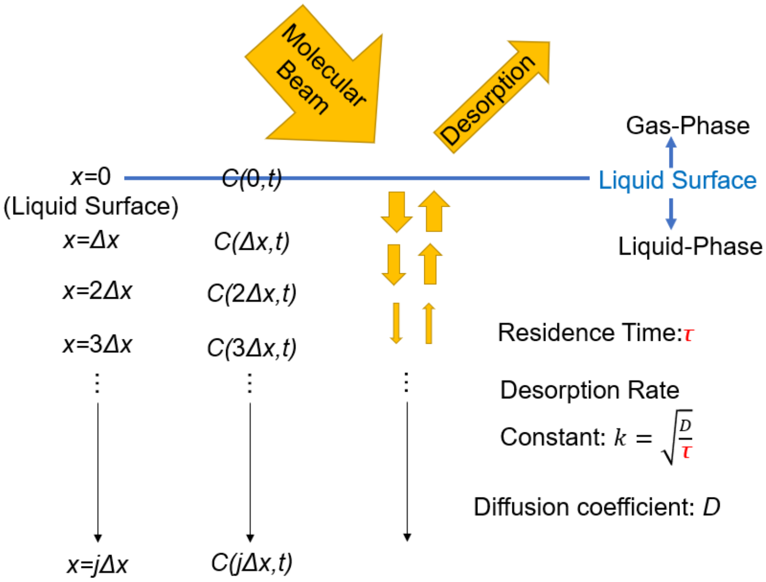


Figure 5.1: Schematic of a diffusion and evaporation dynamic for gases in a molecular beam dissolve into the liquid then desorb.

By rewriting Equation 5.3, one can get concentration at a certain time and liquid depth as

$$C_x^t = -a_1C_{x-1}^{t+1} + (2a_1 + 1)C_x^{t+1} - a_1C_{x+1}^{t+1} \tag{5.4}$$

where $a_1 = D(\Delta t)/(\Delta x)^2$. This equation basically tells us that the diffusion at liquid layer x only relates to its adjacent layers $x-1$ and $x+1$. Therefore, one can obtain the concentration propagating profile at different t and x as long as the boundary condition of liquid concentration at surface $x = 0$ at $t+1$, C_0^{t+1}, is defined. Using the Fick's first law, the layer gradient of C_0 can be defined as

$$-D\frac{\partial C(x,t)}{\partial x}\bigg|_{x=0} = k(I_{beam} - C(x,t))\bigg|_{x=0} \tag{5.5}$$

where I_{beam} is the intensity of the beam. Since the numerical form of Equation 5.5 is $-D((C_1^{t+1} - C_0^{t+1})/\Delta x) = k(I_{beam} - C_0^{t+1})$, we can written it in a general form including I_{beam} as

$$C_0^{t+1} = \frac{kI_{beam}\Delta x}{D + k\Delta x} + \frac{D}{D + k\Delta x}C_1^{t+1}. \tag{5.6}$$

We can now apply Equation 5.6 into Equation 5.4 to replace C_0^{t+1} to a function of C_1^{t+1}. The result shows

$$C_1^t = \left((2a_1 + 1) - \frac{a_1 D}{D + k\Delta x}\right)C_1^{t+1} - a_1 C_2^{t+1} - a_1\frac{kI_{beam}\Delta x}{D + k\Delta x}. \tag{5.7}$$

With Equation 5.7 defining C_1^t and Equation 5.4 defining the rest of the x, the total concentration propagating equations for this constant beam-on condition can be written in a matrix form shown below

$$\begin{bmatrix} (2a_1+1) - \frac{a_1 D}{D+k\Delta x} & -a_1 & 0 & ... & 0 & 0 & 0 \\ -a_1 & 2a_1+1 & -a_1 & ... & 0 & 0 & 0 \\ 0 & -a_1 & 2a_1+1 & ... & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & -a_1 & 2a_1+1 & -a_1 \\ 0 & 0 & 0 & ... & 0 & -a_1 & 2a_1+1 \end{bmatrix} \begin{bmatrix} C_1^{t+1} \\ C_2^{t+1} \\ C_3^{t+1} \\ \vdots \\ C_j^{t+1} \end{bmatrix}$$

$$= \begin{bmatrix} C_1^t + \frac{kI_{beam}\Delta x a_1}{D+k\Delta x} \\ C_2^t \\ C_3^t \\ \vdots \\ C_j^t \end{bmatrix} \tag{5.8}$$

where j is the total numbers of depth step that is used. With this general form of Equation 5.8, one can calculate all layers C^{t+1} by solving this matrix equation. Take a square pulse as an example, I_{beam} can be assumed to be a constant 1. Therefore, Equation 5.8 can now be rewritten as

$$\begin{bmatrix} (2a_1+1)-\frac{a_1 D}{D+k\Delta x} & -a_1 & 0 & ... & 0 & 0 & 0 \\ -a_1 & 2a_1+1 & -a_1 & ... & 0 & 0 & 0 \\ 0 & -a_1 & 2a_1+1 & ... & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & -a_1 & 2a_1+1 & -a_1 \\ 0 & 0 & 0 & ... & 0 & -a_1 & 2a_1+1 \end{bmatrix} \begin{bmatrix} C_1^{t+1} \\ C_2^{t+1} \\ C_3^{t+1} \\ \vdots \\ C_j^{t+1} \end{bmatrix}$$

$$= \begin{bmatrix} C_1^t + \frac{k\Delta x a_1}{D+k\Delta x} \\ C_2^t \\ C_3^t \\ \vdots \\ C_j^t \end{bmatrix}. \tag{5.9}$$

As for beam-off condition, it is defined simply as putting $I_{beam} = 0$. Now we can then get the total concentration propagating equations as

$$\begin{bmatrix} (2a_1+1)-\frac{a_1 D}{D+k\Delta x} & -a_1 & 0 & ... & 0 & 0 & 0 \\ -a_1 & 2a_1+1 & -a_1 & ... & 0 & 0 & 0 \\ 0 & -a_1 & 2a_1+1 & ... & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & -a_1 & 2a_1+1 & -a_1 \\ 0 & 0 & 0 & ... & 0 & -a_1 & 2a_1+1 \end{bmatrix} \begin{bmatrix} C_1^{t+1} \\ C_2^{t+1} \\ C_3^{t+1} \\ \vdots \\ C_j^{t+1} \end{bmatrix} = \begin{bmatrix} C_1^t \\ C_2^t \\ C_3^t \\ \vdots \\ C_j^t \end{bmatrix}. \tag{5.10}$$

One should note that the beam-on condition for square pulse using Equation 5.9 should show the same result as the analytical form shown in Equation 5.2. With pre-setting parameters, $D = 10^{-12}\,\mathrm{m^2\,s^{-1}}$ and $\tau = 10^{-6}\,\mathrm{s}$, one can get the concentration plot of gases desorption from the liquid at different layers shown in Figure 5.2 (a).

As for a supersonic beam, which is more related to our setup, the beam-on form is simply defining I_{beam} as $N_{SS,meas}$ shown in Equation 2.14 by kept calculating different I_{beam} for different time t. Using the same D and τ as mentioned in square pulse, the concentration plot is shown in Figure 5.2 (b). Python code to simulated this is shown in Appendix C.4.

5.2 Experimental Methods and Results

Ideally, one can directly use the scattering data and fit it with the concentration code (Appendix C.4) in order to extract residence time. However, scattering usually includes two processes, the impulsive scattering and thermal desorption, which the combination of these processes might increase too many fitting constants and make the fitting becomes harder.

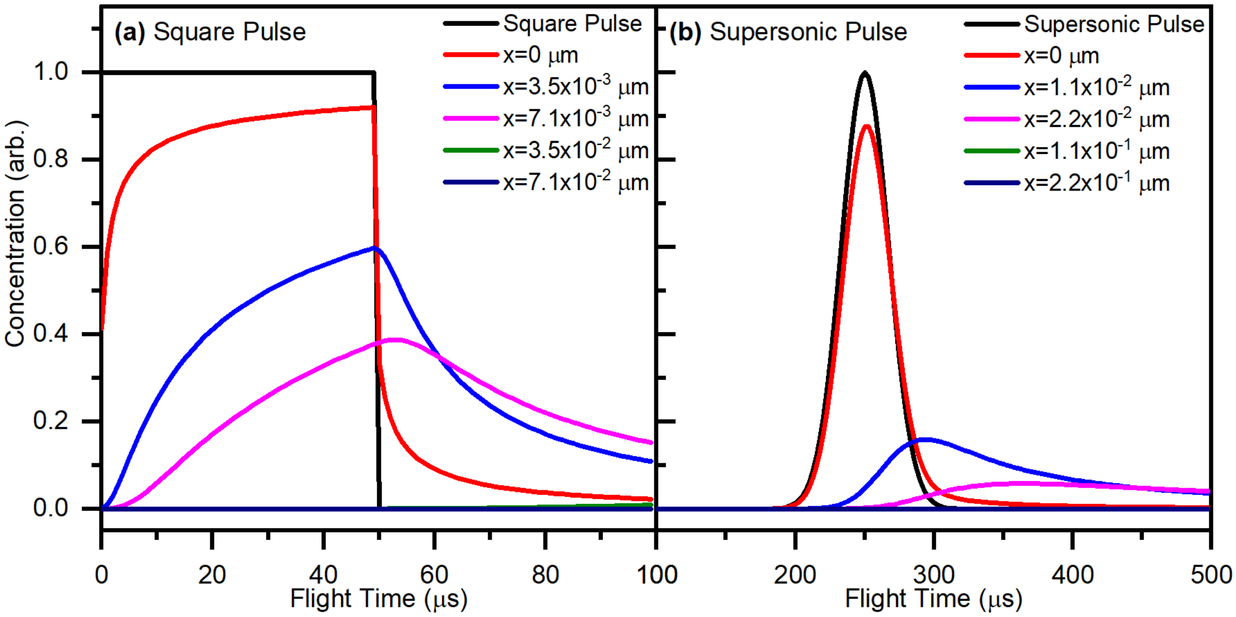


Figure 5.2: Concentration plot for a (a) square pulse and a (b) supersonic pulse. Parameters are set as $D = 10^{-12}\,\mathrm{m^2\,s^{-1}}$ and $\tau = 10^{-6}\,\mathrm{s}$. For other parameters not mentioned, they are shown in Python code in Appendix C.4.

A way to improve this is by adding a chopper into the detector axis. Scattering signal is now chopped by the chopper. The benefit of it is to separate IS and TD by the chopper. Since TD is a slower process, it should occur later on the TOF spectrum, in other words, the pulsed valve delay needs to be shorter. An experimental example is shown in Figure 5.3, where fast neon beam shooting at $\theta_i = 60°$ is data collected with outgoing angle at 30°, 60° and 90°.

In Figure 5.3, a trend of TOF with observing different parts of scattered signals are shown. At beam delay time at 40 μs for $\theta_f = 60°$, it is clear that the TOF show two clear features. The faster one is assigned to be IS, while the slower one is TD. Using this delay time as a benchmark, delay time larger than 40 μs should show mostly TD process, which from there on, we can use the concentration code convolving with $N_{MB,meas}$ and the 45 μs of opening window from the chopper to find a fit of tau.

The code for this simulation process is shown in Appendix C.5. The code, however, doesn't simulate the result well and take a long time to compile. This could be due bad coding style. Bad S/N ratio of the data and wide opening time might also make the process worse. The later one could be improved if using a smaller chopper slit, which, however, will also reduce the signal intensity.

The chopped signals are also compared with the no-chopped signals from Chapter 4. Theoretically, the chopped signal should give the same distribution as the no-chopped results

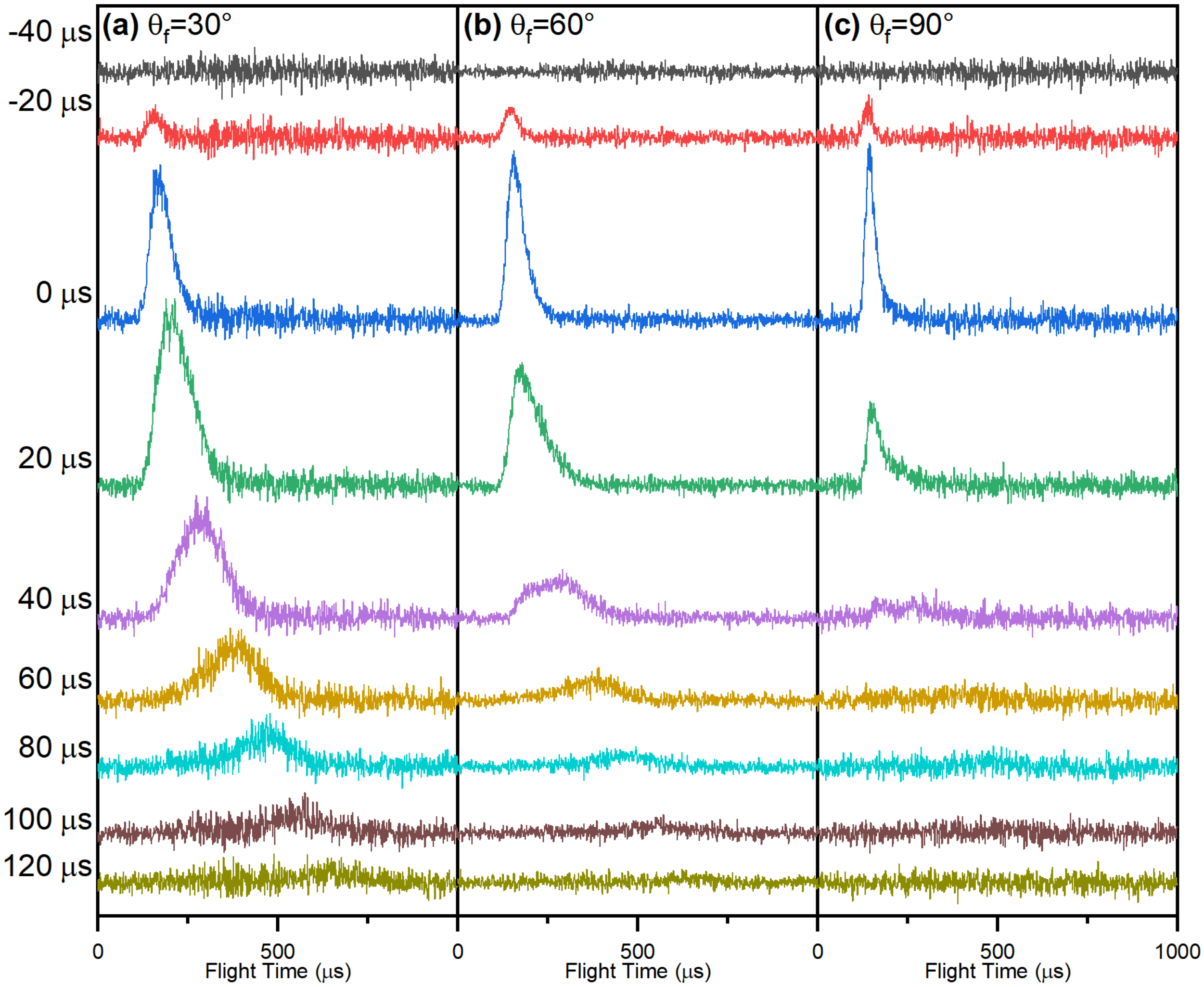


Figure 5.3: Fast neon scattering from a dodecane flat jet with chopper wheel chopping the scattered beam. $\theta_i = 60^\circ$ and θ_f are (a) 30°, (b) 60° and (c) 90°. Pulsed valve delay related to the molecular beam center is listed on the left, where larger positive numbers mean detecting slower evaporating particles, while negative numbers mean detecting signal from earlier beam.

after adding all the TOF for different delay times. The comparisons are shown in Figure 5.4 and, however, shows different TOF for the two cases. The chopped data shows much stronger IS then the no-chopped data, which indicates the current chopper and valve delay give too much weighting to the fast component. This is something that one should be aware about in the future if residence time experiments are wished to perform.

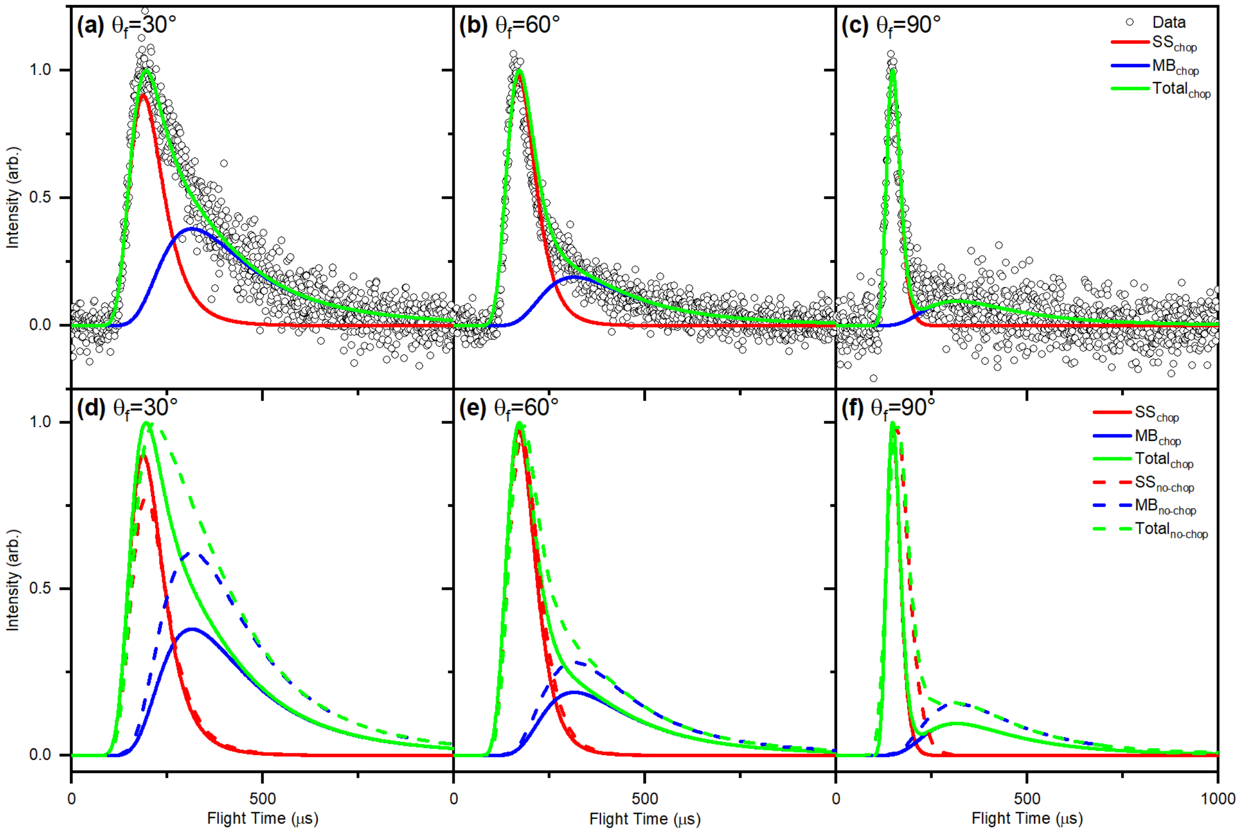


Figure 5.4: Comparison between a summed-up chopped data and no-chopped data. IS and TD fitting for summed-up chopped data from a set of delay time varied by 20 µs are shown with outgoing angles at (a) 30°, (b) 60° and (c) 90°. The fitting results for both chopper conditions are compared in lower subfigures with (d) 30°, (e) 60° and (f) 90°.

Bibliography

[1] T. J. Wallington, E. W. Kaiser, and J. T. Farrell. “Automotive fuels and internal combustion engines: a chemical perspective”. In: *Chem. Soc. Rev.* 35 (2006), pp. 335–347.

[2] A. Singh and M. Agrawal. “Acid rain and its ecological consequences”. In: *J. Environ. Biol.* 29 (2008), pp. 15–24.

[3] R. Putikam and M.-C. Lin. “A novel mechanism for the isomerization of N_2O_4 and its implication for the reaction with H_2O and acid rain formation”. In: *Int. J. Quantum Chem.* 118 (2018), e25560.

[4] M. F. Ruiz-López et al. “A New Mechanism of Acid Rain Generation from HOSO at the Air–Water Interface”. In: *J. Am. Chem. Soc.* 141 (2019), pp. 16564–16568.

[5] K. T. Valsaraj. “A Review of the Aqueous Aerosol Surface Chemistry in the Atmospheric Context”. In: *Open J. Phys. Chem.* 2 (2012), pp. 58–66.

[6] C. L. Sabine et al. “The Oceanic Sink for Anthropogenic CO_2”. In: *Science* 305 (2004), pp. 367–371.

[7] J. E. Hurst et al. “Observation of Direct Inelastic Scattering in the Presence of Trapping-Desorption Scattering: Xe on Pt(111)”. In: *Phys. Rev. Lett.* 43 (1979), pp. 1175–1177.

[8] C. B. Mullins et al. “Variation of the trapping probability of Ar on Pt(111) with kinetic energy and angle of incidence: The changing role of parallel momentum with surface temperature”. In: *Chem. Phys. Lett.* 163 (1989), pp. 111–115.

[9] E. H. Kennard, ed. New York and London: McGraw-Hill Book Company, Inc., 1938.

[10] G. Comsa and R. David. “Dynamical parameters of desorbing molecules”. In: *Surf. Sci. Rep.* 5 (1985), pp. 145–198.

[11] R. C. Cosser et al. “Angle-resolved thermal desorption of N_2 from W[310] and W[110]”. In: *Vacuum* 31 (1981), pp. 503–506.

[12] B. J. Finlayson-Pitts and Jr. J. N. Pitts, eds. San Diego: Academic Press, 2000.

[13] P. Jungwirth and D. J. Tobias. “Specific Ion Effects at the Air/Water Interface”. In: *Chem. Rev.* 106 (2006), pp. 1259–1281.

[14] M. Ahmed et al. “Molecular Properties and Chemical Transformations Near Interfaces”. In: *J. Phys. Chem. B* 125 (2021), pp. 9037–9051.

[15] G. L. Richmond. “Molecular Bonding and Interactions at Aqueous Surfaces as Probed by Vibrational Sum Frequency Spectroscopy”. In: *Chem. Rev.* 102 (2002), pp. 2693–2724.

[16] J. A. McGuire and Y. R. Shen. “Ultrafast Vibrational Dynamics at Water Interfaces”. In: *Science* 313 (2006), pp. 1945–1948.

[17] P. B. Petersen and R. J. Saykally. “On the nature of ions at the liquid water surface”. In: *Annu. Rev. Phys. Chem.* 57 (2006), pp. 333–364.

[18] S. Thürmer et al. “Photoelectron Angular Distributions from Liquid Water: Effects of Electron Scattering”. In: *Phys. Rev. Lett.* 111 (2013), p. 173005.

[19] Y.-I. Suzuki et al. “Effective attenuation length of an electron in liquid water between 10 and 600 eV”. In: *Phys. Rev. E* 90 (2014), p. 010302.

[20] R. Signorell. “Electron Scattering in Liquid Water and Amorphous Ice: A Striking Resemblance”. In: *Phys. Rev. Lett.* 124 (2020), p. 205501.

[21] L. Artiglia et al. “A surface-stabilized ozonide triggers bromide oxidation at the aqueous solution-vapour interface”. In: *Nat. Commun.* 8 (2017), p. 700.

[22] S. Ghosal et al. “Electron Spectroscopy of aqueous solution interfaces reveals surface enhancement of halides”. In: *Science* 307 (2005), pp. 563–566.

[23] K. J. Kappes et al. “Chemistry and Photochemistry of Pyruvic Acid at the Air–Water Interface”. In: *J. Phys. Chem. A* 125 (2021), pp. 1036–1049.

[24] Y. Li, X. Yan, and R. G. Cooks. “The Role of the Interface in Thin Film and Droplet Accelerated Reactions Studied by Competitive Substituent Effects”. In: *Angew. Chem. Int. Ed.* 55 (2016), pp. 3433–3437.

[25] I. Nam et al. “Abiotic production of sugar phosphates and uridine ribonudeoside in aqueous microdroplets”. In: *Proc. Natl. Acad. Sci. U.S.A.* 114 (2017), pp. 12396–12400.

[26] M. E. Diveky et al. “Shining New Light on the Kinetics of Water Uptake by Organic Aerosol Particles”. In: *J. Phys. Chem. A* 125 (2021), pp. 3528–3548.

[27] D. M. Neumark et al. “Molecular beam studies of the F+H_2 reaction”. In: *J. Chem. Phys.* 82 (1985), pp. 3045–3066.

[28] R. E. Continetti, B. A. Balko, and Y. T. Lee. “Crossed molecular beams study of the reaction $D+H_2 \rightarrow DH+H$ at collision energies of 0.53 and 1.01 eV”. In: *J. Chem. Phys.* 93 (1990), pp. 5719–5740.

[29] W. H. Weinberg. “Molecular beam scattering from solid surfaces”. In: *Adv. Colloid Interface Sci.* 4 (1975), pp. 301–347.

[30] M. J. Cardillo. “Molecular beam scattering from solid surfaces: He diffraction, hyperthermal scattering, and surface dynamics”. In: *Surf. Sci.* 299 (1994), pp. 277–283.

[31] G. M. nathanson. “Molecular beam studies of gas-liquid interfaces”. In: *Annu. Rev. Phys. Chem.* 55 (2004), pp. 231–255.

[32] M. E. Saecher et al. “Molecular Beam Scattering from Liquid Surfaces”. In: *Science* 252 (1991), pp. 1421–1424.

[33] S. L. Lednovich and J. B. Fenn. “Absolute evaporation rates for some polar and nonpolar liquids”. In: *AlChE J.* 23 (1977), pp. 454–459.

[34] H. Siegbahn. “Electron spectroscopy for chemical analysis of liquids and solutions”. In: *J. Phys. Chem.* 89 (1985), pp. 897–909.

[35] M. E. King et al. “Probing the microscopic corrugation of liquid surfaces with gas-liquid collisions”. In: *Phys. Rev. Lett.* 70 (1993), pp. 1026–1029.

[36] G. M. Nathanson et al. “Dynamics and Kinetics at the Gas-Liquid Interface”. In: *J. Phys. Chem.* 100 (1996), pp. 13007–13020.

[37] B. Wu et al. “Scattering Dynamics of Hyperthermal Oxygen Atoms on Ionic Liquid Surfaces: [emim][NTf$_2$] and [C$_{12}$mim][NTf$_2$]”. In: *J. Phys. Chem. C* 114 (2010), pp. 4015–4027.

[38] S. M. Brastad and G. M. Nathanson. “Molecular beam studies of HCl dissolution and dissociation in cold salty water”. In: *Phys. Chem. Chem. Phys.* 13 (2011), pp. 8284–8295.

[39] P. D. Lane et al. “Inelastic Scattering of CN Radicals at the Gas-Liquid Interface Probed by Frequency-Modulated Absorption Spectroscopy”. In: *J. Phys. Chem. C* 124 (2020), pp. 16439–16448.

[40] M. Faubel, S. Schlemmer, and J. P. Toennies. “A molecular beam study of the evaporation of water from a liquid jet”. In: *Z. Phys. D: At., Mol. Clusters* 10 (1988), pp. 269–277.

[41] M. Faubel and Th. Kisters. “Non-equilibrium molecular evaporation of carboxylic acid dimers”. In: *Nature* 339 (1989), pp. 527–529.

[42] D. K. Lancaster et al. “Inert Gas Scattering from Liquid Hydrocarbon Microjets”. In: *J. Phys. Chem. Lett.* 4 (2013), pp. 3045–3049.

[43] J. A. Faust, T. B. Sobyra, and G. M. Nathanson. “Gas-Microjet Reactive Scattering: Collisions of HCl and DCl with Cool Salty Water”. In: *J. Phys. Chem. Lett.* 7 (2016), pp. 730–735.

[44] M. A. Shaloski et al. “Reactions of N_2O_5 with Salty and Surfactant-Coated Glycerol: Interfacial Conversion of Br^- to Br_2 Mediated by Alkylammonium Cations”. In: *J. Phys. Chem. A* 121 (2017), pp. 3708–3719.

[45] J. R. Gord et al. “Control of Interfacial Cl_2 and N_2O_5 Reactivity by a Zwitterionic Phospholipid in Comparison with Ionic and Uncharged Surfactants”. In: *J. Phys. Chem. A* 122 (2018), pp. 6593–6604.

[46] T. B. Sobyra, M. P. Melvin, and G. M. Nathanson. “Liquid Microjet Measurements of the Entry of Organic Acids and Bases into Salty Water”. In: *J. Phys. Chem. C* 121 (2017), pp. 20911–20924.

[47] J. W. M. Bush and A. E. Hasha. “On the collision of laminar jets: fluid chains and fishbones”. In: *J. Fluid Mech.* 511 (2004), pp. 285–310.

[48] M. Ekimova et al. “A liquid flatjet system for solution phase soft-x-ray spectroscopy”. In: *Struct. Dyn.* 2 (2015), p. 054301.

[49] T. T. Luu et al. “Extreme-ultraviolet high-harmonic generation in liquids”. In: *Nat. Commun.* 9 (2018), p. 3723.

[50] S. Menzi et al. “Generation and simple characterization of flat, liquid jets”. In: *Rev. Sci. Instrum.* 91 (2020), p. 105109.

[51] A. N. Tcypkin et al. “Flat liquid jet as a highly efficient source of terahertz radiation”. In: *Opt. Express* 27 (2019), pp. 15485–15494.

[52] S. Malerz et al. “A setup for studies of photoelectron circular dichroism from chiral molecules in aqueous solution”. In: *Rev. Sci. Instrum.* 93 (2022), p. 015101.

[53] J. A. Faust and G. M. Nathanson. “Microjets and coated wheels: versatile tools for exploring collisions and reactions at gas-liquid interfaces”. In: *Chem. Soc. Rev.* 45 (2016), pp. 3609–3620.

[54] K. R. Wilson et al. “Investigation of volatile liquid surfaces by synchrotron x-ray spectroscopy of liquid microjets”. In: *Rev. Sci. Instrum.* 75 (2004), pp. 725–736.

[55] C. D. Cappa et al. “Isotope Fractionation of Water during Evaporation without Condensation”. In: *J. Phys. Chem. B* 109 (2005), pp. 24391–24400.

[56] K. W. Kolasinski, ed. United Kingdom: John Wiley & Sons, Ltd, 2012.

[57] R. Hołyst, M. Litniewski, and D. Jakubczyk. “A molecular dynamics test of the Hertz-Knudsen equation for evaporating liquids”. In: *Soft Matter* 11 (2015), pp. 7201–7206.

[58] H. Sippola and P. Taskinen. “Activity of Supercooled Water on the Ice Curve and Other Thermodynamic Properties of Liquid Water up to the Boiling Point at Standard Pressure”. In: *J. Chem. Eng. Data* 63 (2018), 2986-2998.

[59] D. Schlesinger et al. “Evaporative cooling of microscopic water droplets *in vacuo*: Molecular dynamics simulations and kinetic gas theory”. In: *J. Chem. Phys.* 144 (2016), p. 124502.

[60] K. Ando, M. Arakawa, and A. Terasaki. “Freezing of micrometer-sized liquid droplets of pure water evaporatively cooled in a vacuum”. In: *Phys. Chem. Chem. Phys.* 20 (2018), pp. 28435–28444.

[61] Y.-P. Chang et al. "Temperature measurements of liquid flat jets in vacuum". In: *Struct. Dyn.* 9 (2022), p. 014901.

[62] L. B. Loeb, ed. New York: Dover, 1961.

[63] J. D. Koralek et al. "Generation and characterization of ultrathin free-flowing liquid sheets". In: *Nat. Commun.* 9 (2018), p. 1353.

[64] M. Ryazanov and D. J. Nesbitt. "Quantum-state-resolved studies of aqueous evaporation dynamics: NO ejection from a liquid water microjet". In: *J. Chem. Phys.* 150 (2019), p. 044201.

[65] R. G. J. Fraser and L. F. Broadway. "Experiments on molecular scattering gases. I.—The method of crossed molecular beams". In: *Proc. R. Soc. Lond. A* 141 (1933), pp. 626–633.

[66] T. H. Bull and P. B. Moon. "A mechanical method for the activation of fast reactions". In: *Discuss. Faraday Soc.* 17 (1954), pp. 54–57.

[67] E. H. Taylor and S. Datz. "Study of Chemical Reaction Mechanisms with Molecular Beams. The Reaction of K with HBr". In: *J. Chem. Phys.* 23 (1955), pp. 1711–1718.

[68] Y. T. Lee et al. "Molecular Beam Reactive Scattering Apparatus with Electron Bombardment Detector". In: *Rev. Sci. Instrum.* 40 (1969), pp. 1402–1408.

[69] R. K. Sparks. "Crossed Beam Studies of Full and Half Collisions". PhD thesis. University of California, Berkeley, 1974.

[70] J. C. Robinson. "Photofragment Translational Spectroscopy Studies of Unsaturated Hydrocarbons". PhD thesis. University of California, Berkeley, 2002.

[71] D. Irimia et al. "A short pulse (7 μs FWHM) and high repetition rate (dc-5kHz) cantilever piezovalve for pulsed atomic and molecular beams". In: *Rev. Sci. Instrum.* 80 (2009), p. 113303.

[72] P. Chen et al. "Flash Pyrolytic Production of Rotationally Cold Free Radicals in a Supersonic Jet. Resonant Multiphoton Spectrum of the $3p^2A_2'' \leftarrow X^2A_2''$ Origin Band of CH_3". In: *J. Phys. Chem.* 90 (1986), pp. 2319–2321.

[73] D. W. Kohn, H. Clauberg, and P. Chen. "Flash pyrolysis nozzle for generation of radicals in a supersonic jet expansion". In: *Rev. Sci. Instrum.* 63 (1992), pp. 4003–4005.

[74] N. C. Cole-Filipiak. "Production and Photodissociation of neutral Free Radicals". PhD thesis. University of California, Berkeley, 2015.

[75] R. A. Haefer. "Cryogenic vacuum techniques". In: *J. Phys. E: Sci. Instrum.* 14 (1981), pp. 273–288.

[76] W. Umrath, ed. Cologne, Germany: Oerlikon Leybold Vacuum, 2007.

[77] K. Kuhnke et al. "High efficiency molecular-beam ionization detector with short ionization region". In: *Rev. Sci. Instrum.* 65 (1994), pp. 3458–3465.

[78] *Axial molecular beam ionizer user manual*. English. Extrel. 2004. 96 pp.

[79] N. R. Daly. "Scintillation Type Mass Spectrometer Ion Detector". In: *Rev. Sci. Instrum.* 31 (1960), pp. 264–267.

[80] M. J. Shapero. "Photodissociation Studies of Combustion Relevant Radical Molecules". PhD thesis. University of California, Berkeley, 2017.

[81] D. R. Herschbach. "Molecular Dynamics of Elementary Chemical Reactions (Nobel Lecture)". In: *Angew. Chem. Int. Ed. Engl.* 26 (1986), pp. 1221–1243.

[82] J. B. Anderson, R. P. Andres, and J. B. Fenn. "8. Supersonic nozzle beams". In: *Advances in chemical physics*. Ed. by J. Ross. New York-London-Sydney: John Wiley & Sons, Inc., 1966, pp. 275–317.

[83] L. Dunoyer. "Sur la réalisation d'un rayonnement matériel d'origine purement thermique. Cinétique expérimentale". In: *Le Radium* 8 (1911), pp. 142–146.

[84] O. Stern. "Eine direkte Messung der thermischen Molekulargeschwindigkeit". In: *Z. Phys.* 2 (1920), pp. 49–56.

[85] O. Stern. "Nachtrag zu meiner Arbeit: „Eine direkte Messung der thermischen Molekulargeschwindigkeit"". In: *Z. Phys.* 3 (1920), pp. 417–421.

[86] W. Gerlach and O. Stern. "Der experimentelle Nachweis der Richtungsquantelung im Magnetfeld". In: *Z. Phys.* 9 (1922), pp. 349–352.

[87] N. F. Ramsey. "1. Thermal beam sources". In: *Advances in chemical physics*. Ed. by F. B. Dunning and R. G. Hulet. 29B vols. London: Academic Press, Inc., 1996, pp. 1–20.

[88] A. Kantrowitz and J. Grey. "A High Intensity Source for the Molecular Beam. Part I. Theoretical". In: *Rev. Sci. Instrum.* 22 (1951), pp. 328–332.

[89] M. D. Morse. "2. Supersonic beam sources". In: *Advances in chemical physics*. Ed. by F. B. Dunning and R. G. Hulet. 29B vols. London: Academic Press, Inc., 1996, pp. 21–47.

[90] N. F. Ramsey. "Molecular beams: our legacy from Otto Stern". In: *Z. Phys. D: At., Mol. Clusters* 10 (1988), pp. 121–125.

[91] J. R. Dunning et al. "Velocity of Slow Neutrons by Mechanical Velocity Selector". In: *Phys. Rev.* 48 (1935), p. 704.

[92] J. G. Dash and H. S. Sommers. "A High Transmission Slow Neutron Velocity Selector". In: *Rev. Sci. Instrum.* 24 (1953), pp. 91–96.

[93] H. U. Hostettler and R. B. Bernstein. "Improved Slotted Disk Type Velocity Selector for Molecular Beams". In: *Rev. Sci. Instrum.* 31 (1960), pp. 872–877.

[94] D. R. Miller. "2. Free Jet Sources". In: *Atomic and Molecular Beam Methods*. Ed. by G. Scoles. 1 vols. New York-Oxford: Oxford University Press, 1988, pp. 14–53.

[95] I. S. Shpigel. “Quick-operating electrodynamic vacuum valve”. In: *Instrum. Exp. Tech.* 1 (1959), p. 156.

[96] B. Gorowitz, K. Moses, and P. Gloersen. “Magnetically Driven Fast-Acting Valve for Gas Injection into High Vacua”. In: *Rev. Sci. Instrum.* 31 (1960), pp. 146–148.

[97] N. Inoue and T. Uchida. “New Type of High Speed Valve”. In: *Rev. Sci. Instrum.* 39 (1968), pp. 1461–1464.

[98] W. R. Gentry and C. F. Giese. “Ten-microsecond pulsed molecular beam source and a fast ionization detector”. In: *Rev. Sci. Instrum.* 49 (1978), pp. 595–600.

[99] M. R. Adriaens, W. Allison, and B. Feuerbacher. “A pulsed molecular beam source”. In: *J. Phys. E: Sci. Instrum.* 14 (1981), pp. 1375–1377.

[100] L. Abad et al. “Performance of a solenoid-driven pulsed molecular-beam source”. In: *Rev. Sci. Instrum.* 66 (1995), pp. 3826–3832.

[101] U. Even. “Pulsed Supersonic Beams from High Pressure Source: Simulation Results and Experimental Measurements”. In: *Adv. Chem.* 2014 (2014), p. 636042.

[102] S. J. Goncher. “Studies of Photofragment Translational Spectroscopy”. PhD thesis. University of California, Berkeley, 2008.

[103] B. Negru. “Photodissociation Dynamics of Neutral Free Radicals”. PhD thesis. University of California, Berkeley, 2012.

[104] E. N. Sullivan. “Investigations of the two- and three-body photodissociation processes of free radicals using the fast radical beam machine”. PhD thesis. University of California, Berkeley, 2020.

[105] *Amsterdam cantilever piezo valve ACPV2/ACPV3 and Electronic driver unit EDU5-200V User manual.* English. MassSpecpecD BV. 2021. 18 pp.

[106] I. A. Ramphal. “Molecular Beam Photodissociation and Liquid Surface Scattering Dynamics”. PhD thesis. University of California, Berkeley, 2021.

[107] B. Ha, D. P. DePonte, and J. G. Santiago. “Device design and flow scaling for liquid sheet jets”. In: *Phys. Rev. Fluids* 3 (2018), p. 114202.

[108] G. Galinis et al. “Micrometer-thickness liquid sheet jets flowing in vacuum”. In: *Rev. Sci. Instrum.* 88 (2017), p. 083117.

[109] A. Charvat et al. “New design for a time-of-flight mass spectrometer with a liquid beam laser desorption ion source for the analysis of biomolecules”. In: *Rev. Sci. Instrum.* 75 (2004), pp. 1209–1218.

[110] D. J. Krajnovich. “Molecular Beam Studies of Photodissociation Reactions”. PhD thesis. University of California, Berkeley, 1983.

[111] W. C. Wiley and I. H. McLaren. “Time-of-Flight Mass Spectrometer with Improved Resolution”. In: *Rev. Sci. Instrum.* 26 (1955), pp. 1150–1157.

[112] B. R. Ringeisen. “Collisions and Reactions of HCl and HBr with Liquid Glycerol”. PhD thesis. University of Wisconsin-Madison, 2000.

[113] T. Edwards and L. Q. Maurice. “Surrogate Mixtures to Represent Complex Aviation and Rocket Fuels”. In: *J. Propuls. Power* 17 (2001), pp. 461–466.

[114] V. Bochko and Z. K. Silagadze. “On the electrostatic potential and electric field of a uniformly charged disk”. In: *Eur. J. Phys.* 41 (2020), p. 045201.

[115] E. M. Purcell and D. J. Morin, eds. New York: Cambridge University Press, 2013.

[116] Z. R. Kann and J. L. Skinner. “Sub- and super-Maxwellian evaporation of simple gases from liquid water”. In: *J. Chem. Phys.* 144 (2016), p. 154701.

[117] G. A. McD. Cummings and A. R. Ubbelohde. “Collision Diameters of Flexible Hydrocarbon Molecules in the Vapour Phase: The ”Hydrogen Effect.”” In: *J. Chem. Soc.* (1953), pp. 3751–3755.

[118] F. M. Mourits and F. H. A. Rummens. “A critical evaluation of Lennard–Jones and Stockmayer potential parameters and of some correlation methods”. In: *Can. J. Chem.* 55 (1977), pp. 3007–3020.

[119] C.-k. Chen, M. Banaszak, and M. Radosz. “RETURN TO ISSUEPREVARTICLE Statistical Associating Fluid Theory Equation of State with Lennard-Jones Reference Applied to Pure and Binary *n*-Alkane Systems”. In: *J. Phys. Chem. B* 102 (1998), pp. 2427–2431.

[120] M. Head-Gordon and J. C. Tully. “On the nature of trapping and desorption at high surface temperatrues. Theory and experiments for the Ar-Pt(111) system”. In: *J. Chem. Phys.* 94 (1991), pp. 1516–1527.

[121] M. A. Tesa-Serrate et al. “Atomic and Molecular Collisions at Liquid Surfaces”. In: *Annu. Rev. Phys. Chem.* 67 (2016), pp. 515–540.

[122] N. Andric and P. Jenny. “Molecular dynamics investigation of energy transfer during gas-surface collisions”. In: *Phys. Fluids* 30 (2018), p. 077104.

[123] W. A. Alexander et al. “Kinematics and dynamics of atomic-beam scattering on liquid and self-assembled monolayer surfaces”. In: *Faraday Discuss.* 157 (2012), pp. 355–374.

[124] C. T. Rettner and M. N. R. Ashfold, eds. Cambridge: Royal Society of Chemistry, 1991.

[125] A. H. Muenter. “Reactions of DCl With Pure and Salty Glycerol”. PhD thesis. University of Wisconsin-Madison, 2004.

[126] J. R. Morris et al. “Molecular Beam Scattering from Supercooled Sulfuric Acid: Collisions of HCl, HBr, and HNO_3 with 70 wt% D_2SO_4”. In: *J. Phys. Chem. A* 104 (2000), pp. 6738–6751.

[127] B. R. Ringeisen, A. H. Muenter, and G. M. Nathanson. “Collisions of HCl, DCl, and HBr with Liquid Glycerol: Gas Uptake, D → H Exchange, and Solution Thermodynamics”. In: *J. Phys. Chem. B* 106 (2002), pp. 4988–4998.

[128] R. Celotta and T. Lucatorto, eds. *Experimental methods in the physical sciences*. 29B vols. London: Academic Press, Inc., 1996.

Appendix A

Codes to Estimate Jet Properties

The codes shown in this section are written in Mathematica version 13.1. Liquid system of water cylindrical jet and flat jet are studied as an example. One can use these codes to estimate other liquid systems by changing the parameters. Following sections includes codes to calculate vapor pressure of the liquid at different temperatures, jet temperature along the jet axis and number of collisions for the outgoing particles. The order of these codes is also the same order of thought flow for one to estimate the jet properties.

In the first section, codes to calculate vapor pressures at different temperatures are shown. This is the base step one should consider before calculating jet temperature and number of collisions. The second part is the jet temperature section, which allows one to estimate T_{jet} cooled down by evaporative cooling at different jet height, where the most important position might be the detector viewing height. Finally, the third part shows how to calculate number of collisions which gives estimation whether the outgoing particles are mostly nascent or not.

A.1 Vapor Pressure and Viscosity

This code uses equations in Sippola's paper [58] and Clausius-Clapeyron relation to estimate the vapor pressure of water. Sippola's equations, apparently, can only be used for water system, while the Clausius-Clapeyron relation can be used for all kinds of gases which act as ideal gas.

Second part of the code calculate water viscosity at different temperatures, which is shown as an extra information. This value is useful when the jet is pre-cooled and can provide rough estimation on how the HPLC pump backing pressure being increased due to the increasing viscosity.

In[1]:=

```
(*Mathematica Code_Vapor Pressure_Water*)
(*Outline*)
(*I.   Vapor Pressure(P_vap)*)
(*II.  Viscosity(η)*)
```

In[2]:=

```
Needs["Notation`"];
Symbolize[ParsedBoxWrapper[SubscriptBox["_", "_"]]]
ClearAll["Global`*"]
```

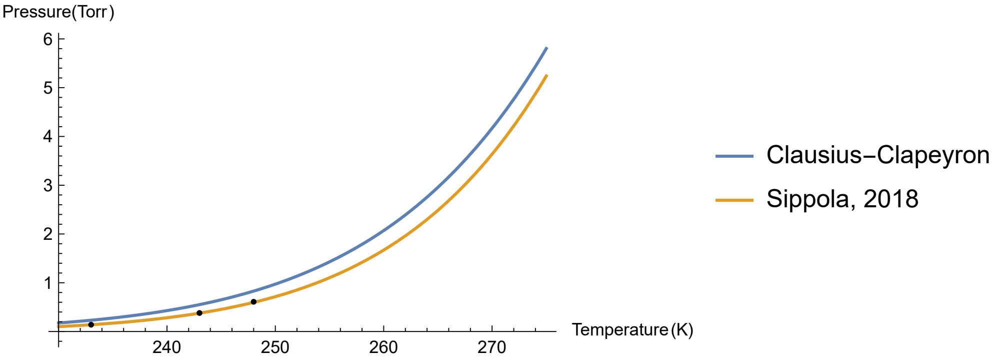
In[5]:=

```
(*-----------------------I. Vapor Pressure-----------------------*)
(*---------------------------Parameters---------------------------*)
M = 18;(*g/mol*)(*for water: 18*)
ΔH_vap = 2270 * M;(*J/mol*)(*for water: 2270*M*)
T_set = 265;(*K*)
(*----------------------------------------------------------------*)

(*---------------------------Equations--------------------------*)
P_CC[T_] := 17.5424 Exp[(-ΔH_vap/8.314) (1/T - 1/293.15)];(*Torr*)


K[T_] := Exp[-421105.608/T + 13205.3106 - 2364.09638 Log[T] +
    5.92146122 T - 0.0027787306 T^2 + 14091079/T^2];(*equilibrium constant*)
P_ice,Sippola[T_] :=
  Exp[9.550426 - 5723.265/T + 3.53068 Log[T] - 0.00728332 T] * 0.00750061683;(*Torr*)
P_water,Sippola[T_] := P_ice,Sippola[T]/K[T];(*Torr*)
(*----------------------------------------------------------------*)
(*---------------------------Plotting---------------------------*)
Print[Style["Supercooled Water Vapor Pressure", 18, Bold, Blue]]
Print["At T_set = ", T_set, " K, P_CC = ", ScientificForm[P_CC[T_set], 3],
  " Torr; P_Sippola = ", ScientificForm[P_water,Sippola[T_set], 3], " Torr"];
Print["At T = ", 248, " K, P_CC = ", ScientificForm[P_CC[248], 3],
  " Torr; P_Sippola = ", ScientificForm[P_water,Sippola[248], 3],
  " Torr; P_Engineering ToolBox = 6.1*10^-1 Torr"];
Print["At T = ", 243, " K, P_CC = ", ScientificForm[P_CC[243], 3],
  " Torr; PSippola = ", ScientificForm[Pwater,Sippola[243], 3],
  " Torr; PEngineering ToolBox = 3.8*10^-1 Torr"];
Print["At T = ", 233, " K, PCC = ", ScientificForm[PCC[233], 3],
  " Torr; PSippola = ", ScientificForm[Pwater,Sippola[233], 3],
  " Torr; PEngineering ToolBox = 1.4*10^-1 Torr"];
Plot[{PCC[T], Pwater,Sippola[T]}, {T, 230, 275},
 Epilog → {Point[{233, 0.14}], Point[{243, 0.38}], Point[{248, 0.61}]},
 AxesLabel → {"Temperature(K)", "Pressure(Torr)"},
 PlotLegends → {"Clausius-Clapeyron", "Sippola, 2018"}]
(*Data points from Engineering Toolbox is from online source. Website: https://
 www.engineeringtoolbox.com/water-vapor-saturation-pressure-d_599.html*)
(*-----------------------------------------------------------------*)
```

Supercooled Water Vapor Pressure

```
At Tset = 265 K, PCC = 2.96 Torr; PSippola = 2.48 Torr

At T = 248 K, PCC = 8.29×10^-1 Torr; PSippola =
 5.98×10^-1 Torr; PEngineering ToolBox = 6.1*10^-1 Torr

At T = 243 K, PCC = 5.51×10^-1 Torr; PSippola =
 3.76×10^-1 Torr; PEngineering ToolBox = 3.8*10^-1 Torr

At T = 233 K, PCC = 2.31×10^-1 Torr; PSippola =
 1.39×10^-1 Torr; PEngineering ToolBox = 1.4*10^-1 Torr
```

Out[17]=



In[18]:=

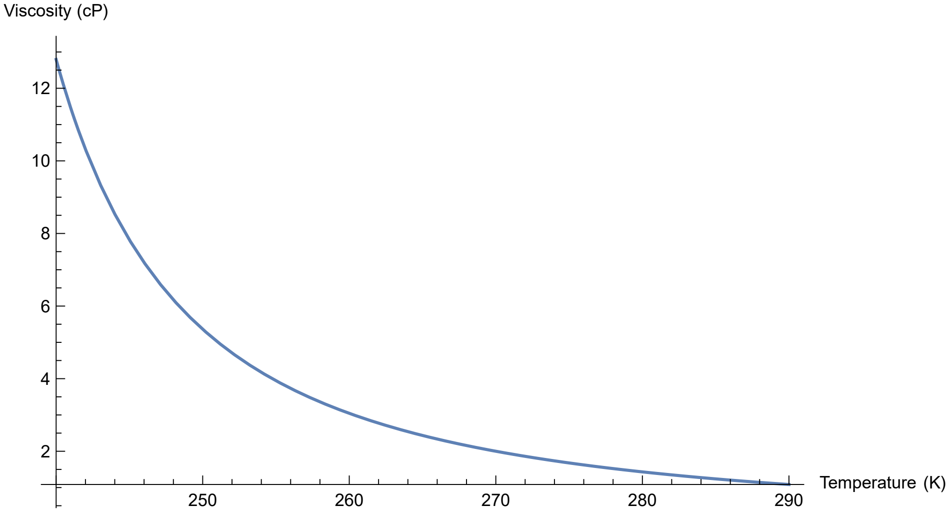
```
(*------------------------II. Viscosity------------------------*)
(*--------------------------Parameters-------------------------*)
Ts = 225.66; (*K*)
η0 = 1.3788×10^-4; (*Pa*s*) (*Caupin's 2015 paper, power law*)
γ = 1.6438;
```

```
Cm = 1.2;(*Morrow's 2012 paper, eq 11*)(*volume fraction*)

Tset = 293;(*K*)
xvolume = 0.614;(*volume fraction*)
(*------------------------------------------------------------------*)

(*----------------------------Equations---------------------------*)
ηwater[T_] := η0 (T/Ts - 1)^-γ 10^3;(*cP*)
(*Caupin's 2015 paper, power law*)(*239.15-373.15 K*)
(*------------------------------------------------------------------*)

(*----------------------------Plotting----------------------------*)
Print[Style["Water Viscosity", 18, Bold, Blue]]
Print["At T = 300 K, ηwater = ", ScientificForm[ηwater[300], 3], " cP"];
Print["At T = ", Tset, " K, ηwater = ", ScientificForm[ηwater[Tset], 3], " cP"];
Plot[ηwater[T], {T, 240, 290}, AxesLabel → {"Temperature (K)", "Viscosity (cP)"}]
(*------------------------------------------------------------------*)
```

Water Viscosity

```
At T = 300 K, ηwater = 8.55 × 10^-1 cP
At T = 293 K, ηwater = 1.01 cP
```

Out[28]=



A.2 Jet Temperature

This code estimate both cylindrical jet and flat jet temperatures with the effect of evaporative cooling. Starting from cylindrical jet, the temperature gradient along the jet axis, dT/dz, can be estimated by different methods, which are different by the definition of radial evaporative ablation rate, dr/dt. dT/dz is defined in Equation 1.2 and is copied below as

$$\frac{dT}{dz} = -2\frac{\frac{dr}{dt}}{v_{jet}}\frac{\Delta H_{vap}}{C_p}\frac{1}{r_{jet}}.$$

Two methods have been used to calculate dr/dt and compared in the code. The first method to calculate dr/dt is by using theory of evaporation ($evap$). With different definition of evaporating rate by Faubel's [40] and Saykally's [17, 55] groups, the equations can be written as

$$\left(\frac{dr}{dt}\right)_{evap,Faubel} = \sqrt{\frac{8k_BT}{\pi m_{gas}}}\frac{\rho_{gas}}{\rho_{liq}},$$

$$\left(\frac{dr}{dt}\right)_{evap,Saykally} = \phi_e\frac{P_{vap}}{\sqrt{2\pi m_{gas}k_BT}}\frac{1}{\rho_{liq}}$$

where the first and second equations are same as Equation 1.3 and 1.5. The other way to calculate dr/dt is the reference method (ref), which use a reference value of $(dr/dt)_{T_{ref}}$ at reference temperature T_{ref}. With Clausius-Clapeyron relation, $(dr/dt)_{ref}$ at any given temperature T can be written as

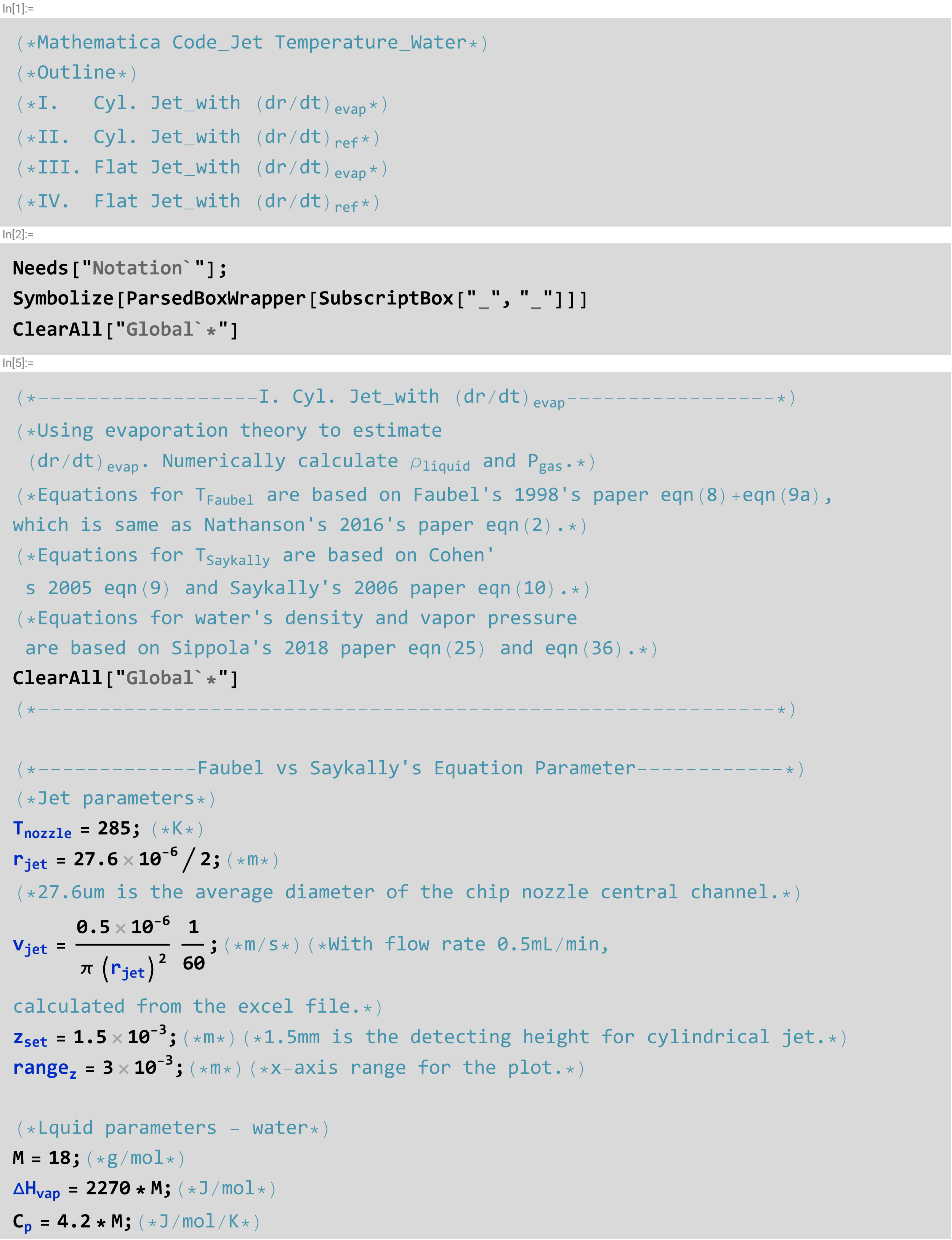
$$\left(\frac{dr}{dt}\right)_{ref} = \left(\frac{dr}{dt}\right)_{T_{ref}}\sqrt{\frac{T}{T_{ref}}}exp\left[-\frac{\Delta H_{vap}}{k_BT_{ref}}\frac{T_{ref}-T}{T}\right]$$

which has the same form of Equation 1.7. Based on different reference values of $(dr/dt)_{T_{ref}}$ and T_{ref}, Faubel's [40] and Saykally's [54] groups again provide different values of $(dr/dt)_{ref}$. Therefore, after plugging the above dr/dt equations back to Equation 1.2, there are four different forms of dT/dz can be listed. They are calculated and compared in this Mathematica code.

Besides cylindrical jet, temperature of flat jet is also estimated. With the mentioned dT/dz, flat jet's dT/dz can be calculated by adding ratio, R_α, in the the dT/dz equation as

$$\left(\frac{dT}{dz}\right)_{flat} = R_\alpha\left(\frac{dT}{dz}\right)_{cyl}$$

where R_α is defined in Equation 1.10. Estimated by four different forms of $(dT/dz)_{cyl}$ for cylindrical jet, four $(dT/dz)_{flat}$ for flat jet are also calculated and compared in the code.



In[1]:=

```
(*Mathematica Code_Jet Temperature_Water*)
(*Outline*)
(*I.   Cyl. Jet_with (dr/dt)_evap*)
(*II.  Cyl. Jet_with (dr/dt)_ref*)
(*III. Flat Jet_with (dr/dt)_evap*)
(*IV.  Flat Jet_with (dr/dt)_ref*)
```

In[2]:=

```
Needs["Notation`"];
Symbolize[ParsedBoxWrapper[SubscriptBox["_", "_"]]]
ClearAll["Global`*"]
```

In[5]:=

```
(*------------------I. Cyl. Jet_with (dr/dt)_evap-----------------*)
(*Using evaporation theory to estimate
 (dr/dt)_evap. Numerically calculate ρ_liquid and P_gas.*)
(*Equations for T_Faubel are based on Faubel's 1998's paper eqn(8)+eqn(9a),
which is same as Nathanson's 2016's paper eqn(2).*)
(*Equations for T_Saykally are based on Cohen'
 s 2005 eqn(9) and Saykally's 2006 paper eqn(10).*)
(*Equations for water's density and vapor pressure
 are based on Sippola's 2018 paper eqn(25) and eqn(36).*)
ClearAll["Global`*"]
(*--------------------------------------------------------------*)

(*-------------Faubel vs Saykally's Equation Parameter------------*)
(*Jet parameters*)
T_nozzle = 285; (*K*)
r_jet = 27.6×10^-6/2; (*m*)
(*27.6um is the average diameter of the chip nozzle central channel.*)
v_jet = 0.5×10^-6/(π (r_jet)^2) 1/60; (*m/s*) (*With flow rate 0.5mL/min,
calculated from the excel file.*)
z_set = 1.5×10^-3; (*m*) (*1.5mm is the detecting height for cylindrical jet.*)
range_z = 3×10^-3; (*m*) (*x-axis range for the plot.*)

(*Lquid parameters - water*)
M = 18; (*g/mol*)
ΔH_vap = 2270*M; (*J/mol*)
C_p = 4.2*M; (*J/mol/K*)
```

```
R = 8.314;(*J/K/mol*)

(*Lquid parameters in Saykally's eqn – water*)
ϕe = 0.35;(*0.35, for r<5um jet, Saykally's 2005 paper,
experimentally determined parameter for water*)
          (*0.62, for droplet,
Saykally's 2006 paper, evaporation coefficient*)
(*-----------------------------------------------------------*)

(*------------Equations for Density and Vapor Pressure------------*)
```

$$\rho_{\text{liq,water}}[T\_] := 1.007853\ \text{Exp}\left[-228\left(3.9744\times10^{-4}+1.6785\times10^{-3}\left(\frac{T}{228}-1\right)+2\left(-7.8165^{-4}\right)\sqrt{\frac{T}{228}-1}\right)\right]\frac{10^{-3}}{10^{-6}};$$

(*kg/m^3*) (*Sippola's paper eqn(36)*)

$$K[T\_] := \text{Exp}\left[\frac{-421105.608}{T}+13205.3106-2364.09638\,\text{Log}[T]+5.92146122\,T-0.0027787306\,T^2+\frac{14091079}{T^2}\right];$$

(*Sippola's paper eqn(35)*)

$$P_{\text{ice,Sippola}}[T\_] := \text{Exp}\left[9.550426-\frac{5723.265}{T}+3.53068\,\text{Log}[T]-0.00728332\,T\right]*0.00750061683;$$

(*Torr*) (*Sippola's paper eqn(41)*)

$$P_{\text{water,Sippola}}[T\_] := \frac{P_{\text{ice,Sippola}}[T]}{K[T]};$$

(*Torr*) (*Sippola's paper eqn(25)*)

$$z = \text{Table}\left[i\ ,\ \left\{i,\ 0,\ \text{range}_z,\ 10^{-6}\right\}\right];$$

(*m*) (*In 10^{-6}m=1um steps.*)

```
(*-----------------------------------------------------------*)

(*----------------------Faubel's Equation----------------------*)
```

(*Equations for T_{Faubel} are based on Faubel's 1998's paper eqn(8)+eqn(9a),
which is same as Nathanson's 2016's paper eqn(2).*)

$$T_{\text{Faubel}} = \text{Table}\left[0,\ \left\{i,\ 0,\ \text{range}_z,\ 10^{-6}\right\}\right];$$

$$T_{\text{Faubel}}[\![1]\!] = T_{\text{nozzle}};$$

$$\text{For}\Big[i = 1,\ i < \text{Dimensions}[z][\![1]\!],\ i\text{++},$$

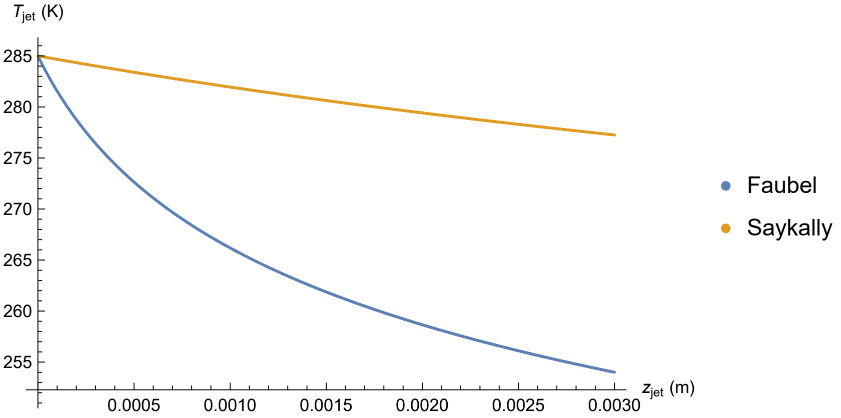
$$T_{\text{Faubel}}[\![i+1]\!] = T_{\text{Faubel}}[\![i]\!]-\sqrt{\frac{32\,\frac{M}{1000}}{\pi\,R\,T_{\text{Faubel}}[\![i]\!]}}\left(\left(P_{\text{water,Sippola}}[T_{\text{Faubel}}[\![i]\!]]\ 133.322368\ \Delta H_{\text{vap}}\right)\Big/\left(\rho_{\text{liq,water}}[T_{\text{Faubel}}[\![i]\!]]\ C_p\ r_{\text{jet}}\right)\right)\frac{1}{v_{\text{jet}}}\left(z[\![i+1]\!]-z[\![i]\!]\right)\Big];$$

```
List_Faubel = Partition[Riffle[z, T_Faubel], 2];
(*-----------------------------------------------------------------*)

(*----------------------Saykally's Equation---------------------*)
(*Equations for T_Saykally are based on Cohen'
 s 2005 eqn(9) and Saykally's 2006 paper eqn(10).*)
T_Saykally = Table[0, {i, 0, range_z, 10^-6}];
T_Saykally[[1]] = T_nozzle;
For[i = 1, i < Dimensions[z][[1]], i++, T_Saykally[[i + 1]] =
   T_Saykally[[i]] - Sqrt[(2 φ_e^2)/(π M/1000 R T_Saykally[[i]])] ((P_water,Sippola[T_Saykally[[i]]] 133.322368 ΔH_vap) /
      (1/(M/1000) ρ_liq,water[T_Saykally[[i]]] C_p r_jet)) 1/v_jet (z[[i + 1]] - z[[i]])];
List_Saykally = Partition[Riffle[z, T_Saykally], 2];
(*-----------------------------------------------------------------*)

(*---------------------------Plotting-------------------------*)
Print[Style["I. Cyl. Jet_with (dr/dt)_evap", 18, Bold, Blue]]
Print["For cyl. jet with T_nozzle = ", N[T_nozzle],
 " K, r_0 = ", N[r_jet * 10^6], " μm, v_jet = ", N[v_jet, 1], " m/s"]
Print["    at z = ", N[z_set 10^3], " mm,  T_Faubel = ", List_Faubel[[z_set / 10^-6 + 1]][[2]],
 " K, T_Saykally = ", List_Saykally[[z_set / 10^-6 + 1]][[2]], " K"]
ListPlot[{List_Faubel, List_Saykally},
 PlotLegends → {"Faubel", "Saykally"}, AxesLabel → {"z_jet (m)", "T_jet (K)"}]
(*-----------------------------------------------------------------*)
```

I. Cyl. Jet_with $(dr/dt)_{evap}$

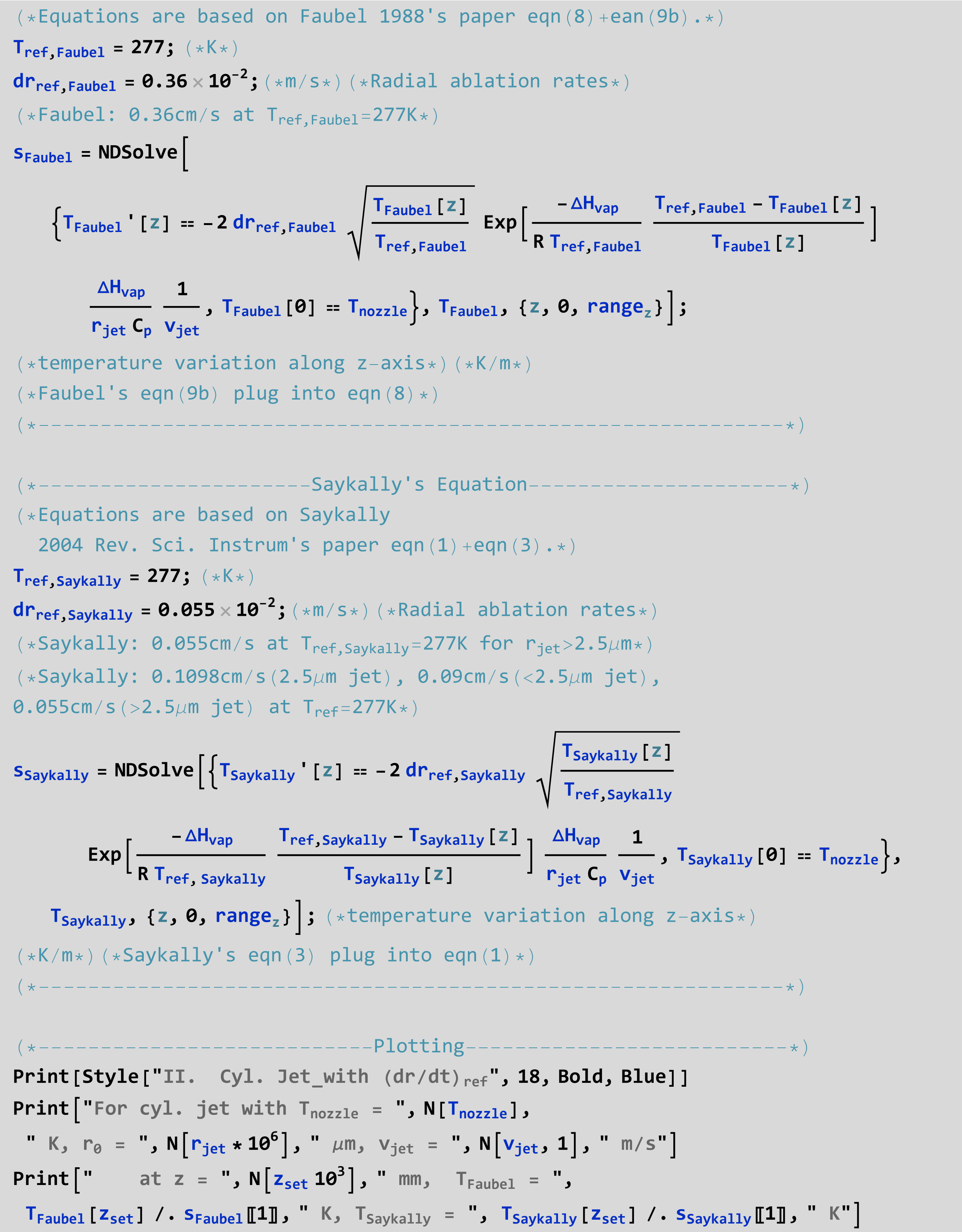
```
For cyl. jet with T_nozzle = 285. K, r_0 = 13.8 μm, v_jet = 13.9287 m/s
    at z = 1.5 mm,  T_Faubel = 261.872 K, T_Saykally = 280.628 K
```

Out[30]=



In[31]:=

```
(*-----------------II.  Cyl. Jet_with (dr/dt)_ref-----------------*)
(*Using reference point and C.-C. relation to calculate (dr/dt)_ref.*)
(*Equations for T_Faubel are based on Faubel 1988's paper eqn(8)+ean(9b).*)
(*Equations for T_Saykally are based on
  Saykally 2004 Rev. Sci. Instrum's paper eqn(1)+eqn(3).*)
ClearAll["Global`*"]
(*-----------------------------------------------------------------*)

(*-------------Faubel vs Saykally's Equation Parameter------------*)
(*Jet parameters*)
T_nozzle = 285; (*K*)
r_jet = 27.6 × 10^-6 / 2; (*m*)
(*27.6um is the average diameter of the chip nozzle central channel.*)
v_jet = (0.5 × 10^-6)/(π (r_jet)^2) (1/60); (*m/s*) (*With flow rate 0.5mL/min,
calculated from the excel file.*)
z_set = 1.5 × 10^-3; (*m*) (*1.5mm is the detecting height for cylindrical jet.*)
range_z = 3 × 10^-3; (*m*) (*x-axis range for the plot.*)

(*Lquid parameters - water*)
M = 18; (*g/mol*)
ΔH_vap = 2270 * M; (*J/mol*)
C_p = 4.2 * M; (*J/mol/K*)
R = 8.314; (*J/K/mol*)
(*-----------------------------------------------------------------*)

(*-----------------------Faubel's Equation-----------------------*)
```



```
(*Equations are based on Faubel 1988's paper eqn(8)+ean(9b).*)
T_{ref,Faubel} = 277; (*K*)
dr_{ref,Faubel} = 0.36×10^{-2};(*m/s*)(*Radial ablation rates*)
(*Faubel: 0.36cm/s at T_{ref,Faubel}=277K*)
s_{Faubel} = NDSolve[
   {T_{Faubel}'[z] == -2 dr_{ref,Faubel} Sqrt[T_{Faubel}[z]/T_{ref,Faubel}] Exp[(-ΔH_{vap})/(R T_{ref,Faubel}) (T_{ref,Faubel} - T_{Faubel}[z])/(T_{Faubel}[z])]
       (ΔH_{vap})/(r_{jet} C_p) 1/(v_{jet}), T_{Faubel}[0] == T_{nozzle}}, T_{Faubel}, {z, 0, range_z}];
(*temperature variation along z-axis*)(*K/m*)
(*Faubel's eqn(9b) plug into eqn(8)*)
(*-------------------------------------------------------------*)

(*----------------------Saykally's Equation---------------------*)
(*Equations are based on Saykally
  2004 Rev. Sci. Instrum's paper eqn(1)+eqn(3).*)
T_{ref,Saykally} = 277; (*K*)
dr_{ref,Saykally} = 0.055×10^{-2};(*m/s*)(*Radial ablation rates*)
(*Saykally: 0.055cm/s at T_{ref,Saykally}=277K for r_{jet}>2.5μm*)
(*Saykally: 0.1098cm/s(2.5μm jet), 0.09cm/s(<2.5μm jet),
0.055cm/s(>2.5μm jet) at T_{ref}=277K*)
s_{Saykally} = NDSolve[{T_{Saykally}'[z] == -2 dr_{ref,Saykally} Sqrt[T_{Saykally}[z]/T_{ref,Saykally}]
      Exp[(-ΔH_{vap})/(R T_{ref, Saykally}) (T_{ref,Saykally} - T_{Saykally}[z])/(T_{Saykally}[z])] (ΔH_{vap})/(r_{jet} C_p) 1/(v_{jet}), T_{Saykally}[0] == T_{nozzle}},
   T_{Saykally}, {z, 0, range_z}]; (*temperature variation along z-axis*)
(*K/m*)(*Saykally's eqn(3) plug into eqn(1)*)
(*-------------------------------------------------------------*)

(*---------------------------Plotting--------------------------*)
Print[Style["II.  Cyl. Jet_with (dr/dt)_{ref}", 18, Bold, Blue]]
Print["For cyl. jet with T_{nozzle} = ", N[T_{nozzle}],
 " K, r_0 = ", N[r_{jet} * 10^6], " μm, v_{jet} = ", N[v_{jet}, 1], " m/s"]
Print["    at z = ", N[z_{set} 10^3], " mm,  T_{Faubel} = ",
 T_{Faubel}[z_{set}] /. s_{Faubel}[[1]], " K, T_{Saykally} = ", T_{Saykally}[z_{set}] /. s_{Saykally}[[1]], " K"]
```

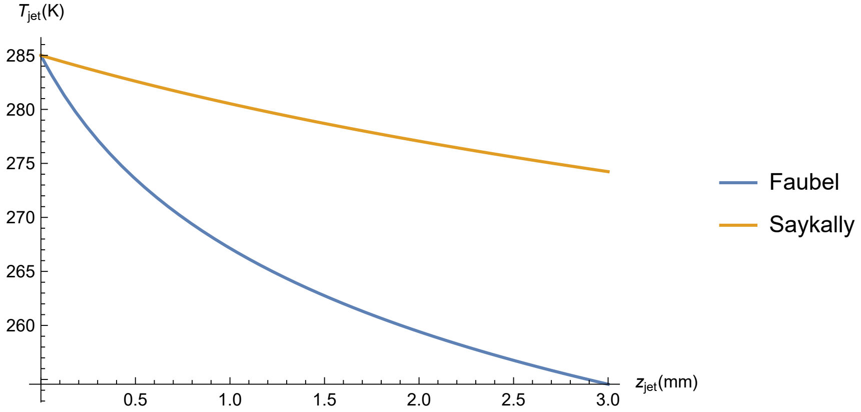
```
Plot[{Evaluate[T_Faubel[z 10^-3] /. s_Faubel], Evaluate[T_Saykally[z 10^-3] /. s_Saykally]},
 {z, 0, range_z 10^3}, AxesLabel → {"z_jet (mm)", "T_jet (K)"},
 PlotRange → All, PlotLegends → {"Faubel", "Saykally"}]
(*-------------------------------------------------------------*)
```

II. Cyl. Jet_with $(dr/dt)_{ref}$

For cyl. jet with T_{nozzle} = 285. K, r_0 = 13.8 μm, v_{jet} = 13.9287 m/s

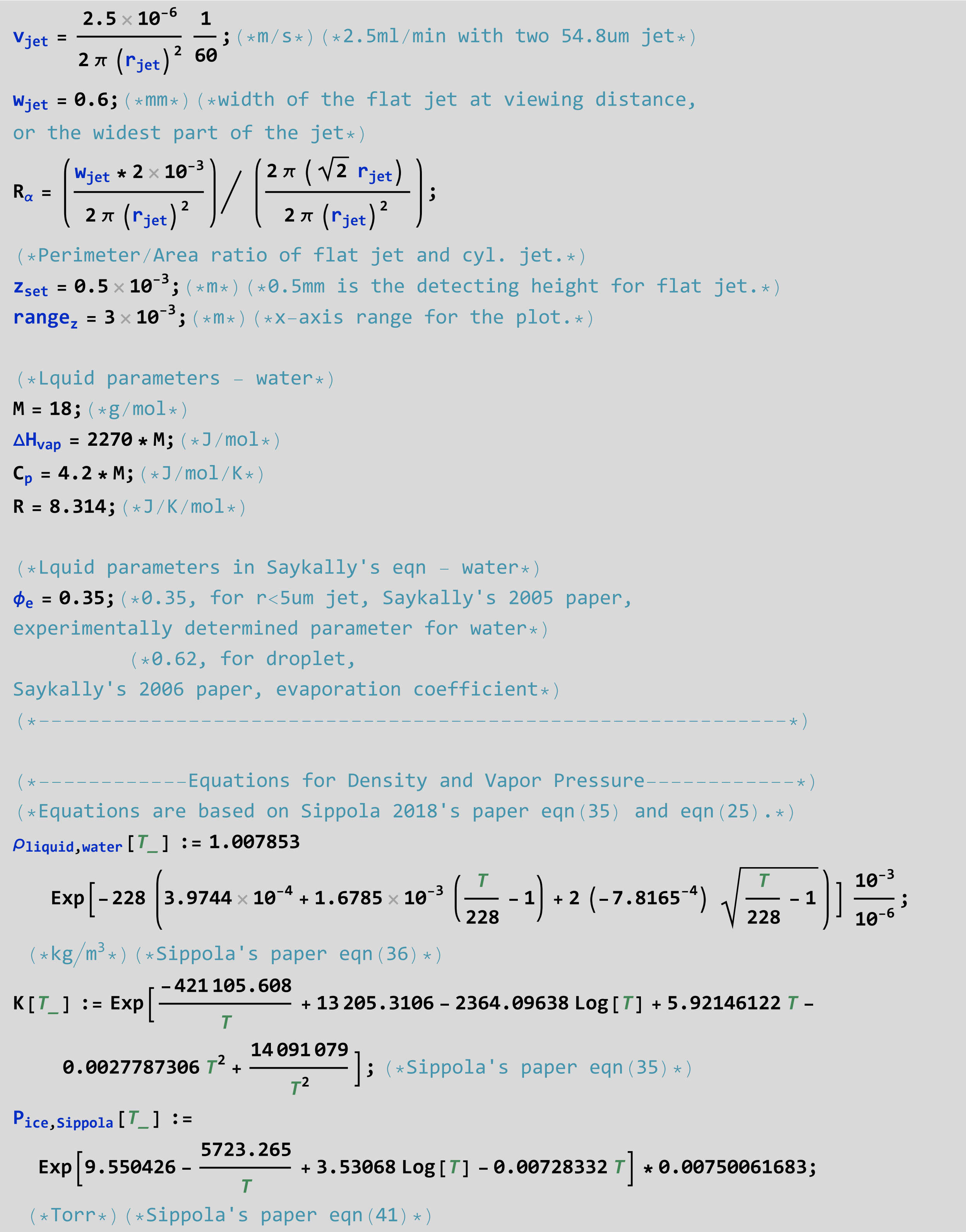
at z = 1.5 mm, T_{Faubel} = 262.749 K, $T_{Saykally}$ = 278.697 K

Out[50]=



In[51]:=

```
(*-----------------III. Flat Jet_with (dr/dt)_evap----------------*)
(*Using evaporation theory to estimate
 (dr/dt)_evap. Numerically calculate ρ_liquid and P_gas.*)
(*Equations for T_Faubel are based on Faubel's 1998's paper eqn(8)+eqn(9a),
which is same as Nathanson's 2016's paper eqn(2).*)
(*Equations for T_Saykally are based on Cohen'
 s 2005 eqn(9) and Saykally's 2006 paper eqn(10).*)
(*Equations for water's density and vapor pressure
 are based on Sippola's 2018 paper eqn(25) and eqn(36).*)
(*An extra constant, R_flat jet ratio,
is used to estimate temperature of flat jet.*)
ClearAll["Global`*"]
(*-------------------------------------------------------------*)

(*------------Faubel vs Saykally's Equation Parameter------------*)
(*Jet parameters*)
T_nozzle = 285; (*K*)
r_jet = 54.8 × 10^-6 / 2; (*m*)
(*54.8um is the average diameter of the chip nozzle outer channel*)
```



$v_{jet} = \frac{2.5\times10^{-6}}{2\pi\left(r_{jet}\right)^{2}}\frac{1}{60}$; (*m/s*) (*2.5ml/min with two 54.8um jet*)

$w_{jet} = 0.6$; (*mm*) (*width of the flat jet at viewing distance, or the widest part of the jet*)

$R_{\alpha} = \left(\frac{w_{jet} * 2\times10^{-3}}{2\pi\left(r_{jet}\right)^{2}}\right) \Big/ \left(\frac{2\pi\left(\sqrt{2}\, r_{jet}\right)}{2\pi\left(r_{jet}\right)^{2}}\right)$;

(*Perimeter/Area ratio of flat jet and cyl. jet.*)

$z_{set} = 0.5\times10^{-3}$; (*m*) (*0.5mm is the detecting height for flat jet.*)

$range_{z} = 3\times10^{-3}$; (*m*) (*x-axis range for the plot.*)

(*Lquid parameters - water*)

$M = 18$; (*g/mol*)

$\Delta H_{vap} = 2270 * M$; (*J/mol*)

$C_{p} = 4.2 * M$; (*J/mol/K*)

$R = 8.314$; (*J/K/mol*)

(*Lquid parameters in Saykally's eqn - water*)

$\phi_{e} = 0.35$; (*0.35, for r<5um jet, Saykally's 2005 paper, experimentally determined parameter for water*)

(*0.62, for droplet, Saykally's 2006 paper, evaporation coefficient*)

(*--*)

(*------------Equations for Density and Vapor Pressure------------*)

(*Equations are based on Sippola 2018's paper eqn(35) and eqn(25).*)

$$\rho_{liquid,water}[T\_] := 1.007853\ \mathrm{Exp}\left[-228\left(3.9744\times10^{-4} + 1.6785\times10^{-3}\left(\frac{T}{228}-1\right) + 2\left(-7.8165^{-4}\right)\sqrt{\frac{T}{228}-1}\right)\right]\frac{10^{-3}}{10^{-6}};$$

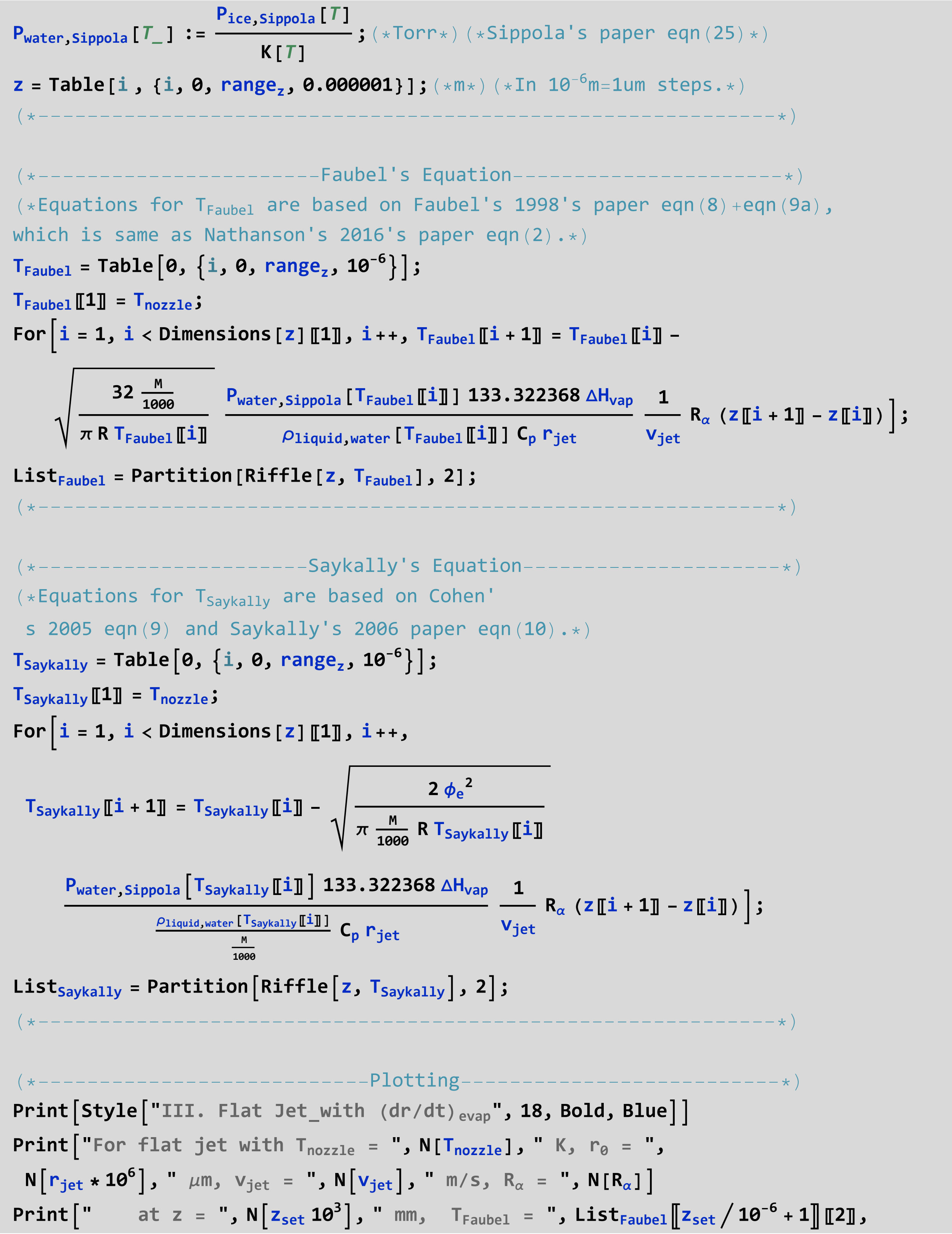
(*kg/m^3*) (*Sippola's paper eqn(36)*)

$$K[T\_] := \mathrm{Exp}\left[\frac{-421\,105.608}{T} + 13\,205.3106 - 2364.09638\,\mathrm{Log}[T] + 5.92146122\,T - 0.0027787306\,T^{2} + \frac{14\,091\,079}{T^{2}}\right];$$

(*Sippola's paper eqn(35)*)

$$P_{ice,Sippola}[T\_] := \mathrm{Exp}\left[9.550426 - \frac{5723.265}{T} + 3.53068\,\mathrm{Log}[T] - 0.00728332\,T\right] * 0.00750061683;$$

(*Torr*) (*Sippola's paper eqn(41)*)



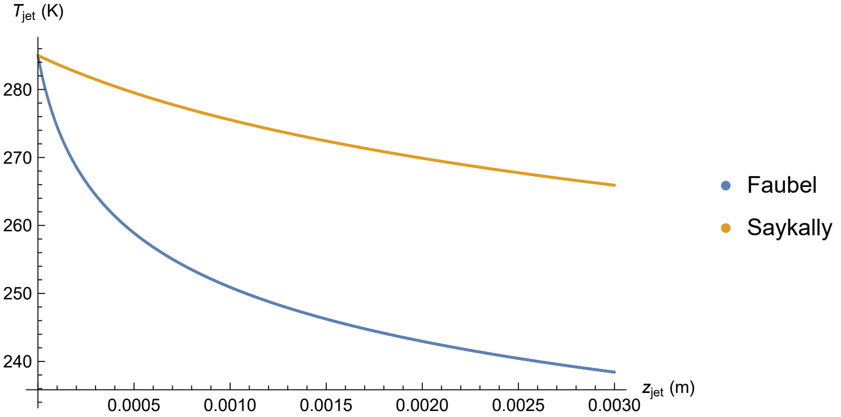
```
P_{water,Sippola}[T_] := (P_{ice,Sippola}[T])/(K[T]); (*Torr*) (*Sippola's paper eqn(25)*)
z = Table[i , {i, 0, range_z, 0.000001}]; (*m*) (*In 10^-6m=1um steps.*)
(*-------------------------------------------------------------*)

(*-----------------------Faubel's Equation-----------------------*)
(*Equations for T_{Faubel} are based on Faubel's 1998's paper eqn(8)+eqn(9a),
which is same as Nathanson's 2016's paper eqn(2).*)
T_{Faubel} = Table[0, {i, 0, range_z, 10^-6}];
T_{Faubel}[[1]] = T_{nozzle};
For[i = 1, i < Dimensions[z][[1]], i++, T_{Faubel}[[i + 1]] = T_{Faubel}[[i]] -
   Sqrt[(32 M/1000)/(π R T_{Faubel}[[i]])] (P_{water,Sippola}[T_{Faubel}[[i]]] 133.322368 ΔH_{vap})/(ρ_{liquid,water}[T_{Faubel}[[i]]] C_p r_{jet}) 1/v_{jet} R_α (z[[i + 1]] - z[[i]])];
List_{Faubel} = Partition[Riffle[z, T_{Faubel}], 2];
(*-------------------------------------------------------------*)

(*----------------------Saykally's Equation----------------------*)
(*Equations for T_{Saykally} are based on Cohen'
 s 2005 eqn(9) and Saykally's 2006 paper eqn(10).*)
T_{Saykally} = Table[0, {i, 0, range_z, 10^-6}];
T_{Saykally}[[1]] = T_{nozzle};
For[i = 1, i < Dimensions[z][[1]], i++,

 T_{Saykally}[[i + 1]] = T_{Saykally}[[i]] - Sqrt[(2 φ_e^2)/(π M/1000 R T_{Saykally}[[i]])]

    (P_{water,Sippola}[T_{Saykally}[[i]]] 133.322368 ΔH_{vap})/((ρ_{liquid,water}[T_{Saykally}[[i]]])/(M/1000) C_p r_{jet}) 1/v_{jet} R_α (z[[i + 1]] - z[[i]])];
List_{Saykally} = Partition[Riffle[z, T_{Saykally}], 2];
(*-------------------------------------------------------------*)

(*----------------------------Plotting---------------------------*)
Print[Style["III. Flat Jet_with (dr/dt)_{evap}", 18, Bold, Blue]]
Print["For flat jet with T_{nozzle} = ", N[T_{nozzle}], " K, r_0 = ",
 N[r_{jet} * 10^6], " μm, v_{jet} = ", N[v_{jet}], " m/s, R_α = ", N[R_α]]
Print["    at z = ", N[z_{set} 10^3], " mm,  T_{Faubel} = ", List_{Faubel}[[z_{set} / 10^-6 + 1]][[2]],
 " K, T_Saykally = ", List_Saykally[[z_set / 10^-6 + 1]][[2]], " K"]
ListPlot[{List_Faubel, List_Saykally},
 PlotLegends → {"Faubel", "Saykally"}, AxesLabel → {"z_jet (m)", "T_jet (K)"}]
(*------------------------------------------------------------------*)
```

III. Flat Jet_with (dr/dt)$_{evap}$

For flat jet with T_{nozzle} = 285. K, r_0 = 27.4 μm, v_{jet} = 8.83299 m/s, R_α = 4.92874

at z = 0.5 mm, T_{Faubel} = 258.886 K, $T_{Saykally}$ = 279.514 K

Out[78]=



In[79]:=

```
(*------------------IV.  Flat Jet_with (dr/dt)_ref------------------*)
(*Using reference point and C.-C. relation to calculate (dr/dt)_ref.*)
(*Equations for T_Faubel are based on Faubel 1988's paper eqn(8)+ean(9b).*)
(*Equations for T_Saykally are based on
  Saykally 2004 Rev. Sci. Instrum's paper eqn(1)+eqn(3).*)
(*An extra constant, R_flat jet ratio,
is used to estimate temperature of flat jet.*)
ClearAll["Global`*"]
(*------------------------------------------------------------------*)


(*--------------Faubel vs Saykally's Equation Parameter-------------*)
(*Jet parameters*)
T_nozzle = 285; (*K*)
r_jet = 54.8 × 10^-6 / 2; (*m*)
(*54.8um is the average diameter of the chip nozzle outer channel*)
v_jet = (2.5 × 10^-6 / (2 π (r_jet)^2)) (1/60); (*m/s*) (*2.5ml/min with two 54.8um jet*)
w_jet = 0.6; (*mm*) (*width of the flat jet at viewing distance,
or the widest part of the jet*)
```

$$R_{\alpha} = \left(\frac{w_{jet} * 2 \times 10^{-3}}{2\,\pi\,\left(r_{jet}\right)^{2}}\right) \Big/ \left(\frac{2\,\pi\,\left(\sqrt{2}\;r_{jet}\right)}{2\,\pi\,\left(r_{jet}\right)^{2}}\right);$$

```
(*Perimeter/Area ratio of flat jet and cyl. jet.*)
```

$z_{set} = 0.5 \times 10^{-3}$; `(*m*) (*0.5mm is the detecting height for flat jet.*)`

$range_{z} = 3 \times 10^{-3}$; `(*m*) (*x-axis range for the plot.*)`

```
(*Lquid parameters - water*)
M = 18; (*g/mol*)
```

$\Delta H_{vap} = 2270 * M$; `(*J/mol*)`

$C_{p} = 4.2 * M$; `(*J/mol/K*)`

```
R = 8.314; (*J/K/mol*)
(*-----------------------------------------------------------------*)

(*------------------------Faubel's Equation-----------------------*)
(*Equations are based on Faubel 1988's paper eqn(8)+ean(9b).*)
```

$T_{ref,Faubel} = 277$; `(*K*)`

$dr_{ref,Faubel} = 0.36 \times 10^{-2}$; `(*m/s*) (*Radial ablation rates*)`

`(*Faubel: 0.36cm/s at` T_{ref}`=277K*)`

$$s_{Faubel} = \text{NDSolve}\Big[\Big\{T_{Faubel}{}'[z] == -2\, dr_{ref,Faubel} \sqrt{\frac{T_{Faubel}[z]}{T_{ref,Faubel}}}\ \text{Exp}\left[\frac{-\Delta H_{vap}}{R\, T_{ref,Faubel}}\ \frac{T_{ref,Faubel} - T_{Faubel}[z]}{T_{Faubel}[z]}\right] \frac{\Delta H_{vap}}{r_{jet}\, C_{p}}\ \frac{1}{v_{jet}}\ R_{\alpha},\ T_{Faubel}[0] == T_{nozzle}\Big\},\ T_{Faubel},\ \{z, 0, range_{z}\}\Big];$$

```
(*temperature variation along z-axis*) (*K/m*)
(*Faubel's Eq.9b plug into Eq.8*)
(*-----------------------------------------------------------------*)

(*-----------------------Saykally's Equation----------------------*)
(*Equations are based on Saykally
  2004 Rev. Sci. Instrum's paper eqn(1)+eqn(3).*)
```

$T_{ref,Saykally} = 277$; `(*K*)`

$dr_{ref,Saykally} = 0.055 \times 10^{-2}$; `(*m/s*)`

`(*Radial ablation rates*) (*Saykally 2004's paper: 0.1098cm/s(2.5um jet), 0.09cm/s(<2.5um jet), 0.055cm/s(>2.5um jet) at` T_{ref}`=277K*)`

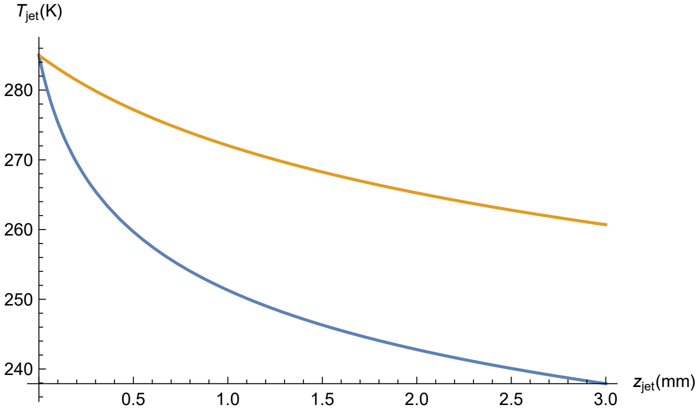
```
sSaykally = NDSolve[{TSaykally'[z] == -2 dr_ref,Saykally Sqrt[TSaykally[z]/Tref,Saykally]
      Exp[(-ΔHvap)/(R Tref,Saykally) (Tref,Saykally - TSaykally[z])/TSaykally[z]] ΔHvap/(rjet Cp) 1/vjet Rα, TSaykally[0] == Tnozzle},
   TSaykally, {z, 0, rangez}]; (*temperature variation along z-axis*)
(*K/m*) (*Saykally's eqn(3) plug into eqn(1)*)
(*-----------------------------------------------------------------*)

(*----------------------------Plotting---------------------------*)
Print[Style["IV.  Flat Jet_with (dr/dt)ref", 18, Bold, Blue]]
Print["For flat jet with Tnozzle = ", N[Tnozzle], " K, r0 = ",
  N[rjet * 10^6], " μm, vjet = ", N[vjet], " m/s, Rα = ", N[Rα]]
Print["    at z = ", N[zset 10^3], " mm,  TFaubel = ",
  TFaubel[zset] /. sFaubel[[1]], " K, TSaykally = ", TSaykally[zset] /. sSaykally[[1]], " K"]
Plot[{Evaluate[TFaubel[z 10^-3] /. sFaubel], Evaluate[TSaykally[z 10^-3] /. sSaykally]},
  {z, 0, rangez 10^3}, AxesLabel → {"zjet(mm)", "Tjet(K)"},
  PlotRange → All, PlotLegends → {"Faubel", "Saykally"}]
(*-----------------------------------------------------------------*)
```

IV. Flat Jet_with $(dr/dt)_{ref}$

For flat jet with T_{nozzle} = 285. K, r_0 = 27.4 μm, v_{jet} = 8.83299 m/s, R_α = 4.92874

at z = 0.5 mm, T_{Faubel} = 259.68 K, $T_{Saykally}$ = 277.193 K

Out[100]=



A.3 Number of Collisions

In this code, it includes four sub-codes to calculate four useful properties of the gas surrounding the jet. They are the gas pressure (P), mean free path (λ), number of collisions (N_{coll}) and non-collision probability ($Prob_{non-coll}$). All the properties are calculated using an electrostatic analogy as a function of R, which is the axis along the jet surface normal.

These four properties are calculated for three types of jet. The jets are cylindrical jet with infinite length, cylindrical jet with fixed length L and flat jet. With electrostatic analogy, equations of P for these jets are written as [40, 42, 53]

$$P_{cyl,inf} = \frac{P_z}{2}\frac{r_{jet}}{R},$$

$$P_{cyl,L} = \frac{L_{jet}P_z r_{jet}}{2R\sqrt{{L_{jet}}^2 + R^2}},$$

$$P_{flat} = \frac{P_z}{2}\left(1 - \frac{R}{\sqrt{R^2 + {r_{disc}}^2}}\right)$$

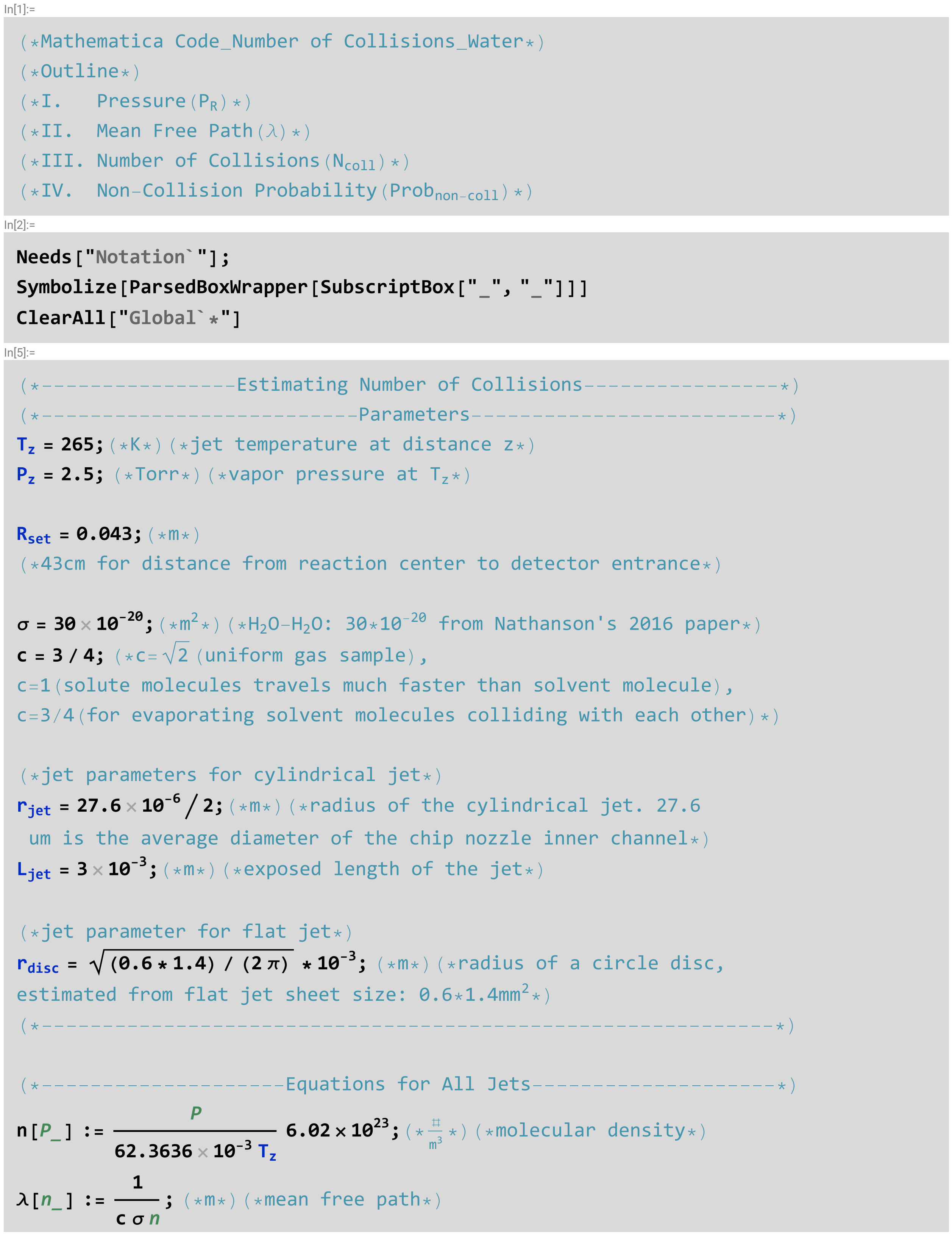
which are same as Equation 1.11, 1.12 and 1.13. One should note that there's a 1/2 factor for all the pressure equation, which is because that particles evaporating towards half of the direction get to be pumped out in vacuum. λ is a defined as $\lambda = (c\sigma n)^{-1}$, which can be converted from P. As for N_{coll}, the equations are

$$N_{cyl,inf} = \frac{r_{jet}}{\lambda}\ln\left[\frac{R_{set}}{r_{jet}}\right],$$

$$N_{cyl,L} = \frac{r_{jet}}{\lambda}\ln\left[\frac{R_{set}}{r_{jet}}\frac{L_{jet} + \sqrt{{r_{jet}}^2 + {L_{jet}}^2}}{L_{jet} + \sqrt{{R_{set}}^2 + {L_{jet}}^2}}\right],$$

$$N_{flat} = \frac{r_{dis}}{\lambda}\left(1 + \frac{R}{r_{disc}} - \sqrt{1 + \left(\frac{R}{r_{disc}}\right)^2}\right)$$

which are essentially the same equations in Equation 1.15, 1.16 and 1.17. $Prob_{non-coll}$ is calculated by using the relation of $Prob_{non-coll} = \exp\left[-N_{coll}\right]$. For more details of these properties, they are described in Section 1.3.



In[1]:=

```
(*Mathematica Code_Number of Collisions_Water*)
(*Outline*)
(*I.   Pressure(P_R)*)
(*II.  Mean Free Path(λ)*)
(*III. Number of Collisions(N_coll)*)
(*IV.  Non-Collision Probability(Prob_non-coll)*)
```

In[2]:=

```
Needs["Notation`"];
Symbolize[ParsedBoxWrapper[SubscriptBox["_", "_"]]]
ClearAll["Global`*"]
```

In[5]:=

```
(*----------------Estimating Number of Collisions----------------*)
(*-------------------------Parameters------------------------*)
T_z = 265;(*K*)(*jet temperature at distance z*)
P_z = 2.5; (*Torr*)(*vapor pressure at T_z*)

R_set = 0.043;(*m*)
(*43cm for distance from reaction center to detector entrance*)

σ = 30 × 10^-20;(*m^2*)(*H_2O-H_2O: 30*10^-20 from Nathanson's 2016 paper*)
c = 3 / 4; (*c=√2 (uniform gas sample),
c=1(solute molecules travels much faster than solvent molecule),
c=3/4(for evaporating solvent molecules colliding with each other)*)

(*jet parameters for cylindrical jet*)
r_jet = 27.6 × 10^-6 / 2;(*m*)(*radius of the cylindrical jet. 27.6
 um is the average diameter of the chip nozzle inner channel*)
L_jet = 3 × 10^-3;(*m*)(*exposed length of the jet*)

(*jet parameter for flat jet*)
r_disc = √((0.6 * 1.4) / (2 π)) * 10^-3; (*m*)(*radius of a circle disc,
estimated from flat jet sheet size: 0.6*1.4mm^2*)
(*-----------------------------------------------------------*)

(*--------------------Equations for All Jets--------------------*)
                   P
n[P_] := ───────────────────── 6.02 × 10^23; (*#/m^3*)(*molecular density*)
         62.3636 × 10^-3 T_z

           1
λ[n_] := ─────; (*m*)(*mean free path*)
          c σ n
```

```
Prob[N_] := Exp[-N];(*non-collision probability*)
(*------------------------------------------------------------*)

(*----------Equations for Cylindrical Jet with Length inf---------*)
```

$$P_{cyl,inf}[R\_] := \frac{P_z}{2}\,\frac{r_{jet}}{R};\ (*Torr*)\,(*\frac{P_{vap}}{2}*)$$

$$N_{cyl,inf}[R\_] := \frac{r_{jet}}{\lambda\left[n\left[P_{cyl,inf}\left[r_{jet}\right]\right]\right]}\,\mathrm{Log}\left[\frac{R}{r_{jet}}\right];\ (*\#*)\,(*number\ of\ collisions*)$$

```
(*------------------------------------------------------------*)

(*-----------Equations for Cylindrical Jet with Length L----------*)
```

$$P_{cyl,L}[R\_] := \frac{L_{jet}\,P_z\,r_{jet}}{2\,R\,\sqrt{L_{jet}{}^2+R^2}};\ (*Torr*)\,(*pressure*)$$

$$N_{cyl,L}[R\_] := \frac{r_{jet}}{\lambda\left[n\left[P_{cyl,L}\left[r_{jet}\right]\right]\right]}\,\mathrm{Log}\left[\frac{R}{r_{jet}}\,\frac{L_{jet}+\sqrt{r_{jet}{}^2+L_{jet}{}^2}}{L_{jet}+\sqrt{R^2+L_{jet}{}^2}}\right];$$

```
(*#*)(*number of collisions*)
(*------------------------------------------------------------*)

(*---------------------Equations for Flat Jet--------------------*)
```

$$P_{flat}[R\_] := \frac{P_z}{2}\left(1-\frac{R}{\sqrt{R^2+r_{disc}{}^2}}\right);\ (*Torr*)\,(*pressure*)$$

$$N_{flat}[R\_] := \frac{r_{disc}}{\lambda[n[P_{flat}[0]]]}\left(1+\frac{R}{r_{disc}}-\sqrt{1+\left(\frac{R}{r_{disc}}\right)^2}\right);$$

```
(*#*)(*number of collisions*)
(*------------------------------------------------------------*)

(*--------------------------Plotting--------------------------*)
Print[Style["Estimating Number of Collision", 18, Bold, Blue]]
```

Print["Liquid conditions: T_z = ", T_z, " K where P_z = ", P_z, " Torr."]

Print["Jet conditions: r_{jet} = ", $r_{jet}\,10^6$ " μm; L_{jet} = ",

$L_{jet}\,10^3$, " mm; r_{disc} = ", NumberForm[$r_{disc}\,10^3$, 3], " mm."]

Print[Style["I. Pressure(P_R)", 16, Bold, Blue]]

Print["At R_{set} = ", $R_{set}\,10^3$, " mm:"];

Print[" $P_{cyl,inf}$= ", ScientificForm[$P_{cyl,inf}[R_{set}]$, 3], " Torr"];

Print[" $P_{cyl,L}$=   ", ScientificForm[$P_{cyl,L}[R_{set}]$, 3], " Torr"];

```
Print["    P_flat=     ", ScientificForm[P_flat[R_set], 3], " Torr"];
Plot[{P_cyl,inf[R 10^-3], P_cyl,L[R 10^-3], {P_flat[R 10^-3]}}, {R, r_jet, R_set 10^3},
 AxesLabel → {"R(mm)", "Pressure(Torr)"}, PlotLegends → {"Cyl_inf", "Cyl.", "Flat"}]

Print[Style["II. Mean Free Path(λ)", 16, Bold, Blue]]
Print["At R_set = ", R_set 10^3, " mm:"];
Print["    λ_cyl,inf=   ", ScientificForm[λ[n[P_cyl,inf[R_set]]], 3], " m"];
Print["    λ_cyl,L=    ", ScientificForm[λ[n[P_cyl,L[R_set]]], 3], " m"];
Print["    λ_flat=     ", ScientificForm[λ[n[P_flat[R_set]]], 3], " m"];
Plot[{λ[n[P_cyl,inf[R 10^-3]]], λ[n[P_cyl,L[R 10^-3]]], λ[n[P_flat[R 10^-3]]]},
 {R, r_jet, R_set 10^3}, AxesLabel → {"R(mm)", "Mean Free Path λ(m)"},
 PlotLegends → {"Cyl_inf", "Cyl.", "Flat"}]

Print[Style["III. Number of Collisions(N_coll)", 16, Bold, Blue]]
Print["At R_set = ", R_set 10^3, " mm:"];
Print["    N_cyl,inf=   ", ScientificForm[N_cyl,inf[R_set], 3], " #"];
Print["    N_cyl,L=    ", ScientificForm[N_cyl,L[R_set], 3], " #"];
Print["    N_flat=     ", ScientificForm[N_flat[R_set], 3], " #"];
Plot[{N_cyl,inf[R 10^-3], N_cyl,L[R 10^-3], N_flat[R 10^-3]},
 {R, r_jet, R_set 10^3}, AxesLabel → {"R(mm)", "Number of Collisions(#)"},
 PlotLegends → {"Cyl_inf", "Cyl.", "Flat"}]

Print[Style["IV. Non-Collision Probability(Prob_non-coll)", 16, Bold, Blue]]
Print["At R_set = ", R_set 10^3, " mm:"];
Print["    Prob_cyl,inf=   ", 100 Prob[N_cyl,inf[R_set]], " %"];
Print["    Prob_cyl,L=    ", 100 Prob[N_cyl,L[R_set]], " %"];
Print["    Prob_flat=     ", 100 Prob[N_flat[R_set]], " %"];
Plot[{100 Prob[N_cyl,inf[R 10^-3]], 100 Prob[N_cyl,L[R 10^-3]], 100 Prob[N_flat[R 10^-3]]},
 {R, r_jet, R_set 10^3}, AxesLabel → {"R(mm)", "Non-Collision Probability(%)"},
 PlotLegends → {"Cyl_inf", "Cyl_L", "Flat"}]
(*----------------------------------------------------------------*)
```

Estimating Number of Collision

Liquid conditions: $T_z = 265$ K where $P_z = 2.5$ Torr.

Jet conditions: $r_{jet} = 13.8$ μm; $L_{jet} = 3$ mm; $r_{disc} = 0.366$ mm.

I. Pressure(P_R)

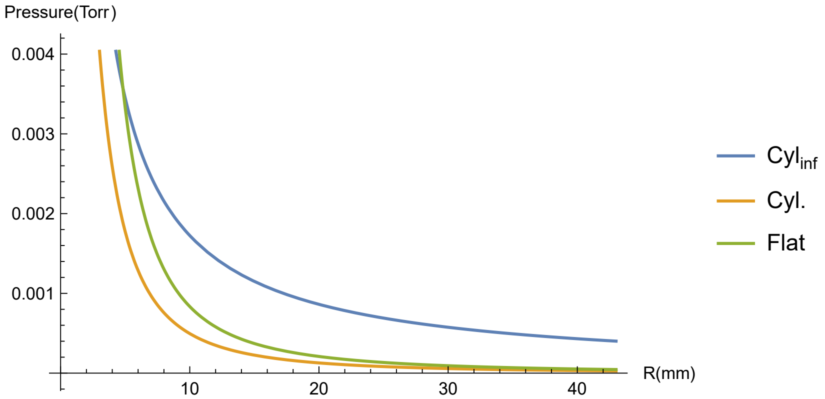
At $R_{set} = 43.$ mm:

$P_{cyl,inf}=$ 4.01×10^{-4} Torr

$P_{cyl,L}=$ 2.79×10^{-5} Torr

$P_{flat}=$ 4.52×10^{-5} Torr

Out[30]=



II. Mean Free Path (λ)

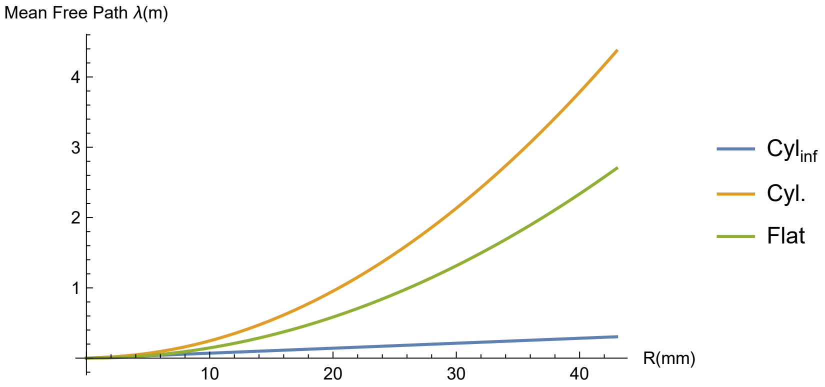
At R_{set} = 43. mm:

$\lambda_{cyl,inf}=$ 3.04×10^{-1} m

$\lambda_{cyl,L}=$ 4.37 m

$\lambda_{flat}=$ 2.7 m

Out[36]=



III. Number of Collisions (N_{coll})

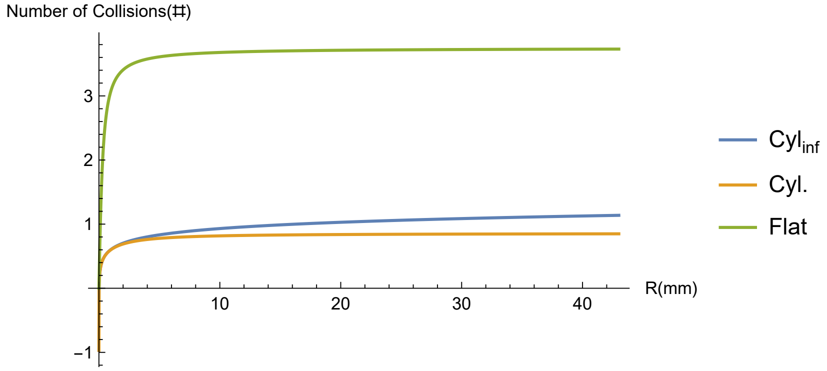
At R_{set} = 43. mm:

$N_{cyl,inf}=$ 1.14 ⌗

$N_{cyl,L}=$ 8.49×10^{-1} ⌗

$N_{flat}=$ 3.73 ⌗

Out[42]=



IV. Non-Collision Probability (Prob$_{non-coll}$)

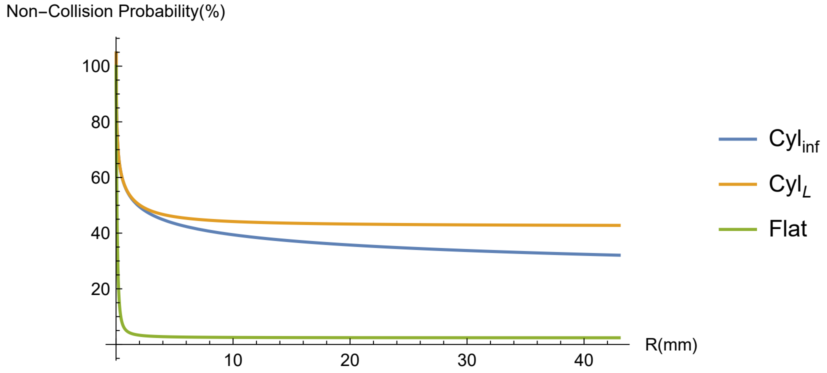
At R_{set} = 43. mm:

$Prob_{cyl,inf}$= 32.0681 %

$Prob_{cyl,L}$= 42.7841 %

$Prob_{flat}$= 2.39923 %

Out[48]=



Appendix B

Laboratory Software

This appendix presents two programs for operating the flat jet experiments. Both programs are written in National Instruments' (NI) LabVIEW 2019. In each program, both the front panel and the block diagram are shown, which the former part shows the user interface, and the later part shows the LabVIEW codes. The programming style for these codes are quite rough and one can definitely re-build another code for it.

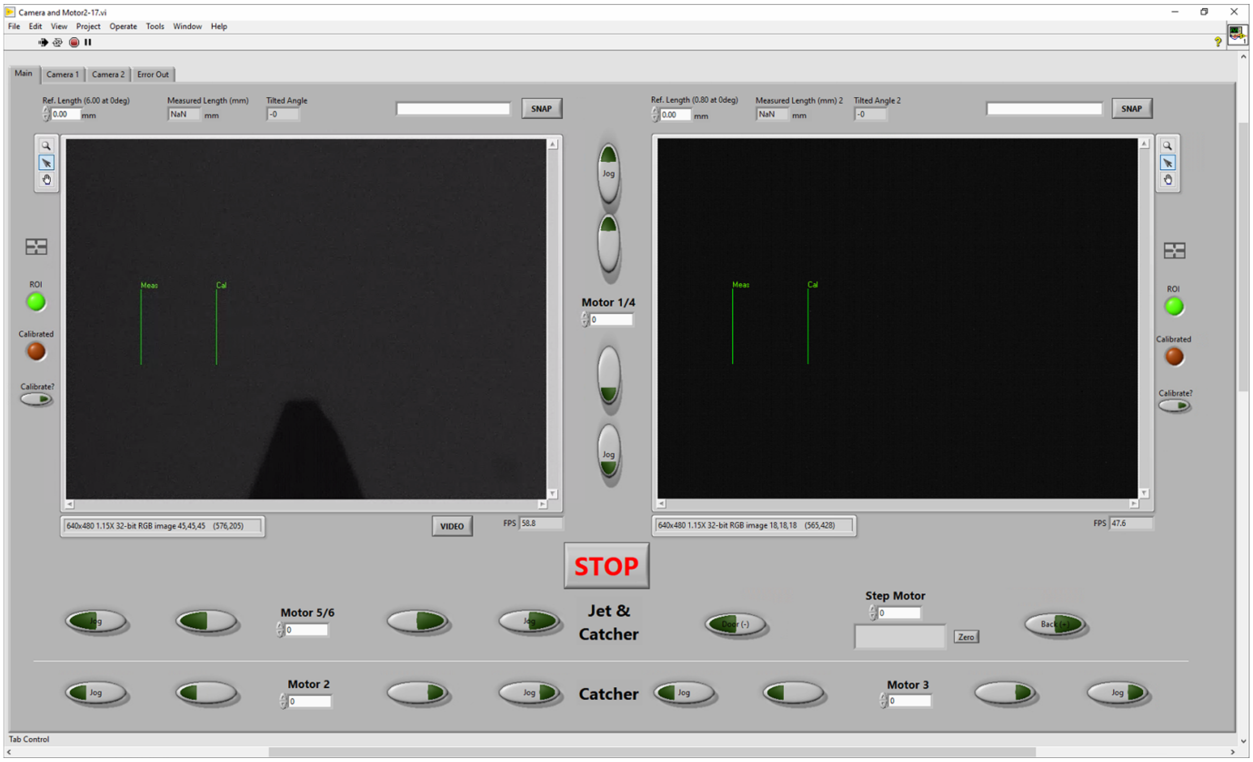
B.1 Liquid Jet Monitoring Program

This program can be found on the jet monitor computer in D10 Latimer as "Camera and Motor2-17.vi". In the front panel, it includes four sub-pages, which are Main, Camera 1, Camera 2 and Error Out. The Main page, shown in Figure B.1, is the major control panel to observe the jet status and control the jet position. Camera pages show a zoom-in view of the two cameras are shown in Figure B.2. The last page is called Error Out, which shows statuses of the devices such as the two cameras, translational stages, and step motor. The layout is shown in Figure B.3.

Front Panel

The main page of the user interface for liquid jet monitoring program is shown in Figure B.1. First, it contains two camera views watching the jet setup on the molecular beam axis and the axis perpendicular to it. The view of the camera can be saved with user defined file name by hitting the SNAP button. The two green lines in the view are used to calibrate and measure the size of the jet. The actual length of the calibrated line (Cal) is defined by a reference length set by user, which is the box shown above the view. After hitting the Calibrate? button besides the camera view, the actual length of the measured line (Meas) is now calibrated by the Cal line and the value is shown in the output box above.

There are also two round buttons shown on the side of the camera view. They are used to show or hide the measured lines (green) and calibrated line (orange) in the view. The



Camera and Motor2-17.vi

Ref. Length (6.00 at 0deg) 0.00 mm

Measured Length (mm) NaN mm

Tilted Angle -0

Ref. Length (0.80 at 0deg) 0.00 mm

Measured Length (mm) 2 NaN mm

Tilted Angle 2 -0

640x480 1.15X 32-bit RGB image 45,45,45 (576,205)

FPS 58.8

640x480 1.15X 32-bit RGB image 18,18,18 (565,428)

FPS 47.6

Motor 1/4 0

Motor 5/6 0

Step Motor 0

Motor 2 0

Motor 3 0

Figure B.1: Main page in the front panel of the liquid jet monitoring LabVIEW program. This is the main panel to control the jet position and monitor the jet status.

button with crossed icon is shown above these round buttons. It adds a red aiming cross onto the camera view to help align the jet to the center of the view. It is noted that both cameras should be aligned to the main chamber before performing any experiments.

At the bottom of the front panel, sets of buttons to control the catcher and jet + catcher set are shown. In each button set for actuator, it includes four buttons for moving the actuator in either jogging or stepping mode. The jogging mode allow user to move the jet only if the button is hit. While the stepping mode move the jet setup with a certain step defined by the box shown in between the buttons. One should note that motors 5/6 and 1/4, share the same controlling button set. The 5/6 motors control either the jet+catcher or the chopper position, while the 1/4 motors control the catcher or jet+catcher position on z-axis. One can select the desired motor through a toggled control box. The actuator control system is set as this because the Newport controller only has four actuator input.

A button set to control the step motor is used to position the jet+catcher setup in the y-axis. Since the step motor uses a open-loop to control, its positioning is reproducible and reliable. Steps to control the moving distance of the motor can be defined in the Step Motor box. A typical value for moving the jet+catcher setup from reaction center to fully out is 900,000 steps. The display box beneath the input box shows the status of the step motor.

A Zero button is shown besides the display box to set the zero-position based on user's preference. The coding for the step motor is buggy that sometimes it doesn't work. If this happens, hit the red STOP button at the center then restart the program.

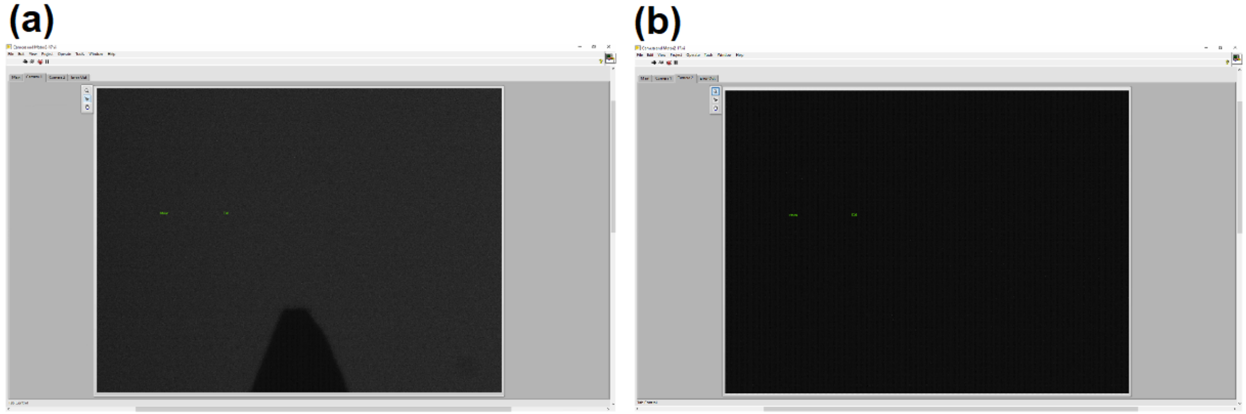
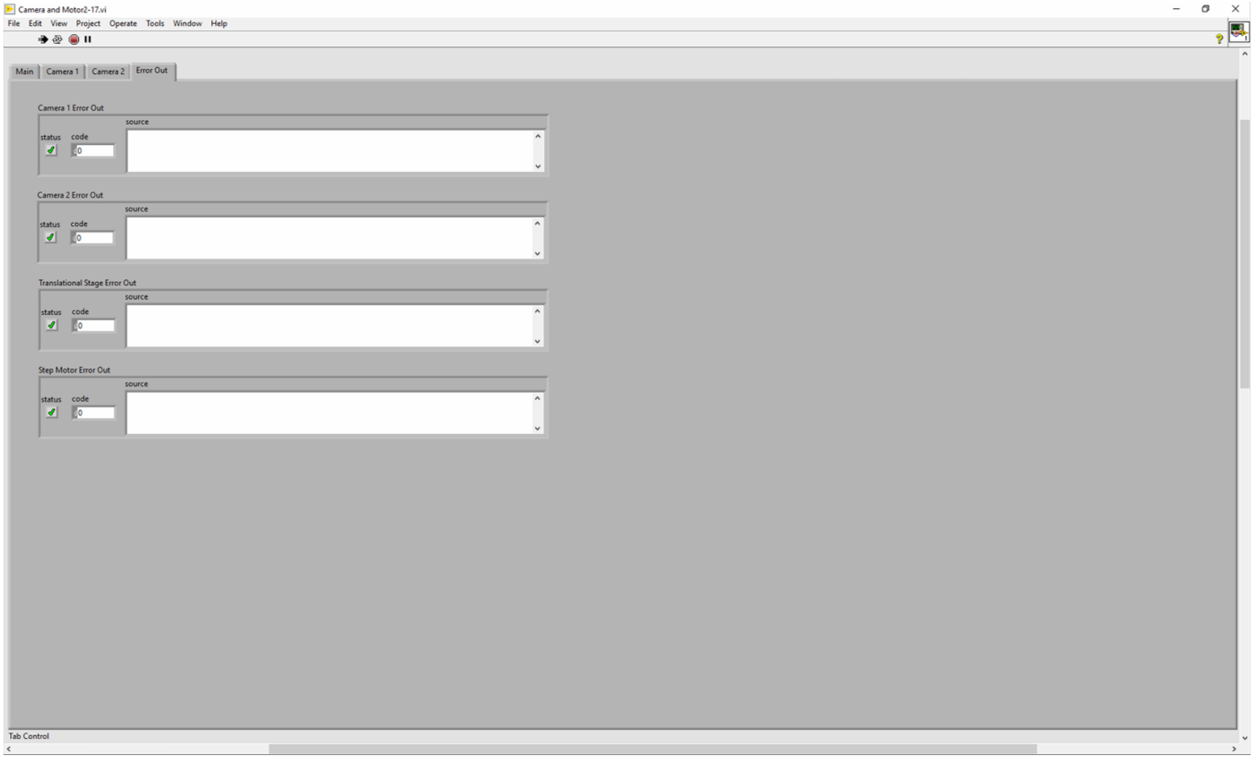


Figure B.2: The Camera pages in the front panel of the liquid jet monitoring LabVIEW program. Page (a) Camera 1 and (b) Camera 2 show zoom-in view of the camera looking at the y axis (perpendicular to molecular beam axis) and x axis (the molecular beam axis).

Figure B.2 shows a zoom-in view of the two camera views. A red aiming cross can be shown on the views if the aiming button is hit on the Main page. These two pages are specifically written in the program for easier alignment processing. Traditionally, the molecular beam alignment is done by using a telescope, which is tedious to setup and requires two people to perform the beam alignment. Now with the cameras, they are sitting on translational stages with tilting abilities, which make them serve as telescopes. Therefore, one can now align the beam by him/herself without running back and forth between the telescope and molecular beam setup.

Finally, the Error Out page is shown in Figure B.3, which shows the normal condition when all the devices are working properly. If any of the status boxes shows red cross, it means something is wrong with the device that program cannot run the function. This is usually caused by bad connections or wrong device address in the block diagram.

The rotational stage to control the jet angle is not included in this program. It is controlled by its own controlling software from the company, Standa. Standa also provides LabVIEW codes for user to use it when programming. I tried it and it did not work. The rotational stage controller only connected to this LabVIEW program when the program is first started. Customer service from Standa couldn't resolved this issue and therefore the rotational stage is than working independently.



Camera 1 Error Out
status ✓ code 0 source

Camera 2 Error Out
status ✓ code 0 source

Translational Stage Error Out
status ✓ code 0 source

Step Motor Error Out
status ✓ code 0 source

Figure B.3: The Error Out page in the front panel of the liquid jet monitoring LabVIEW program. It shows status of controlling devices.

Block Diagram

The block diagram is shown in Figure B.4. It can be classified into three parts depends on the type of the device. The first part is the camera blocks, which locate at the top of the block diagram. The cameras are controlled using virtual instrument (VI) from the company to call the camera, which the name of the VI is ICImagingControl. Then, the video from the camera will be processed by VIs in the Vision Development Module in LabVIEW, which is an added-on module that requires user to download. A zoom-in view of the block diagram for a camera 1 is shown in Figure B.5. Camera 2 has the same diagram except that there's no record option. Since the region of interest (ROI) manipulations are used in the camera controlling system, initial values for ROI need to be set. Its block diagram is shown in Figure B.6. Block diagrams of the translational stages and step motor controlling system are shown in Figure B.7 and Figure B.8, respectively.

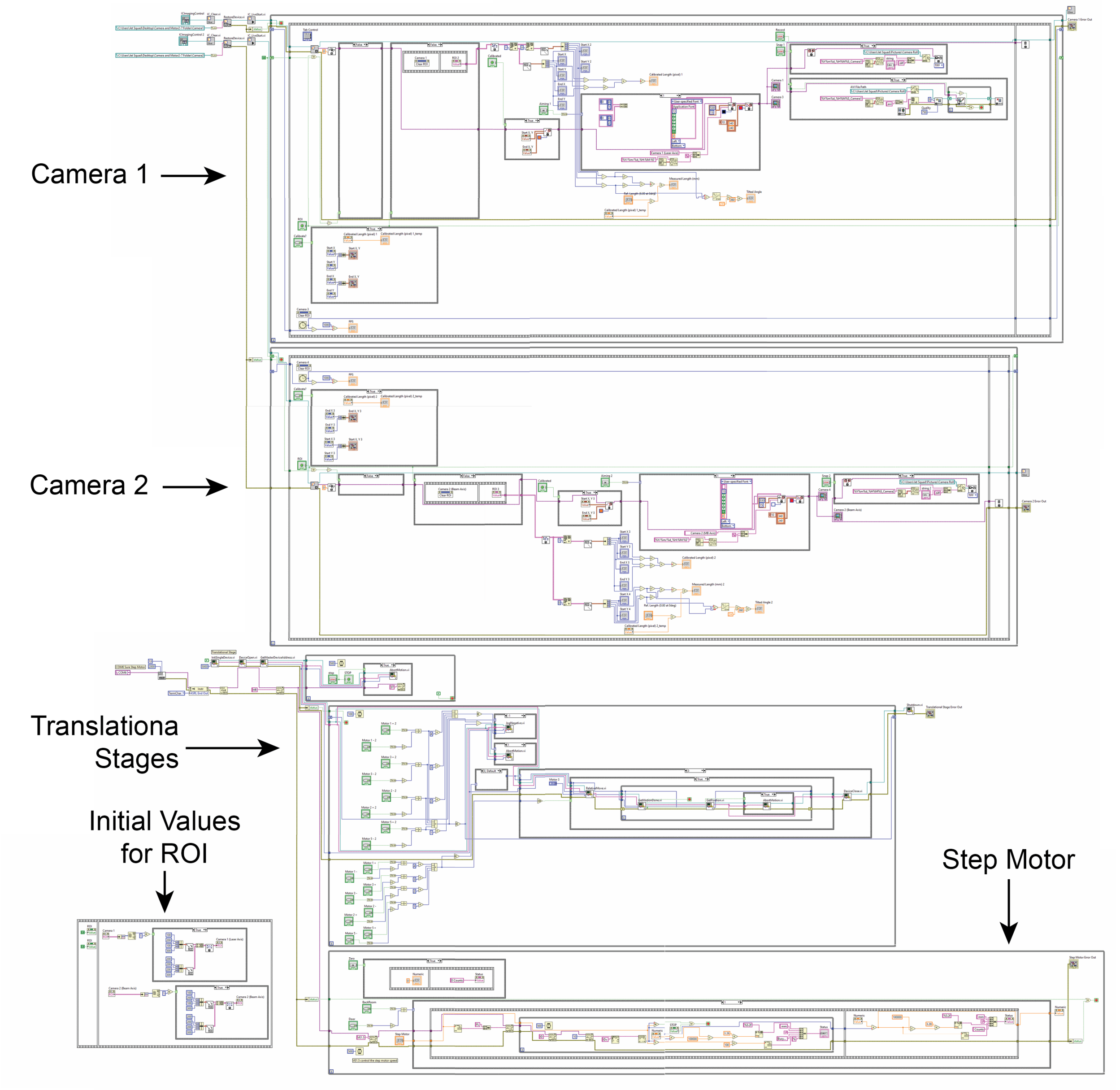


Figure B.4: The entire block diagram of the liquid jet monitoring LabVIEW program. It includes controlling program for camera 1, camera 2, translational stages and step motor. Initial values of ROI are set for camera programs to run.

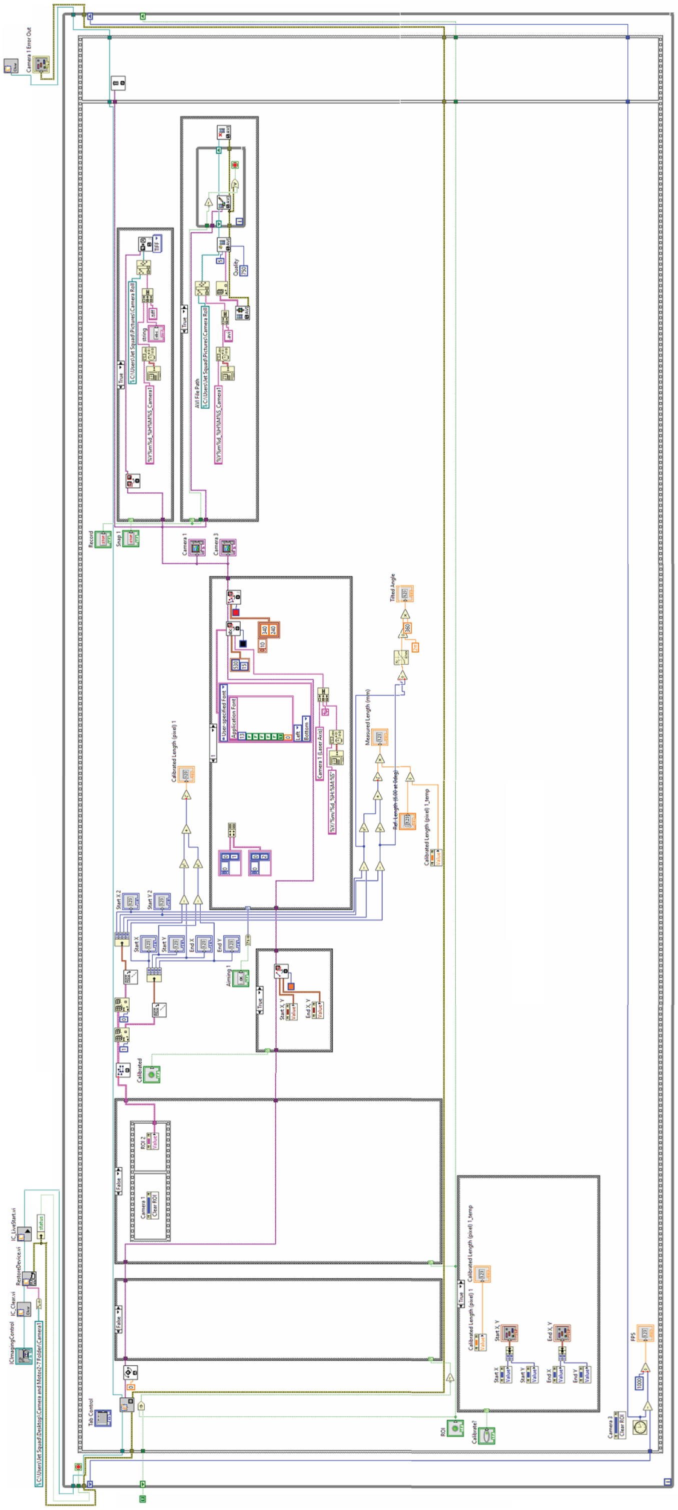


Figure B.5: Block diagram of camera controlling program.

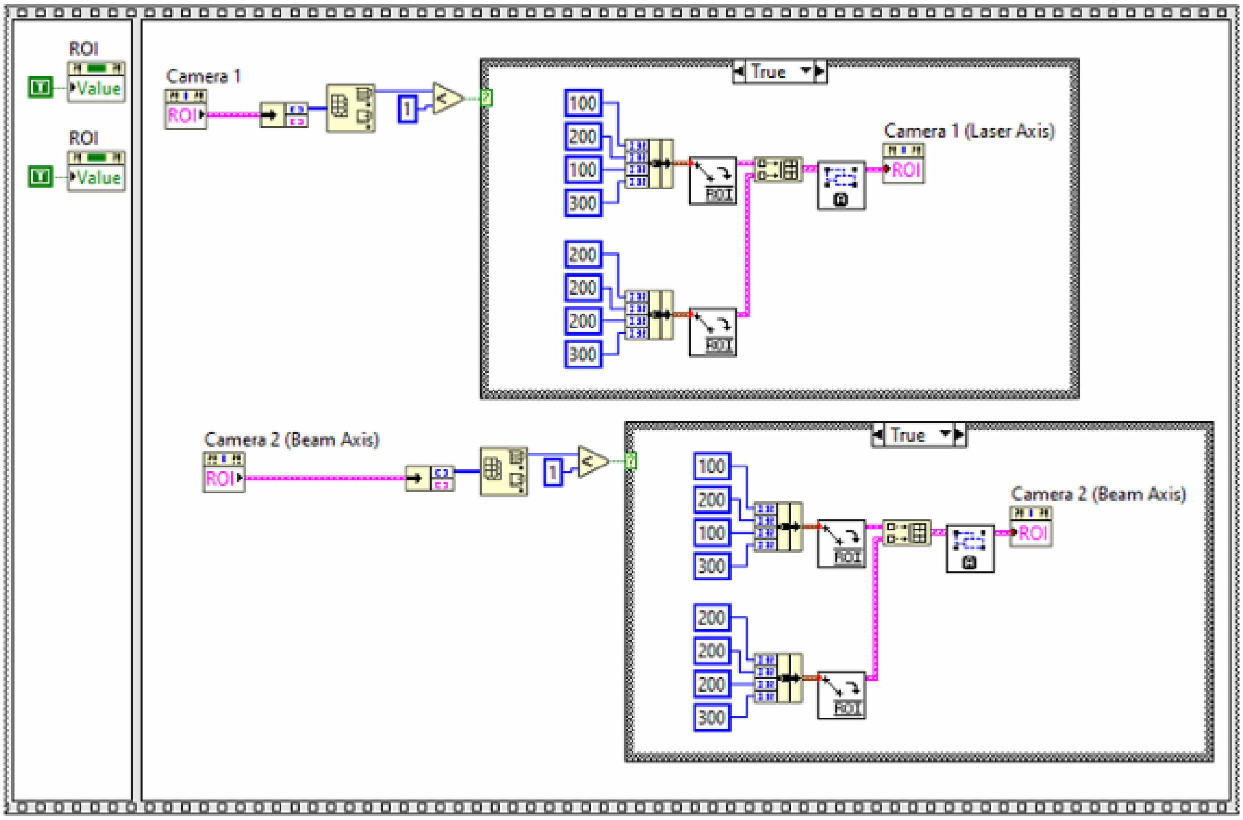


Figure B.6: Block diagram to set initial values for region of interest (ROI) in camera controlling program.

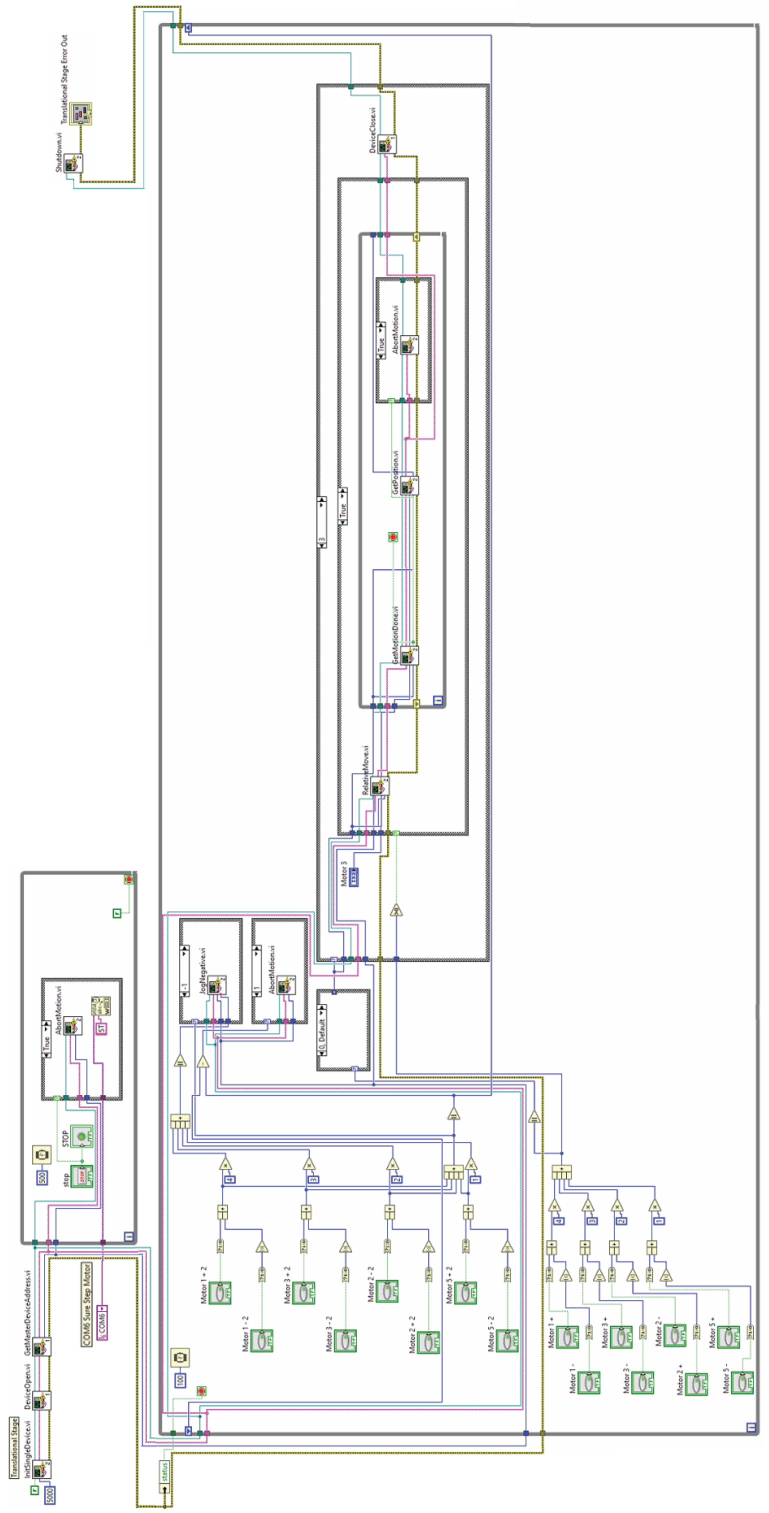


Figure B.7: Block diagram of translational stages controlling program.

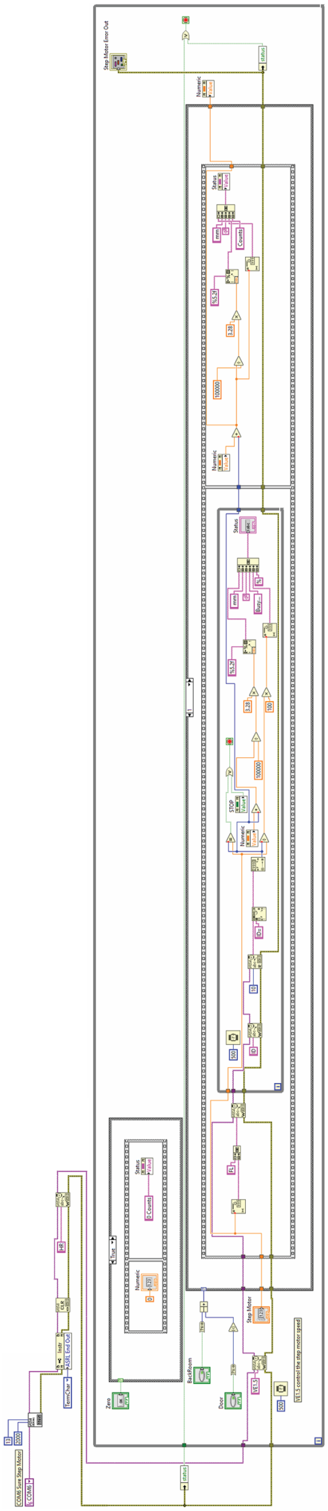


Figure B.8: Block diagram of step motor controlling program.

B.2 Pressure Monitoring Program

The pressure monitoring program is stored on the data acquisition computer in D10 Latimer as ”Pressure Monitor.vi”. It shows real time pressures in chambers of main, source, RI, RII and RIII. Two main reasons give motivation on developing this program. First is because that we want to remotely monitor the chamber’s pressures and record them down. This is especially useful when people are not around the lab for a long time, such as winter break. A 1-month lockdown occurred in May of year 2020 due to COVID-19 pandemic. Although the chemistry department at UC Berkeley is one of the departments that first has access to the campus after lockdown, the work time is still limited. Therefore, having a pressure monitoring program that can be remotely checked through internet is extremely useful. Plus, if something goes wrong during off-hours, for example, batteries in the turbo controller drains out at midnight, one can trace back to when this happened by checking the records.

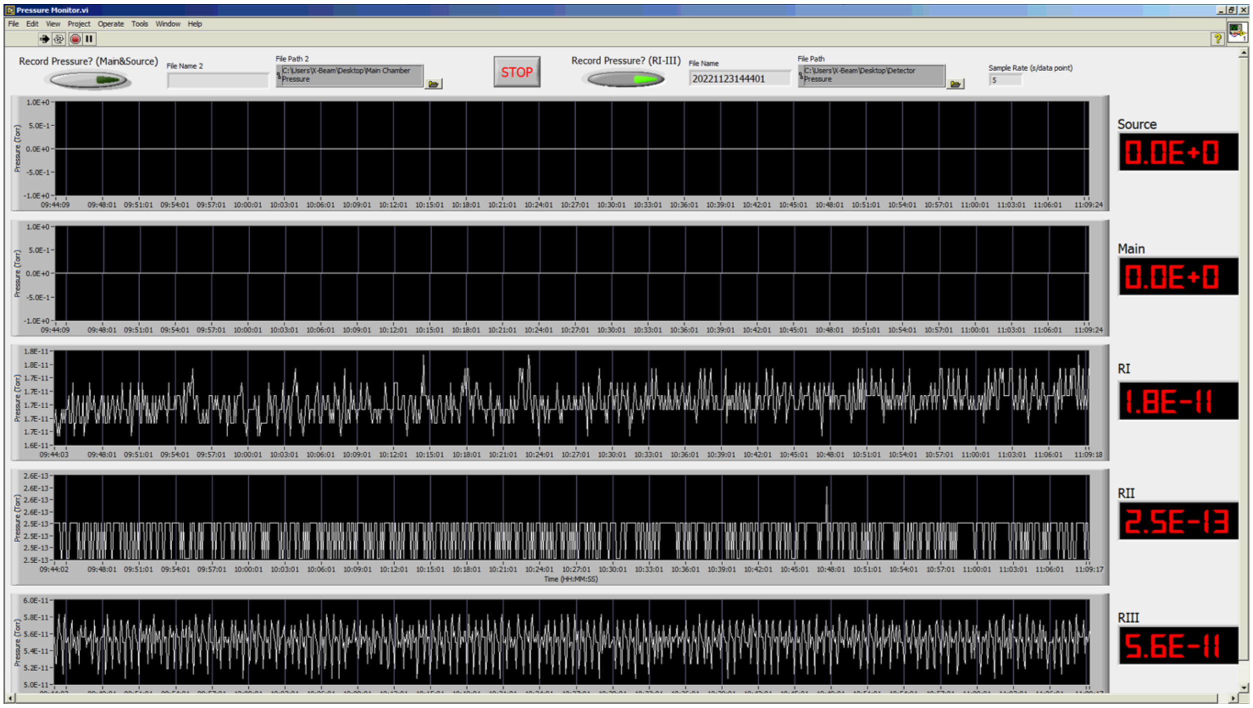
The second motivation to build this program is to monitor the jet condition with digital values recorded. One of the difficulties to run a liquid jet under vacuum is that if the jet misaligned or freeze up, it will generate large vapor burst and might damage the electronics in the detector. Having a real time pressure monitoring plot helps these accidental situations being observed in an early stage, and therefore damages can be limited. Although one can use the cameras views shown in the jet monitoring program to keep tracking the jet condition, it is much easier and more sensitive for one to realize jet accidents by seeing a "peak" in the pressure plots.

Front Panel

The user interface of the pressure monitoring program is shown in Figure B.9. Five vacuum chambers’ pressures are monitored. Each data point shown in the plot is actually an average value of five acquired data obtained from a NI data acquisition device (NI USB-6008). One can change the average number in the block diagram.

Pressures of the source and main chambers are detected through ion gauges with Terranova controller model 934. Since these two pressures are only useful when the main chambers are pumped down during the experiments, they do not require to be always recorded. Therefore, they share the same pressure recording system where the starting button is shown on the top left. Detector’s pressures, on the other hand, are always recorded in case something bad happens to the detector’s turbos.

One should note that this program is buggy. The pressure recording button only works when the program is first turned on and cannot restart pressure recording when the program is running. To restart recording, one needs to hit the STOP button then restart the program. After doing so, the recording can then turn on again.



File Path 2: C:\Users\X-Beam\Desktop\Main Chamber Pressure

File Name: 20221123144401

File Path: C:\Users\X-Beam\Desktop\Detector Pressure

Sample Rate (s/data point): 5

Figure B.9: LabVIEW interface of pressure monitoring program.

Block Diagram

Block diagram for the pressure monitor program is shown in Figure B.10. The pressure acquisition code is mainly based on the code written by Mark Shapero in the "Pressure and Cold Head" program [80].

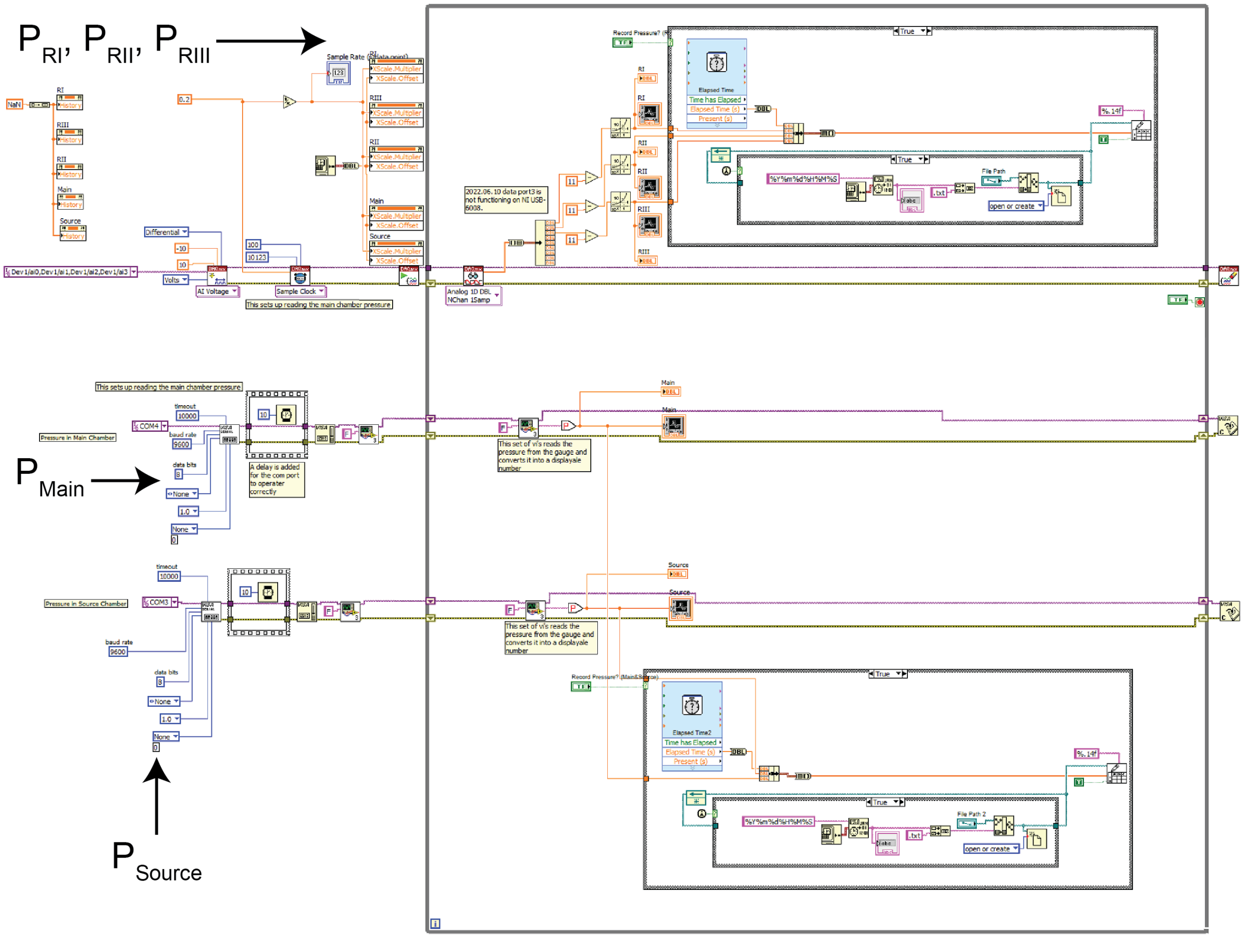

P_{RI}, P_{RII}, P_{RIII}
P_{Main}
P_{Source}
Sample Rate
This sets up reading the main chamber pressure
2022.06.10 data port3 is not functioning on NI USB-6008.
Record Pressure?
Elapsed Time
File Path
open or create
Pressure in Main Chamber
A delay is added for the com port to operater correctly
This set of vi's reads the pressure from the gauge and converts it into a displayale number
Pressure in Source Chamber
Record Pressure? (Main&Source)
Elapsed Time2
File Path 2

Figure B.10: Block diagram of pressure monitoring program. It includes three parts of code to record pressure at the three differential pumping regions in the detector chamber (P_{RI}, P_{RII} and P_{RIII}), the main chamber (P_{Main}) and the source chamber (P_{Source}).

Appendix C

Data Analysis Codes

The data analyses performed in this dissertation are analyzed by codes written in Python 3.0. Experiments of evaporation and scattering have their own code to import data then analyze with different processes. Code for kinematic model uses the results analyzed from the scattering code, which requires manually imported. Details of how to use the code are written in the code itself. Thus, only brief introductions are described in the text content.

C.1 Evaporation Experiment

Two type of evaporation codes are used in this dissertation. A single supersonic distribution is used to fit dodecane evaporation data. While for neon evaporates from liquid dodecane, two Maxwell-Boltzmann distributions are used to fit with the experimental data. The temperature in one of the MB distributions is set to be at liquid temperature, while the other one is left unfixed that the program will find the best fitting temperature, which is the T_{bkg} mentioned in this dissertation.

Supersonic Distribution Fitting

This code uses a supersonic distribution to fit the input TOF. Equation 4.2 is used and shown as

$$N_{SS,meas}(t) \propto \frac{1}{t^4} \exp\left[-\frac{m(\frac{L'}{t} - v_{SS})^2}{2RT_{SS}}\right].$$

```
##########################################################################
# <INTRO>:                                                               #
# This script is used to fit the evaporation TOF using supersonic (SS)   #
# distribution. This is specifically useful for high collision number    #
# above the liquid surface. It gives the fitting parameters of           #
# temperature (T) and flow velocity (v0). These numbers should be used   #
# in the scattering data for estimating the TD part.                     #
#                                                                        #
# <INPUT>:                                                               #
# The input datas need to be beam on/off subtracted and intensity        #
# calibrated. Time interval in each data are default as 1 us. The input  #
# data should be saved in a folder, which is at the same file path as    #
# the script.                                                            #
#                                                                        #
# <OUTPUT>:                                                              #
# The output data are in the variables. Only the important ones are      #
# shown:                                                                 #
#                                                                        #
#    xData:            flight time (us)                                  #
#    yData:            loaded data                                       #
#    yData_norm:       normalized yData                                  #
#    yFit_MB:          fitted Maxwell-Boltzmann profile of yData         #
#    yFit_MB_norm:     normalized yFit_MB                                #
#    yFit_SS:          fitted supersonic profile of yData                #
#    yFit_SS_norm:     normalized yFit_SS                                #
#    yFit_SS_param:    parameters of yFit_SS                             #
#                                                                        #
#    xCOS:             radian of angular distribution (radian)           #
#    yArea_SS:         area of yFit_SS                                   #
#    yArea_SS_norm:    normalized yArea_SS                               #
#    yCOS:             fitted Cosine distribution                        #
#    yCOS_norm:        normalized yCOS                                   #
#                                                                        #
#    xAngle:           angle of the out going beam (radian)              #
#    E_mean:           mean translational energy of yFit_SS              #
#    S:                speed ratio of yFit_SS                            #
#                                                                        #
# <OUTLINE>:                                                             #
#       PARAMETERS                                                       #
#    0. Initialize Enviornment                                           #
#    1. Data Loading                                                     #
#    2. Data Fitting (SS)                                                #
#    3. Angular Distribution                                             #
#    4. Mean Translational Energy and Speed Ratio                        #
#    5. Data Processing                                                  #
##########################################################################





##########----------------------PARAMETERS----------------------##########
folderName = "m57"
# name of the folder that includes molecular beam data
evapName = "Dodecane Evaporation"
# name of the evaporated beam
```

```python
xAngle = [0,30,60,90,0,80,50,20,0,10,40,70,0,75,45,15,0,90]
# outgoing angle of each data. the sequence needs to be the same as the file sequence
L = 0.1716          # unit: m, flight distance, default should be 0.1716 m

# Maxwell-Boltzmann (MB) and Supersonic (SS) distribution fitting parameters
m_MB = 170          # unit: amu
m_SS = 170          # unit: amu
m_ion = 57          # unit: amu
dm = 0.0001         # unit: amu

T_MB = 289          # unit: K
T_SS = 0            # unit: K
dT_SS = 1000        # unit: K
##########--------------------------------------------------------##########



##########---------------0. Initialize Environment--------------##########
from IPython import get_ipython
get_ipython().magic('reset -sf')

import pathlib
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import numpy as np
import math
import os
##########--------------------------------------------------------##########



##########--------------------1. Data Loading--------------------##########
# Function to load data
def LoadData(folderName):
    filePath = str(pathlib.Path().resolve())+str("\\")+folderName
    fileName = os.listdir(filePath)
    yDataRaw = []
    for i in range(len(fileName)):
        file = os.path.join(filePath , fileName[i])
        yDataRaw.append(np.loadtxt(file))
    return fileName, yDataRaw

# Load raw data into Python. Prepare xData and yData with time offset.
fileName, yDataRaw = LoadData(folderName)
yData  = yDataRaw
t_offset = (-6 + 19.5 + 4.8*(m_ion)**0.5)
xData  = np.array([i - t_offset for i in range(len(yData[0]))])

```

```python
Nfile  = len(fileName) # int, number of files
NxData = len(xData)    # list, number of x in xData
##########------------------------------------------------------##########



##########-----------------2. Data Fitting (SS)-----------------##########
# Define Maxwell-Boltzmann (MB) distribution
def MB(t, m, T, A):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6))**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

# Define supersonic (SS) distribution
def SS(t, m, T, A, v0):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6)-v0)**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

# Fit distribution with SS. MB distributions are also shown for comparison
yFit_MB = []
yFit_SS = []
yFit_SS_param = []
for i in range(Nfile):
    y=yData[i]

    # Fit with a SS ditribution at a defined range.
    popt_SS,pcov_SS = curve_fit(SS,xData,y,p0=[m_SS,270,max(y),500],bounds=([m_SS,0,0,0],[m_SS+dm,T_SS+dT_SS,100000,2000]))

    # MB distribution is generated with the same height of yFit_SS for comparison.
    yFit_MB.append(MB(xData,m_MB,T_MB,popt_SS[2]))
    yFit_SS.append(SS(xData,*popt_SS))
    yFit_SS_param.append(popt_SS)

# Plotting
plt.figure(figsize=(16,8))
for i in range(Nfile):
    plt.subplot(2, math.ceil(Nfile/2), i+1)

    plt.plot(xData,yData[i],  'ko',markersize=1,label='Data')
    plt.plot(xData,yFit_MB[i],'b-',markersize=1,label='MB')
    plt.plot(xData,yFit_SS[i],'r-',markersize=1,label='SS')
    plt.xlim(0,2000)
    plt.title("{}\n ({})".format(evapName,fileName[i]),fontsize=10)
plt.tight_layout()
```

```

# Noarmalizing the data
yData_norm   = [[i/max(yFit_SS[j]) for i in yData[j]]for j in range(Nfile)]
yFit_SS_norm = [[i/max(yFit_SS[j]) for i in yFit_SS[j]]for j in range(Nfile)]
yFit_MB_norm = [[i/max(yFit_MB[j]) for i in yFit_MB[j]]for j in range(Nfile)]
##########------------------------------------------------------##########



##########----------------3. Angular Distribution---------------##########
# Calculate the contribution of SS from fitted data.
yArea_SS = []
for i in range(len(xAngle)):
    Area_SS=sum(yFit_SS[i]/xData)
    yArea_SS.append(Area_SS)

# Cosine distribution is normalized at outgoing angle=0deg.
xAngle = [xAngle[i]/360*2*np.pi for i in range(Nfile)] # xAngle is now in unit of radian
xCOS = [i/360*2*np.pi for i in range(0,95,5)]
yCOS_max = sum([yArea_SS[i] for i in range(Nfile) if xAngle[i]==0])/len([yArea_SS[i] for i in range(Nfile) if xAngle[i]==0])
yCOS = [np.cos(xCOS[i])*yCOS_max for i in range(len(xCOS))]

yArea_SS_norm = [yArea_SS[i]/yCOS_max for i in range(len(yArea_SS))]
yCOS_norm = [yCOS[i]/yCOS_max for i in range(len(yCOS))]

# Plotting
plt.figure()
Angular = plt.subplot(111,polar=True)
Angular.plot(xAngle,yArea_SS,’ro’)
Angular.plot(xCOS,  yCOS,    ’k--’)
Angular.set_thetalim(0,np.pi/2)
Angular.set_theta_zero_location("N")
Angular.set_theta_direction(-1)
##########------------------------------------------------------##########



##########-----4. Mean Translational Energy and Speed Ratio-----##########
# Calculate fitted SS’s E_mean (unit=kJ/mol) and S=(speed ratio)**2
E_mean = []
S = []
xTime      = [i for i in range(1,2501)]              # unit: us
xVelocity = [L/(t*10**-6) for t in xTime]           # unit: m/s
yEnergy   = [0.5*m_SS/1000*v**2 for v in xVelocity] # unit: J/mol
for i in range(Nfile):
    m  = yFit_SS_param[i][0]
    T  = yFit_SS_param[i][1]
    v0 = yFit_SS_param[i][3]

    # Below, SS function in the velocity space is used for yIntensity. No Jacobian is required. See Comsa's 1985 paper
    yIntensity = [1/T*np.exp(-m/1000*(v-v0)**2/(2*8.314*T))*v**3 for v in xVelocity]

    E_temp_1 = [(xVelocity[j]-xVelocity[j+1])*(yIntensity[j]+yIntensity[j+1])/2*(yEnergy[j]+yEnergy[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_temp_2 = [(xVelocity[j]-xVelocity[j+1])*(yIntensity[j]+yIntensity[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_mean_temp = sum(E_temp_1)/sum(E_temp_2)/1000 #kJ/mol
    E_mean.append(E_mean_temp)

    S_temp = yFit_SS_param[i][0]/1000*yFit_SS_param[i][3]**2/(2*8.314*yFit_SS_param[i][1])
    S.append(S_temp)
##########-------------------------------------------------------##########



##########------------------5. Data Processing------------------##########
xAngle = np.reshape(xAngle,(1,Nfile)).T
yDataRaw = np.reshape(yDataRaw,(Nfile,NxData)).T
yData = np.reshape(yData,(Nfile,NxData)).T
yData_norm = np.reshape(yData_norm,(Nfile,NxData)).T
yFit_SS = np.reshape(yFit_SS,(Nfile,NxData)).T
yFit_SS_norm = np.reshape(yFit_SS_norm,(Nfile,NxData)).T
yFit_SS_param = np.reshape(yFit_SS_param,(Nfile,np.size(yFit_SS_param[0]))).T
yFit_MB = np.reshape(yFit_MB,(Nfile,NxData)).T
yFit_MB_norm = np.reshape(yFit_MB_norm,(Nfile,NxData)).T

# Deleting parameters that are not important. If wish to debug, unfunction this line.
del Angular,Area_SS,evapName,dm,dT_SS,E_mean_temp,E_temp_1,E_temp_2,fileName,folderName,i,L,m,m_MB,m_SS,m_ion,Nfile,NxData,pcov_SS,popt_SS,S_temp,T,T_MB,T_SS,t_offset,v0,xTime,xVelocity,y,yCOS_max,yEnergy,yDataRaw,yIntensity
##########-------------------------------------------------------##########
```

Listing C.1: Python code to fit evaporating experiments with supersonic distribution.

This code outputs two figures. One is the fitting result of the TOF distributions and the other is the angular distribution. Examples for dodecane evaporation are fit and shown in Figure C.1 and C.2. The TOF distributions shown in Figure 4.3 (a) are the normalized experimental data, supersonic distribution, and Maxwell-Boltzmann distribution, which are results calculated by this code listed in variables `yData_norm`, `yFit_SS_norm` and `yFit_MB_norm`. The red squares shown in Figure 4.5 represent the angular distribution of dodecane evaporation and the data is from variable `yArea_SS_norm`.

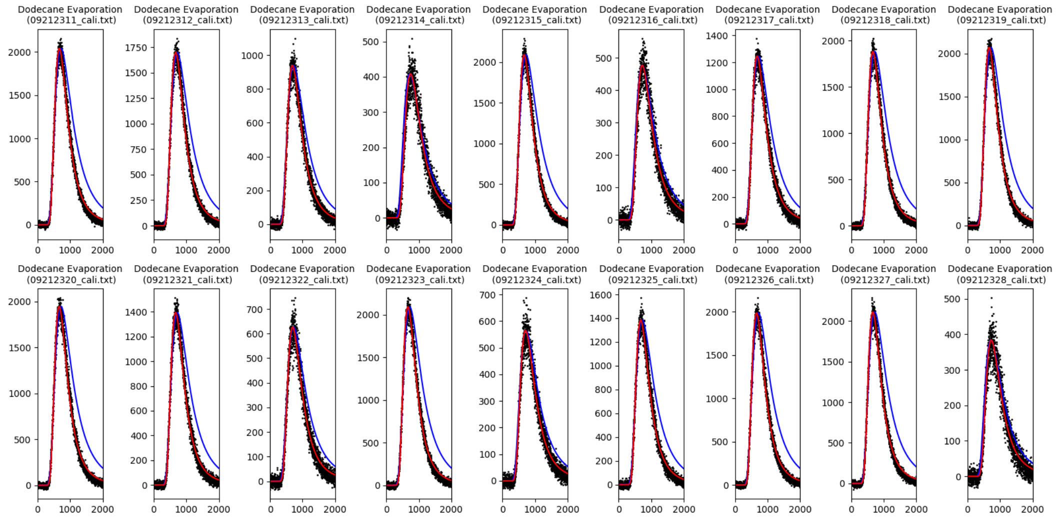


Figure C.1: TOF fitting results using a supersonic distribution in evaporation experiment. Supersonic fittings are shown as red curves, while the Maxwell-Boltzmann distributions at liquid temperature are also shown in blue curves for reference.

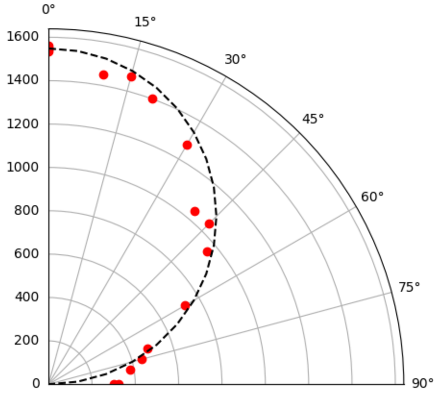


Figure C.2: Angular distribution fitting results using a supersonic distribution in evaporation experiment. Supersonic fitting results are shown as red circles, while the cosine distribution is shown in black dash curve for reference.

Two Maxwell-Boltzmann Distributions Fitting

This code uses two Maxwell-Boltzmann distributions to fit the input TOF. Equation 4.3 is used and shown as

$$N_{MB,MB}(t) = \frac{c_{liq}}{t^4}\exp\left[-\frac{m(\frac{L'}{t})^2}{2RT_{liq}}\right] + \frac{c_{bkg}}{t^4}\exp\left[-\frac{m(\frac{L'}{t})^2}{2RT_{bkg}}\right]$$

where c_{bkg} is constant at all the outgoing angles.

```
##########################################################################
# <INTRO>:                                                               #
# This script is used to fit the evaporation TOF using a constant        #
# Maxwell-Boltzmann distribution at liquid temperature with a constant   #
# background at all detector angle. The background is setted at a fitted #
# temperature (T_bkg) which could be the temperature of the cold Cu wall.#
# The T_bkg and the intensity (A_bkg) are set the same for all the       #
# detector angle. This code can be used to explain the slower TOF in the #
# evaporation experiment, which is from gas molecules that can not be    #
# froze onto the Cu wall.                                                #
#                                                                        #
# <INPUT>:                                                               #
# The input datas need to be beam on/off subtracted and intensity        #
# calibrated. Time interval in each data are default as 1 us. The input  #
# data should be saved in a folder, which is at the same file path as the#
# script.                                                                #
#                                                                        #
# <OUTPUT>:                                                              #
# The output data are in the variables. Only the important ones are      #
# shown:                                                                 #
#                                                                        #
#    xData:           flight time (us)                                   #
#    yData:           loaded data                                        #
#    yData_norm:      normalized yData                                   #
#    yFit_MB:         Maxwell-Boltzmann part of the fitted profile for   #
#                     yData                                              #
#    yFit_MB_norm:    normalized yFit_MB                                 #
#    yFit_MB_param:   parameters of yFit_MB                              #
#    yFit_bkg:        background part of the fitted profile for yData    #
#    yFit_bkg_norm:   normalized yFit_bkg                                #
#    yFit_bkg_param:  parameters of yFit_bkg                             #
#    yFit_MB_bkg:     fitted profile of yData (yFit_MB+yFit_bkg)         #
#    yFit_MB_bkg_norm:normalized yFit_MB_bkg                             #
#                                                                        #
#    xCOS:            radian of angular distribution (radian)            #
#    yArea_MB:        area of yFit_MB                                    #
#    yArea_MB_norm:   normalized yArea_MB                                #
#    yArea_bkg:       area of yFit_bkg                                   #
#    yArea_bkg_norm:  normalized yArea_bkg                               #
#    yCOS:            fitted Cosine distribution                         #
```

```python
#    yCOS_norm:       normalized yCOS                                     #
#                                                                         #
#    xAngle:          angle of the out going beam (radian)                #
#    E_mean:          mean translational energy of yFit_SS                #
#    S:               speed ratio of yFit_SS                              #
#                                                                         #
# <OUTLINE>:                                                              #
#       Parameters                                                        #
#    0. Initialize Enviornment                                            #
#    1. Data Loading                                                      #
#    2. Normalize Data                                                    #
#    3. Data Fitting (MB + const MB bkg)                                  #
#    4. Angular Distribution                                              #
#    5. Mean Translational Energy and Speed Ratio                         #
#    6. Data Processing                                                   #
###########################################################################





##########----------------------PARAMETERS-----------------------##########
folderName = "Ne"           # name of the folder that includes molecular beam data
evapName = "Ne Evaporation" # name of the evaporated beam
xAngle = [0,30,60,90,0,80,50,20,0,10,40,70,0,75,45,15,0,90] # out going angle of each data. the sequence needs to be the same as the file sequence
L = 0.1716                   # m, flight distance, default should be 0.1716 m

# Maxwell-Boltzmann (MB) and Supersonic (SS) distribution fitting parameters
m_MB = 20             # amu
m_bkg = 20            # amu
m_SS = 20             # amu
m_ion = 20            # amu
dm = 0.0001           # amu

T_MB = 284            # K
T_SS = 0              # K
dT_SS = 1000          # K
dT_MB = 0.0001        # K
dT_MB_const_bkg = 200 # K
##########-------------------------------------------------------##########



##########---------------0. Initialize Enviornment---------------##########
from IPython import get_ipython
```

```python
get_ipython().magic('reset -sf')

import pathlib
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import numpy as np
import math
import os
##########------------------------------------------------------##########



##########--------------------1. Data Loading-------------------##########
# Function to load data
def LoadData(folderName):
    filePath = str(pathlib.Path().resolve())+str("\\")+folderName
    fileName = os.listdir(filePath)
    yDataRaw = []
    for i in range(len(fileName)):
        file = os.path.join(filePath , fileName[i])
        yDataRaw.append(np.loadtxt(file))
    return fileName, yDataRaw

# Load raw data into Python. Prepare xData and yData with time offset.
fileName, yDataRaw = LoadData(folderName)
yData = yDataRaw
t_offset = (-6 + 19.5 + 4.8*(m_ion)**0.5)
xData = np.array([i - t_offset for i in range(len(yData[0]))])

Nfile = len(fileName) # int, number of files
NxData = len(xData)   # list, number of x in xData
##########------------------------------------------------------##########



##########------------------2. Normalize Data-------------------##########
# Define supersonic (SS) distribution
def SS(t, m, T, A, v0):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6)-v0)**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

# Fit distribution with SS. MB distributions are also shown for comparison
yFit_SS = []
yFit_SS_param = []
for i in range(Nfile):
    y=yData[i]

    # Fit with a SS ditribution at a defined range.
    popt_SS,pcov_SS = curve_fit(SS,xData,y,p0=[m_SS,270,max(y),500],bounds=([m_SS,0,0,0],[m_SS+dm,T_SS+dT_SS,100000,2000]))

    yFit_SS.append(SS(xData,*popt_SS))
    yFit_SS_param.append(popt_SS)

# Plotting
plt.figure(figsize=(16,8))
for i in range(Nfile):
    plt.subplot(2, math.ceil(Nfile/2), i+1)

    plt.plot(xData,yData[i],  'ko',markersize=1,label='Data')
    plt.plot(xData,yFit_SS[i],'r-',markersize=1,label='norm')
    plt.xlim(0,1500)
    plt.title("{}\n ({})".format(evapName,fileName[i]),fontsize=10)
plt.tight_layout()

# Noarmalizing yData and yFit_SS
yData_norm = [[i/max(yFit_SS[j]) for i in yData[j]]for j in range(Nfile)]
yFit_SS_norm = [[i/max(yFit_SS[j]) for i in yFit_SS[j]]for j in range(Nfile)]
##########-------------------------------------------------------##########



##########----------3. Data Fitting (MB + const MB bkg)---------##########
# The Data are flatten to 1D in order to do so. Area of each data is recorded in order to keep the intensity correct.
xData_1D = np.tile(xData,Nfile)
yData_1D = np.concatenate(yData)
yData_norm_1D = np.concatenate(yData_norm)

# Define Maxwell-Boltzmann (MB) distribution
def MB(t, m, T, A):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6))**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

# Define a two Maxwell-Boltzmann fittings with one of the MB is intensity constant in all the angles.
# This function is used to fit normalized data, which means it consider all the angles with same weighting.
# Nfile, NxData, dm, dT_MB, and yFit_SS_param need to be provided before using this function.
def MB_bkg(t, m_MB, T_MB, m_bkg, T_bkg, A_bkg):
    temp = 0
    yFit_MB_bkg_norm = []
    for i in range(Nfile):
        x = xData_1D[temp:temp+NxData]
        y = yData_1D[temp:temp+NxData] # must use yData_1D instead of yData_norm_1D

        # First generate a bkg function
        yFit_bkg = MB(x, m_bkg, T_bkg, A_bkg)

        # Then Fit y-yFit_bkg with a MB ditribution at a defined range.
        popt_MB,pcov_MB = curve_fit(MB,x,y-yFit_bkg,p0=[m_MB,T_MB,max(y-yFit_bkg)],bounds=([m_MB,T_MB,0],[m_MB+dm,T_MB+dT_MB,100000]))
        yFit_MB  = MB(x, *popt_MB)

        yFit_MB_bkg = yFit_MB+yFit_bkg

        # Convert yFit_MB_bkg back to normalized data
        yFit_MB_bkg_norm.append(yFit_MB_bkg/yFit_SS_param[i][2])

        temp=temp+NxData
    y=np.concatenate(yFit_MB_bkg_norm)
    return y

# Find the right constant intensity (A_bkg in function MB_bkg) and temperature (T_bkg in function MB_bkg) for the constant MB bkg.
# The input data must be normalized 1D yData (yData_norm_1D)
popt_MB_bkg, pcov_MB_bkg = curve_fit(MB_bkg,xData_1D,yData_norm_1D,bounds=([m_MB,T_MB,m_bkg,0,0],[m_MB+dm,T_MB+dT_MB,m_bkg+dm,300,5000]))

# Generate the fitted MB and constant bkg curves for all the angles.
yFit_MB = []
yFit_bkg = []
yFit_MB_bkg = []
yFit_MB_param = []
yFit_bkg_param = []
yFit_MB_norm = []
yFit_bkg_norm = []
yFit_MB_bkg_norm = []
for i in range(Nfile):
    x = xData
    y = yData[i]

    y_bkg = MB(x,*popt_MB_bkg[2:])

    # Fit with a MB ditribution at a defined range.
    popt_MB,pcov_MB = curve_fit(MB,x,y-y_bkg,p0=[m_MB,T_MB,max(y-y_bkg)], bounds=([m_MB,T_MB,0],[m_MB+dm,T_MB+dT_MB,100000]))
    y_MB  = MB(x, *popt_MB)

    # Storing fitted curves and parameters.
    yFit_MB.append(y_MB)
    yFit_bkg.append(y_bkg)
    yFit_MB_bkg.append(y_MB+y_bkg)
    yFit_MB_param.append(popt_MB)
    yFit_bkg_param.append(popt_MB_bkg[2:])

    # Normalizing data
    yFit_MB_norm.append(y_MB/max(yFit_SS[i]))
    yFit_bkg_norm.append(y_bkg/max(yFit_SS[i]))
    yFit_MB_bkg_norm.append((y_MB+y_bkg)/max(yFit_SS[i]))

# Plotting
plt.figure(figsize=(16,8))
for i in range(Nfile):
    plt.subplot(2, math.ceil(Nfile/2), i+1)

    plt.plot(xData,yData[i],      'ko',   markersize=1,label='Data')
    plt.plot(xData,yFit_MB[i],    'b-',   markersize=1,label='MB')
    plt.plot(xData,yFit_bkg[i],   'c-',   markersize=1,label='bkg')
    plt.plot(xData,yFit_MB_bkg[i],'g-',   markersize=1,label='Total')
    plt.xlim(0,1500)
    plt.title("{}\n ({})".format(evapName,fileName[i]),fontsize=10)
plt.tight_layout()
##########-------------------------------------------------------##########



##########----------------4. Angular Distribution---------------##########
# Calculate the contribution of SS from fitted data.
yArea_MB  = [sum(yFit_MB[i]/xData) for i in range(Nfile)]
yArea_bkg = [sum(yFit_bkg[i]/xData) for i in range(Nfile)]

# Cosine distribution is normalized at outgoing angle=0deg.
xAngle = [xAngle[i]/360*2*np.pi for i in range(Nfile)] # xAngle is now in unit of radian
xCOS = [i/360*2*np.pi for i in range(0,95,5)]
yCOS_max = sum([yArea_MB[i] for i in range(Nfile) if xAngle[i]==0])/len([yArea_MB[i] for i in range(Nfile) if xAngle[i]==0])
yCOS = [np.cos(xCOS[i])*yCOS_max for i in range(len(xCOS))]

yArea_MB_norm  = [yArea_MB[i]/yCOS_max for i in range(len(yArea_MB))]
yArea_bkg_norm = [yArea_bkg[i]/yCOS_max for i in range(len(yArea_bkg))]
yCOS_norm = [yCOS[i]/yCOS_max for i in range(len(yCOS))]

# Plotting
plt.figure()
Angular = plt.subplot(111, polar=True)
Angular.plot(xAngle,yArea_MB, 'bo',markersize=6,label='MB')
Angular.plot(xAngle,yArea_bkg,'co',markersize=6,label='bkg')
Angular.plot(xCOS,  yCOS,     'k--')
Angular.set_thetalim(0,np.pi/2)
Angular.set_theta_zero_location("N")
Angular.set_theta_direction(-1)
##########------------------------------------------------------##########



##########------5. Mean Translational Energy and Speed Ratio----##########
# Calculate fitted SS’s E_mean (unit=kJ/mol) and S=(speed ratio)**2
E_mean = []
S = []
xTime = [i for i in range(1,2501)]                # unit: us
xVelocity = [L/(t*10**-6) for t in xTime]         # unit: m/s
yEnergy = [0.5*m_SS/1000*v**2 for v in xVelocity] # unit: J/mol
for i in range(Nfile):
    m = yFit_SS_param[i][0]
    T = yFit_SS_param[i][1]
    v0 = yFit_SS_param[i][3]

    # Below, SS function in the velocity space is used for yIntensity. No Jacobian is required. See Comsa’s 1985 paper
    yIntensity = [1/T*np.exp(-m/1000*(v-v0)**2/(2*8.314*T))*v**3 for v in xVelocity]

    E_temp_1 = [(xVelocity[j]-xVelocity[j+1])*(yIntensity[j]+yIntensity[j+1])/2*(yEnergy[j]+yEnergy[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_temp_2 = [(xVelocity[j]-xVelocity[j+1])*(yIntensity[j]+yIntensity[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_mean_temp = sum(E_temp_1)/sum(E_temp_2)/1000 #kJ/mol
    E_mean.append(E_mean_temp)

    S_temp=yFit_SS_param[i][0]/1000*yFit_SS_param[i][3]**2/(2*8.314*yFit_SS_param[i][1])
    S.append(S_temp)
##########------------------------------------------------------##########



##########------------------6. Data Processing------------------##########
xAngle = np.reshape(xAngle,(1,Nfile)).T
yDataRaw = np.reshape(yDataRaw,(Nfile,NxData)).T
yData = np.reshape(yData,(Nfile,NxData)).T
yData_norm = np.reshape(yData_norm,(Nfile,NxData)).T
yFit_SS = np.reshape(yFit_SS,(Nfile,NxData)).T
yFit_SS_norm = np.reshape(yFit_SS_norm,(Nfile,NxData)).T
yFit_SS_param = np.reshape(yFit_SS_param,(Nfile,np.size(yFit_SS_param[0]))).T
yFit_MB = np.reshape(yFit_MB,(Nfile,NxData)).T
yFit_MB_norm = np.reshape(yFit_MB_norm,(Nfile,NxData)).T

```

```
# Deleting parameters that are not important. If wish to debug, unfunction this line.
del Angular,dm,dT_MB,dT_MB_const_bkg,dT_SS,E_mean_temp,E_temp_1,E_temp_2,evapName,fileName,folderName,i,L,m,m_bkg,m_ion,m_MB,m_SS,Nfile,NxData,pcov_MB,pcov_MB_bkg,pcov_SS,popt_MB,popt_MB_bkg,popt_SS,S_temp,T,T_MB,t_offset,T_SS,v0,x,xData_1D,xTime,xVelocity,y,y_bkg,y_MB,yCOS_max,yData_1D,yData_norm_1D,yDataRaw,yEnergy,yFit_SS,yFit_SS_norm,yFit_SS_param,yIntensity
##########------------------------------------------------------##########
```

Listing C.2: Python code to fit evaporating experiments with two Maxwell-Boltzmann distributions.

This code outputs three figures. The first figure shows data fit by supersonic distribution and shown in Figure C.3. The purpose of doing so is to find the normalizing factors for the data. Figures for the rest of the two, one is the fitting result of the TOF distributions and the other is the angular distribution. Resulting figures and shown in Figure C.4 and C.5. The TOF distributions shown in Figure 4.3 (b) are the normalized experimental data, Maxwell-Boltzmann distribution at liquid temperature, background contribution and total fittings, which are results calculated by this code listed in variables `yData_norm`, `yFit_MB_norm`, `yFit_bkg_norm` and `yFit_MB_bkg_norm`. The blue circles and light blue triangles shown in Figure 4.5 represent the angular distribution of neon evaporation and background. The data are from variables `yArea_MB_norm` and `yArea_bkg_norm`.

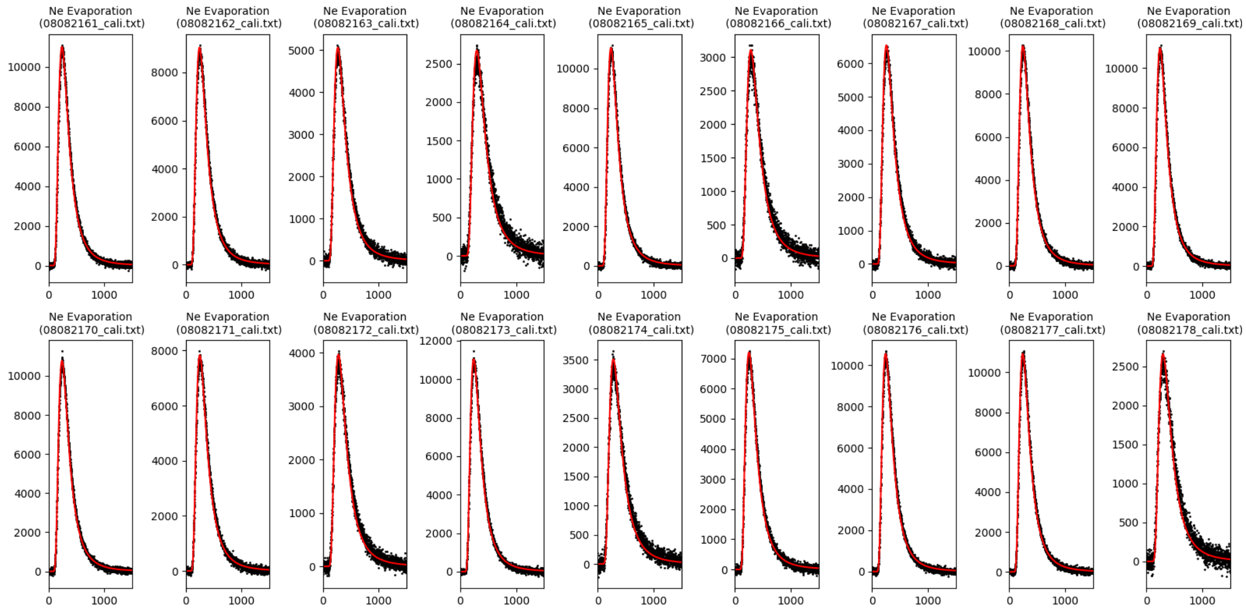


Figure C.3: TOF fitting results using a supersonic distribution in evaporation experiment (red curves).

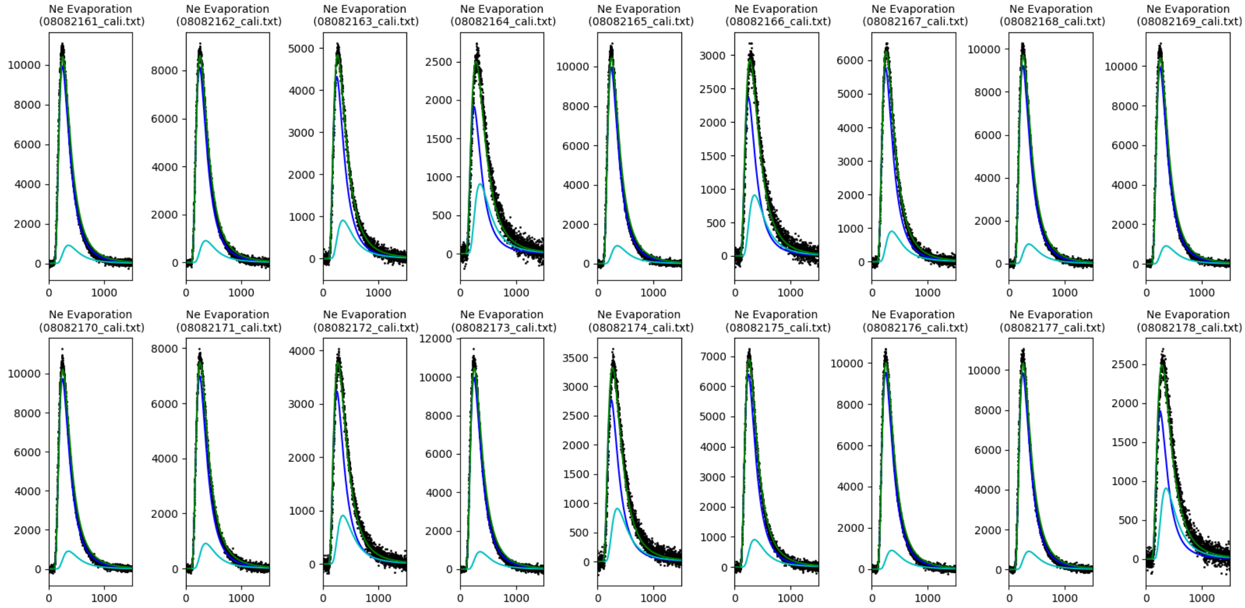


Figure C.4: TOF fitting results using two Maxwell-Boltzmann distribution in evaporation experiment. Maxwell-Boltzmann fittings at liquid temperature are shown as blue curves, while the second Maxwell-Boltzmann fittings at background temperature are shown in light blue curves. The add-up of the two fittings are shown in green curves.

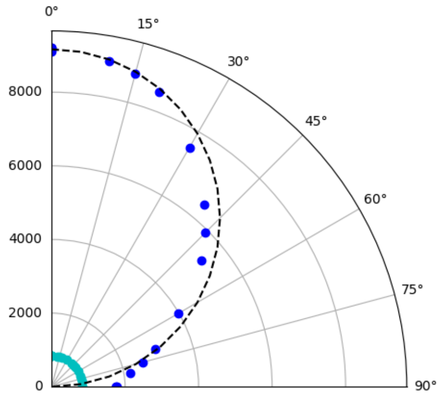


Figure C.5: Angular distribution fitting results using two Maxwell-Boltzmann distribution in evaporation experiment. Maxwell-Boltzmann fittings at liquid temperature are shown as blue circles while the second Maxwell-Boltzmann fittings at background temperature are shown in light blue circles. Cosine distribution is shown in black dash curve for reference.

C.2 Scattering Experiment

To fit the scattering data, the molecular beam profile needs to be defined first. Therefore, two Python codes are included in this section. The first one is the molecular beam profile and the second one is the fitting code with convolution of the fitted molecular beam profile.

Molecular Beam Profile

This code uses a supersonic distribution to fit the input TOF. Equation 4.2 is used and shown as

$$N_{SS,meas}(t) \propto \frac{1}{t^4} \exp\left[-\frac{m(\frac{L'}{t}-v_{SS})^2}{2RT_{SS}}\right].$$

```
##########################################################################
# <INTRO>:                                                               #
# This script is used to fit the molecular beam profile using supersonic #
# (SS) distribution . It gives the fitting parameters of temperature (T) #
# and flow velocity (v0). These numbers should be used in the scattering #
# data for convolution.                                                  #
#                                                                        #
# <INPUT>:                                                               #
# The input data should be full molecular beam profile detected without  #
# chopper wheel in at detector angle at 270deg. The beam profile on this #
# day is only with 1000 channel at 0.5us time step. The MCS delay is     #
# default to be set at the same delay as scattering data MCS delay. If   #
# MCS delay is 1250us, this number can be adjust at parameter            #
# "dMCSdelay". Th input data should be saved in a folder, which is at    #
# the same file path as the script.                                      #
#                                                                        #
# <OUTPUT>:                                                              #
# The output data are in the variables. Only the important ones are      #
# shown:                                                                 #
#                                                                        #
#    xData:          flight time (us)                                    #
#    yDataRaw:       data from the input file                            #
#    yData:          yDataRaw with beam on/off subtraction               #
#    yData_norm:     normalizeation of yData                             #
#    yFit_SS:        fitted supersonic profile of yData                  #
#    yFit_SS_norm:   normalizeation of yFit_SS                           #
#    yFit_SS_param:  parameters of yFit_SS                               #
#                                                                        #
#    beam_T:         average value of fitted beam temperature (K)        #
#    beam_v0:        average value of fitted flow velocity (m/s)         #
#                                                                        #
#    E_mean:         translational energy of the beam (kJ/mol)           #
#    beam_E:         average value of translational energy of the beam   #
#                    (kJ/mol)                                            #
#                                                                         #
# <OUTLINE>:                                                              #
#       PARAMETERS                                                        #
#    0. Initialize Enviornment                                            #
#    1. Data Loading                                                      #
#    2. Normalization and Plotting                                        #
#    3. Mean Translational Energy                                         #
#    4. Data Processing                                                   #
###########################################################################





##########----------------------PARAMETERS-----------------------##########
folderName = "Molecular Beam"
# name of the folder that includes molecular beam data
beamName = "Fast Ne"
# name of the beam
L = 0.1716+0.036
# unit: m, flight distance, default should be 0.1716+0.036 cm
dMCSdelay = 0
# unit: us, if the molecular beam profile are taken using scattering MCS delay time, put 0us. If the MCS delay time is, put (1250-scattering MCS delay time).

# Supersonic (SS) distribution fitting parameters
m = 20           # unit: amu, parent ion mass
m_ion = 20       # unit: amu, daughter ion mass
dm = 0.0001      # unit: amu
T = 0            # unit: K, fitting temperature parameter for SS
dT = 1000        # unit: K
v0 = 0           # unit: m/s, fitting flow velocity parameter for SS
dv0 = 2000       # unit: m/s
##########-------------------------------------------------------##########





##########---------------0. Initialize Enviornment---------------##########
from IPython import get_ipython
get_ipython().magic('reset -sf')

import pathlib
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import numpy as np
import math
import os
```

```
##########------------------------------------------------------##########



##########--------------------1. Data Loading-------------------##########
# Function to load data
def LoadData(folderName):
    filePath = str(pathlib.Path().resolve())+str("\\")+folderName
    fileName = os.listdir(filePath)
    yDataRaw = []
    for i in range(len(fileName)):
        file = os.path.join(filePath , fileName[i])
        yDataRaw.append(np.loadtxt(file))
    return fileName, yDataRaw

# Function do beam subtraction
def Subtract(yDataRaw):
    count = int(len(yDataRaw))
    yData = []
    for i in range(count):
        length = int(len(yDataRaw[i])/2)
        y = [yDataRaw[i][j] - yDataRaw[i][j+length] for j in range(length)]
        yData.append(y)
    return yData

# Load raw data into Python. Prepare xData and yData with time offset and beam on/off subtraction.
fileName, yDataRaw = LoadData(folderName)
yData  = yDataRaw # If the raw data include beam on and off, use "Subtract" function to do beam subtraction
t_offset = (-6 + 19.5 + 4.8*(m_ion)**0.5) - dMCSdelay # dMCSdelay is used when the MCS delay is setted wrong
xData  = np.array([i*0.5 - t_offset for i in range(len(yData[0]))])

Nfile  = len(fileName) # int, number of files
NxData = len(xData)    # list, number of x in xData
##########------------------------------------------------------##########



##########-------------2. Normalization and Plotting-------------##########
# Define supersonic (SS) distribution
def SS(t, m, T, A, v0):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6)-v0)**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

```

```
# Fit molecular beam profile with SS distribution
yFit_SS = []
yFit_SS_param = []
for i in range(Nfile):
    y = yData[i]

    popt_SS,pcov_SS = curve_fit(SS,xData,y,bounds=([m,T,0,v0],[m+dm,T+dT,100000,v0+dv0]))

    yFit_SS.append(SS(xData,*popt_SS))
    yFit_SS_param.append(popt_SS)

# Plotting
plt.figure(figsize=(8,6))
for i in range(Nfile):
    plt.subplot(1, math.ceil(Nfile/1), i+1)

    plt.plot(xData,yData[i],  'ko',markersize=1,label='Data')
    plt.plot(xData,yFit_SS[i],'r-',markersize=1,label='SS')
    plt.xlim(0,400)
    plt.xlabel('Flight Time (us)')
    plt.title("{}\n ({})".format(beamName,fileName[i]),fontsize=10)
    plt.text(380,1000,"T =  {:.2f} K    \n v0 ={:.2f} m/s".format(yFit_SS_param[i][1],yFit_SS_param[i][3]),horizontalalignment='right',fontsize=8)
plt.tight_layout()
plt.legend()

# Print fitted beam temperature (T) and flow velocity (v0)
beam_T =  sum([yFit_SS_param[i][1] for i in range(Nfile)])/Nfile
beam_v0 = sum([yFit_SS_param[i][3] for i in range(Nfile)])/Nfile

print("Average beam temperature (T)  =  %.4f K" % beam_T)
print("Average flow velocity    (v0) = %.4f m/s" % beam_v0)
##########------------------------------------------------------##########



##########-------------3. Mean Translational Energy-------------##########
# Calculate fitted SS's E_mean (unit=kJ/mol) and S=(speed ratio)**2
E_mean = []
S = []
xTime = [i for i in range(1,2501)]             # unit: us
xVelocity = [L/(t*10**-6) for t in xTime]      # unit: m/s
yEnergy = [0.5*m/1000*v**2 for v in xVelocity] # unit: J/mol
for i in range(Nfile):
    m = yFit_SS_param[i][0]
    T = yFit_SS_param[i][1]
    v0 = yFit_SS_param[i][3]

    # Below, SS function in the velocity space is used for yIntensity. No Jacobian is required. See Comsa's 1985 paper
    yIntensity = [1/T*np.exp(-m/1000*(v-v0)**2/(2*8.314*T))*v**3 for v in xVelocity]

    E_temp_1 = [(xVelocity[j]-xVelocity[j+1])*(yIntensity[j]+yIntensity[j+1])/2*(yEnergy[j]+yEnergy[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_temp_2 = [(xVelocity[j]-xVelocity[j+1])*(yIntensity[j]+yIntensity[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_mean_temp = sum(E_temp_1)/sum(E_temp_2)/1000 #kJ/mol
    E_mean.append(E_mean_temp)

    S_temp=yFit_SS_param[i][0]/1000*yFit_SS_param[i][3]**2/(2*8.314*yFit_SS_param[i][1])
    S.append(S_temp)

# Print fitted beam temperature (T) and flow velocity (v0)
beam_E = sum([E_mean[i] for i in range(Nfile)])/Nfile
print("Average beam energy (E)  =  %.2f kJ/mol" % beam_E)
##########------------------------------------------------------##########



##########------------------4. Data Processing------------------##########
# Normalize yData and yFit_SS for data plotting in origin
yData_norm = [[i/max(yFit_SS[j]) for i in yData[j]]for j in range(Nfile)]
yFit_SS_norm = [[i/max(yFit_SS[j]) for i in yFit_SS[j]]for j in range(Nfile)]

# Reshaping the data for easy copying
yDataRaw = np.reshape(yDataRaw,(Nfile,NxData)).T
yData = np.reshape(yData,(Nfile,NxData)).T
yData_norm = np.reshape(yData_norm,(Nfile,NxData)).T
yFit_SS = np.reshape(yFit_SS,(Nfile,NxData)).T
yFit_SS_norm = np.reshape(yFit_SS_norm,(Nfile,NxData)).T
yFit_SS_param = np.reshape(yFit_SS_param,(Nfile,np.size(yFit_SS_param[0]))).T

# Deleting parameters that are not important. If wish to debug, unfunction this line.
del beamName,dm,dT,dv0,dMCSdelay,E_mean_temp,E_temp_1,E_temp_2,fileName,folderName,i,L,m,m_ion,Nfile,NxData,pcov_SS,popt_SS,T,t_offset,v0,xVelocity,y,yEnergy,yIntensity
##########------------------------------------------------------##########
```

Listing C.3: Python code to fit molecular beam profile with a supersonic distribution.

The output gives a average value of fitting parameters of temperature T and flow velocity v_0. An average beam energy E is calculated using the fit parameters. The output figure shows the fitting curve compared with the experimental beam profile. It is shown in Figure C.6.

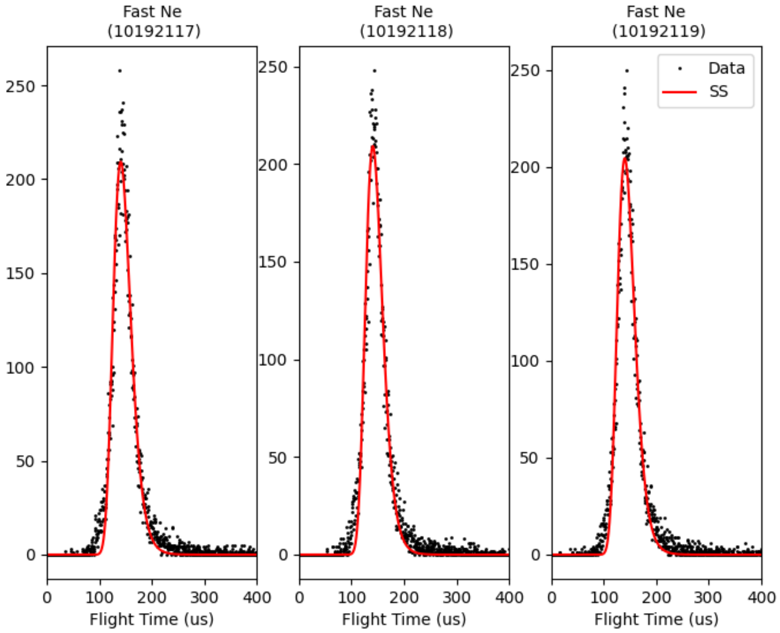


Figure C.6: Molecular beam fitting results for a fast neon beam. Average of the beam velocity and temperature from the three fitting data are shown in the code output, which are not shown here.

Maxwell-Boltzmann + Supersonic Distribution Fitting

Scattering data is analyzed by fitting a combination of a Maxwell-Boltzmann distribution with a supersonic distribution. Equation 4.4 is used and shown as

$$N_{MB,SS}(t) = \frac{c_{MB}}{t^4} \exp\left[-\frac{m(\frac{L}{t})^2}{2RT_{liq}}\right] + \frac{c_{SS}}{t^4} \exp\left[-\frac{m(\frac{L}{t} - v_{SS})^2}{2RT_{SS}}\right].$$

The following code first use the data at later arrival time to rough fit the MB distribution. After subtracting this MB, use a supersonic distribution to fit it. Finally, with both pre-fit parameters, the final fitting results are re-fit by simultaneously fitting both distributions.

```
########################################################################
# <INTRO>:                                                             #
# This script is used to fit the scattering data with a impulsive      #
# scattering (IS) and a thermal desorption (TD) distribution. IS is    #
# fitted with a supersonic (SS) distribution convolved with a molecular #
# beam profile, also fitted with SS. The TD part is fitted with a      #
# Maxwell-Boltzmann distribution convolved with a molecular beam profile.#
#                                                                      #
# <INPUT>:                                                             #
# The input datas need to be beam on/off subtracted and intensity        #
# calibrated. Time interval in each data are default as 1 us. The        #
# molecular beam profile also needs to be pre-fitted with is a           #
# supersonic distribution that the parameters are extracted with another #
# Python script. The input data should be saved in a folder, which is at #
# the same file path as the script.                                      #
#                                                                        #
# <OUTPUT>:                                                              #
# The output data are in the variables. Only the important ones are      #
# shown:                                                                 #
#                                                                        #
#    xData:           flight time (us)                                   #
#    yData:           loaded data                                        #
#    yData_norm:      normalized yData                                   #
#    yFit_MB:         fitted Maxwell-Boltzmann profile of yData          #
#    yFit_MB_norm:    normalized yFit_MB                                 #
#    yFit_MB_param:   parameters of yFit_MB                              #
#    yFit_MB_SS:      fitted SS+MB profile of yData                      #
#    yFit_MB_SS_norm: normalized yFit_MB_SS                              #
#    yFit_SS:         fitted supersonic profile of yData                 #
#    yFit_SS_norm:    normalized yFit_SS                                 #
#    yFit_SS_param:   parameters of yFit_SS                              #
#                                                                        #
#    xCOS:            radian of angular distribution (radian)            #
#    yArea_MB:        area of yFit_MB                                    #
#    yArea_MB_norm:   normalized yArea_MB                                #
#    yArea_SS:        area of yFit_SS                                    #
#    yArea_SS_norm:   normalized yArea_SS                                #
#    yCOS:            fitted Cosine distribution                         #
#    yCOS_norm:       normalized yCOS                                    #
#                                                                        #
#    xAngle:          angle of the out going beam (degree)               #
#    E_mean:          mean translational energy of yFit_SS               #
#    S:               speed ratio of yFit_SS                             #
#                                                                        #
# <OUTLINE>:                                                             #
#       PARAMETERS                                                       #
#    0. Initialize Enviornment                                           #
#    1. Data Loading                                                     #
#    2. Generate Molecular Beam Profile                                  #
#    3. Fit Distribution with Convolved SS and MB with Beam Profile      #
#    4. Angular Distribution                                             #
#    5. Mean Translational Energy and Speed Ratio                        #
#    6. Data Processing                                                  #
##########################################################################
```

```
##########----------------------PARAMETERS----------------------##########
folderName = "Fast Ne_60deg"
# name of the folder that includes molecular beam data
beamName = "Fast Ne"
# name of the beam
xAngle = [60,40,30,50,70,90,80,60]
# out going angle of each data. the sequence needs to be the same as the file sequence
L = 0.1716+0.036
# unit: m, flight distance, default should be 0.1716+0.036 cm

# Maxwell-Boltzmann (MB) and Supersonic (SS) distribution fitting parameters
m_MB = 20           # unit: amu
m_SS = 20           # unit: amu
m_ion = 20          # unit: amu
dm = 0.0001         # unit: amu

T_MB = 277          # unit: K
T_SS = 0            # unit: K
dT_MB = 0.001       # unit: K
dT_SS = 1000        # unit: K

# Fitting range for pre-fitting of Maxwell-Boltzmann distribution
MB_fit_1 = 400      # unit: us
MB_fit_2 = 800      # unit: us

beam_T = 75.2166    # use parameter fitted in "101921_Molecular Beam.py"
beam_v0 = 1392.7678 # use parameter fitted in "101921_Molecular Beam.py"
##########------------------------------------------------------##########





##########---------------0. Initialize Enviornment--------------##########
from IPython import get_ipython
get_ipython().magic('reset -sf')

import pathlib
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import numpy as np
import math
import os
##########------------------------------------------------------##########



##########--------------------1. Data Loading-------------------##########
```

```
# Function to load data
def LoadData(folderName):
    filePath = str(pathlib.Path().resolve())+str("\\")+folderName
    fileName = os.listdir(filePath)
    yDataRaw = []
    for i in range(len(fileName)):
        file = os.path.join(filePath , fileName[i])
        yDataRaw.append(np.loadtxt(file))
    return fileName, yDataRaw

# Load raw data into Python. Prepare xData and yData with time offset.
fileName, yDataRaw = LoadData(folderName)
yData = yDataRaw
t_offset = (-6 + 19.5 + 4.8*(m_ion)**0.5)
xData = np.array([i - t_offset for i in range(len(yData[0]))])

Nfile = len(fileName) # int, number of files
NxData = len(xData)   # list, number of x in xData
##########------------------------------------------------------##########



##########----------2. Generate Molecular Beam Profile----------##########
# Define supersonic (SS) distribution
def SS(t, m, T, A, v0):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6)-v0)
   **2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

xBeam = np.array([float(i) for i in range(1,1000)])
yBeam = SS(xBeam,m_SS,beam_T,1,beam_v0)
##########------------------------------------------------------##########



##########-----3. Fit Distribution with Convolved SS and MB-----##########
# Define Maxwell-Boltzmann (MB) distribution
def MB(t, m, T, A):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6))**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

# Function to convolve two distribution
def Convolution(main, window):
    temp = np.convolve(main,window)
    NofOnes = window.tolist().count(1)
    if NofOnes > 1:
        Dist = temp[int((len(window)-1)/2):len(temp)-int((len(window)-1)/2)]
    else:
        t_max_idx = np.argmax(window)
        Dist = temp[t_max_idx:len(temp)-(len(window)-t_max_idx)+1]
    Dist = Dist/max(Dist)*max(main)
    return Dist

# Supersonic (SS) distribution convolved with molecular beam profile
def SS_con(t, m_SS, T_SS, A, v0):
    temp = Convolution(SS(t, m_SS, T_SS, 1, v0), yBeam)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

# Maxwell-Boltzmann (MB) distribution convolved with molecular beam profile
def MB_con(t, m_MB, T_MB, A):
    temp = Convolution(MB(t, m_MB, T_MB, 1), yBeam)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

idx_MB_fit_1 = int(MB_fit_1 + t_offset)
idx_MB_fit_2 = int(MB_fit_2 + t_offset)

# Maxwell-Boltzmann (MB) distribution convolved with molecular beam profile. Only fit TOF in a certain range of flight time.
def MB_con_fit(t, m_MB, T_MB, A):
    temp = Convolution(MB(t, m_MB, T_MB, 1), yBeam)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    Dist = Dist[idx_MB_fit_1 : idx_MB_fit_2]
    return Dist

# Supersonic (SS) and Maxwell-Boltzmann (MB) distribution convolved with molecular beam profile
def MB_SS_con(t, m_MB, T_MB, A_MB, m_SS, T_SS, A_SS, v0):
    temp_MB = MB_con(t, m_MB, T_MB, A_MB)
    Dist_MB = temp_MB/np.max(temp_MB)
    temp_SS = SS_con(t, m_SS, T_SS, A_SS, v0)
    Dist_SS = temp_SS/np.max(temp_SS)
    Dist = A_MB*Dist_MB + A_SS*Dist_SS
    return Dist

# Fit distribution with a convolved MB and a convolved SS.
yFit_MB = []
yFit_MB_param = []
yFit_SS = []
yFit_SS_param = []
```

```
yFit_MB_SS = []
for i in range(Nfile):
    y = yData[i]

    # First fit with a convolved MB ditribution at a defined range, then fit a convolved SS distribution.
    popt_MB,pcov_MB = curve_fit(MB_con,xData[idx_MB_fit_1:idx_MB_fit_2],y[idx_MB_fit_1:idx_MB_fit_2],bounds=([m_MB,T_MB,0],[m_MB+dm,T_MB+dT_MB,100000])) #pre fit
    popt_SS,pcov_SS = curve_fit(SS_con,xData,y-MB_con(xData,*popt_MB),bounds=([m_SS,T_SS,0,0],[m_SS+dm,T_SS+dT_SS,100000,2000]))

    # Then, a MB+SS convolved distribution is fine tuned using the previous parameters.
    popt_MB_SS,pcov_MB_SS = curve_fit(MB_SS_con,xData,y,p0=[*popt_MB,*popt_SS],bounds=([m_MB,T_MB,0,m_SS,T_SS,0,0],[m_MB+dm,T_MB+dT_MB,100000,m_SS+dm,T_SS+dT_SS,100000,2000]))

    yFit_MB.append(MB_con(xData,*popt_MB_SS[0:3]))
    yFit_SS.append(SS_con(xData,*popt_MB_SS[3:]))
    yFit_MB_param.append(popt_MB_SS[0:3])
    yFit_SS_param.append(popt_MB_SS[3:])
    yFit_MB_SS.append(MB_SS_con(xData,*popt_MB_SS))

# Plotting
plt.figure(figsize=(16,8))
for i in range(Nfile):
    plt.subplot(2, math.ceil(Nfile/2), i+1)

    plt.plot(xData,yData[i],      'ko',markersize=1,label='Data')
    plt.plot(xData,yFit_MB[i],    'b-',markersize=1,label='MB')
    plt.plot(xData,yFit_SS[i],    'r-',markersize=1,label='SS')
    plt.plot(xData,yFit_MB_SS[i],'g-',markersize=1,label='Total')
    plt.xlim(0,1000)
    plt.title("{}\n ({})".format(beamName,fileName[i]),fontsize=10)
plt.tight_layout()

yData_norm = [[i/max(yFit_MB_SS[j]) for i in yData[j]]for j in range(Nfile)]
yFit_SS_norm = [[i/max(yFit_MB_SS[j]) for i in yFit_SS[j]]for j in range(Nfile)]
yFit_MB_norm = [[i/max(yFit_MB_SS[j]) for i in yFit_MB[j]]for j in range(Nfile)]
yFit_MB_SS_norm = [[i/max(yFit_MB_SS[j]) for i in yFit_MB_SS[j]]for j in range(Nfile)]
##########------------------------------------------------------##########



##########----------------4. Angular Distribution---------------##########
```

```python
# Calculate the contribution of MB and SS from fitted data.
yArea_MB = []
yArea_SS = []
for i in range(len(xAngle)):
    Area_MB=sum(yFit_MB[i]/xData)
    Area_SS=sum(yFit_SS[i]/xData)
    yArea_MB.append(Area_MB)
    yArea_SS.append(Area_SS)

# Function that fits a best Cosine distribution
def CosineDist(xCOS,n,A): # xCOS in unit of radian
    Dist = A*np.cos(xCOS)**n
    return Dist

# Calculate the best fitted Cosine distribution
xAngle = [xAngle[i]/360*2*np.pi for i in range(Nfile)]
xCOS=[i/360*2*np.pi for i in range(0,95,5)]
popt_COS, pcov_COS = curve_fit(CosineDist,xAngle,yArea_MB,bounds=([1,0],[1.000001,10000]))
yCOS=[np.cos(xCOS[i])*popt_COS[1] for i in range(len(xCOS))]

yArea_MB_norm = [yArea_MB[i]/popt_COS[1] for i in range(len(yArea_MB))]
yArea_SS_norm = [yArea_SS[i]/popt_COS[1] for i in range(len(yArea_SS))]
yCOS_norm = [yCOS[i]/popt_COS[1] for i in range(len(yCOS))]

# Plotting
plt.figure()
Angular = plt.subplot(111, polar=True)
Angular.plot(xAngle,yArea_MB,'bo')
Angular.plot(xAngle,yArea_SS,'ro')
Angular.plot(xCOS,yCOS,'k--')
Angular.set_thetalim(0,np.pi/2)
Angular.set_theta_zero_location("N")
Angular.set_theta_direction(-1)
##########-------------------------------------------------------##########



##########-----5. Mean Translational Energy and Speed Ratio-----##########
# Calculate fitted SS's E_mean (unit=kJ/mol) and S=(speed ratio)**2
E_mean = []
S = []
xVelocity = [i for i in range(0,3000)]
for i in range(Nfile):
    m = yFit_SS_param[i][0]
    T = yFit_SS_param[i][1]
    v0 = yFit_SS_param[i][3]
    yIntensity = [1/T*np.exp(-m/1000*(v-v0)**2/(2*8.314*T))*v**3 for v in xVelocity]
    yEnergy = [0.5*m/1000*v**2 for v in xVelocity]

    E_temp_1 = [(xVelocity[j+1]-xVelocity[j])*(yIntensity[j]+yIntensity[j+1])/2*(yEnergy[j]+yEnergy[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_temp_2 = [(xVelocity[j+1]-xVelocity[j])*(yIntensity[j]+yIntensity[j+1])/2 for j in range(int(len(xVelocity)-1))]
    E_mean_temp = sum(E_temp_1)/sum(E_temp_2)/1000 #kJ/mol
    E_mean.append(E_mean_temp)

    S_temp=yFit_SS_param[i][0]/1000*yFit_SS_param[i][3]**2/(2*8.314*yFit_SS_param[i][1])
    S.append(S_temp)
##########-------------------------------------------------------##########



##########------------------6. Data Processing------------------##########
xAngle = np.reshape(xAngle,(1,Nfile)).T
yDataRaw = np.reshape(yDataRaw,(Nfile,NxData)).T
yData = np.reshape(yData,(Nfile,NxData)).T
yData_norm = np.reshape(yData_norm,(Nfile,NxData)).T
yFit_MB = np.reshape(yFit_MB,(Nfile,NxData)).T
yFit_MB_norm = np.reshape(yFit_MB_norm,(Nfile,NxData)).T
yFit_SS = np.reshape(yFit_SS,(Nfile,NxData)).T
yFit_SS_norm = np.reshape(yFit_SS_norm,(Nfile,NxData)).T
yFit_MB_SS = np.reshape(yFit_MB_SS,(Nfile,NxData)).T
yFit_MB_SS_norm = np.reshape(yFit_MB_SS_norm,(Nfile,NxData)).T
yFit_MB_param = np.reshape(yFit_MB_param,(Nfile,np.size(yFit_MB_param[0]))).T
yFit_SS_param = np.reshape(yFit_SS_param,(Nfile,np.size(yFit_SS_param[0]))).T

# Deleting parameters that are not important. If wish to debug, unfunction this line.
del Angular,Area_MB,Area_SS,beam_T,beam_v0,beamName,dm,dT_MB,dT_SS,E_mean_temp,E_temp_1,E_temp_2,fileName,folderName,i,idx_MB_fit_1,idx_MB_fit_2,L,m,m_ion,m_MB,m_SS,MB_fit_1,MB_fit_2,Nfile,NxData,pcov_COS,pcov_MB,pcov_MB_SS,pcov_SS,popt_COS,popt_MB,popt_MB_SS,popt_SS,S_temp,T,T_MB,t_offset,T_SS,v0,xVelocity,y,yEnergy,yIntensity,yDataRaw
##########-------------------------------------------------------##########
```

Listing C.4: Python code to fit scattering experiments with a Maxwell-Boltzmann plus a supersonic distribution.

The output of this scattering code gives a TOF fitting figures and an angular distribution. In Figure C.7 and Figure C.8, they show fitting results taking the example of fast neon scattering of dodecane flat jet with incident angle at 60°.

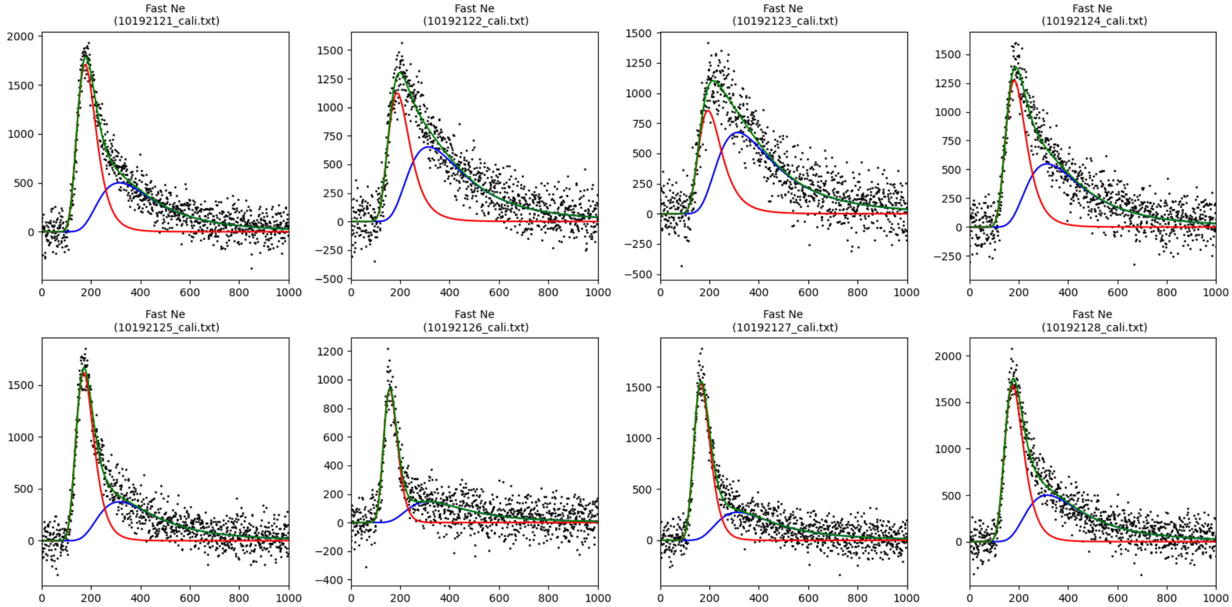


Figure C.7: TOF fitting results using a combination of a Maxwell-Boltzmann and a supersonic distribution in scattering experiment. Maxwell-Boltzmann fittings at liquid temperature are shown as blue curves while the supersonic fittings are shown in red curves. The add-up of the two fittings are shown in green curves.

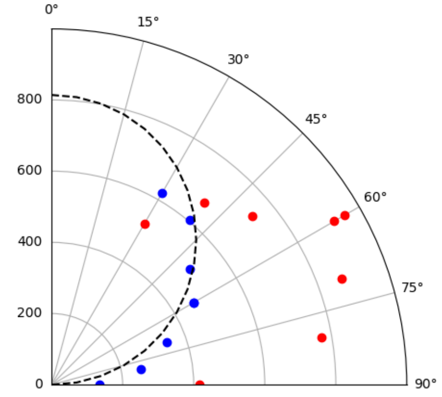


Figure C.8: Angular distribution fitting results using a combination of a Maxwell-Boltzmann and a supersonic distribution in scattering experiment. Maxwell-Boltzmann fittings at liquid temperature are shown as blue circles while the supersonic fittings at background temperature are shown in red circles. Cosine distribution is shown as black dash curve for reference.

C.3 Kinematic Model

This code uses soft-sphere model to fit the input data sets of E with E_i calculated using previous codes. Equation 4.5 is used and shown as

$$\frac{\Delta E}{E_i} \approx \frac{2\mu}{(1+\mu)^2}\left[1+\mu(\sin\chi)^2-\cos\chi\sqrt{1+\mu^2(\sin\chi)^2-\frac{E_{int}}{E_i}(\mu+1)}+\frac{E_{int}}{E_i}+\left(\frac{\mu+1}{2\mu}\right)\right]\left[1+\frac{V-2RT_{liq}}{E_i}\right].$$

```
##########################################################################
# <INTRO>:                                                               #
# This script is used to fit energy transfer parameters of soft and hard #
# sphere model. The fitted parameters are mass ratio (mu) and total      #
# internal excitation (Eint).                                            #
#                                                                        #
# <INPUT>:                                                               #
# No files are required for this code. Only that the parameters need to  #
# be filled up before calculating it, which the numbers can be pre-fit   #
# and found in "Molecular Beam.py" and "xxx_xxdeg.py".                   #
#                                                                        #
# <OUTPUT>:                                                              #
# The output data are in the variables. Only the important ones are      #
# shown:                                                                 #
#                                                                        #
#    xChi:                       deflection angle (degree)               #
#    yE_transfer:                energy transfer (ratio)                 #
#                                                                        #
#    xChi_Fit:                   deflection angle for yE_transfer_Fit    #
#                                (degree)                                #
#    yE_transfer_Fit_hard:       fitted energy transfer with hard sphere #
#                                model                                   #
#    yE_transfer_Fit_hard_param: parameters of yE_transfer_Fit_hard      #
#    yE_transfer_Fit_soft:       fitted energy transfer with soft sphere #
#                                model                                   #
#    yE_transfer_Fit_soft_param: parameters of yE_transfer_Fit_soft      #
#                                                                        #
# <OUTLINE>:                                                             #
#       Parameters                                                       #
#    0. Initialize Enviornment                                           #
#    1. Data Converting                                                  #
#    2. Energy Transfer Fitting and Plotting                             #
#    3. Data Processing                                                  #
##########################################################################



```

```


##########----------------------PARAMETERS---------------------##########
Theta_i = [   45,   60,   80] # degree, incident angle
T_liq   = [  276,  277,  278] # K,      liquid temperature
E_i     = [21.56,21.56,21.56] # kJ/mol, incident beam energy

m_g = 20 # amu, mass of incident gas

xAngle = [[45,60,80,90,70,50,45],          # degree, outgoing angle
          [60,40,30,50,70,90,80,60],
          [80,60,40,20,10,30,50,70,90,80]] # exclude 90deg

yE_SS = [[12.40746,13.36601,15.25672,16.47513,14.36215,12.83520,12.38857],
         [14.84563,13.09670,12.17047,13.92005,15.95691,18.50458,16.87659,1
         4.94734],
         [19.34127,16.69514,14.77236,12.37298,12.37839,13.63066,15.63957,1
         8.24080,19.96007,19.51147]]
          # kJ/mol, mean translational energy of suparsonic part (E_mean)
##########-----------------------------------------------------##########





##########---------------0. Initialize Environment-------------##########
from IPython import get_ipython
get_ipython().magic('reset -sf')

import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import numpy as np
##########-----------------------------------------------------##########



##########------------------1. Data Converting-----------------##########
# Also fit a "total" fit for all the Theta_i
Nfile=len(Theta_i)
Name = ['{}deg'.format(Theta_i[i]) for i in range(Nfile)]
Name.append('Total')

T_liq.append(sum(T_liq)/len(T_liq))
E_i.append(np.average(E_i))

# Convert outgoig angle(xAngle) to deflection angle(xChi)
xChi     = [180-Theta_i[i]-np.array(xAngle[i]) for i in range(Nfile)]
xChi.append(np.array(np.concatenate(xChi)))

```

```
# Convert translational energy of supersonic part(yE_SS) to energy transfer(yE_transfer)
yE_transfer = [(E_i[i]-np.array(yE_SS[i]))/E_i[i] for i in range(Nfile)]
yE_transfer.append(np.array(np.concatenate(yE_transfer)))
##########-------------------------------------------------------##########



##########--------2. Energy Transfer Fitting and Plotting-------##########
# Function to calculate energy transfer
# Ei’s unit: kJ/mol; mu’s unit: non(ratio); Eint’s unit: kJ/mol; T_liq’s unit: K
def Energy_Transfer(X, Ei, mu, Eint, T_liq): #from Minton’s 2012 paper equation (7)
    X = X/360*2*np.pi
    f = (2*mu/(mu+1)**2)*(1+mu*np.sin(X)**2+Eint/Ei*((mu+1)/(2*mu))-np.cos(X)*(1-mu**2*np.sin(X)**2-Eint/Ei*(mu+1))**0.5)
    E = (1-2*8.314*T_liq/1000/Ei)
    result = f*E
    return result

# Fitting and plotting
Colors=[’k’,’r’,’b’,’g’]
plt.figure(figsize=(8,6))

xChi_Fit=[]
yE_transfer_Fit_soft=[]
yE_transfer_Fit_hard=[]
yE_transfer_Fit_soft_param=[]
yE_transfer_Fit_hard_param=[]
for i in range(Nfile+1):
    x = xChi[i]
    y = yE_transfer[i]

    popt_soft,pcov_soft = curve_fit(Energy_Transfer,x,y,p0=[E_i[i], 0.1, 5, T_liq[i]],bounds=([E_i[i], 0.0001, 0.0001, T_liq[i]],[E_i[i]+0.0001, 10, 30,      T_liq[i]+0.0001]))
    popt_hard,pcov_hard = curve_fit(Energy_Transfer,x,y,p0=[E_i[i], 0.1, 0, T_liq[i]],bounds=([E_i[i], 0.0001, 0      , T_liq[i]],[E_i[i]+0.0001, 10,  0.0001, T_liq[i]+0.0001]))
    yE_transfer_Fit_soft_param.append(popt_soft)
    yE_transfer_Fit_hard_param.append(popt_hard)

    xChi_Fit_temp = np.arange(min(x),max(x)+1,5)
    xChi_Fit.append(xChi_Fit_temp)
    yE_transfer_Fit_soft_temp = Energy_Transfer(xChi_Fit_temp,*popt_soft)
    yE_transfer_Fit_hard_temp = Energy_Transfer(xChi_Fit_temp,*popt_hard)
    yE_transfer_Fit_soft.append(yE_transfer_Fit_soft_temp)
    yE_transfer_Fit_hard.append(yE_transfer_Fit_hard_temp)

```

```python
    plt.plot(x,y,'o',color=Colors[i],label=Name[i])
    plt.plot(xChi_Fit_temp,yE_transfer_Fit_soft_temp,'-',color=Colors[i],label='Fitting for {}_soft'.format(Name[i]))
    plt.plot(xChi_Fit_temp,yE_transfer_Fit_hard_temp,'--',color=Colors[i],label='Fitting for {}_hard'.format(Name[i]))
plt.legend()

# Print out soft and hard sphere's fitting parameters for total fitting
print('Soft Sphere Fitting:')
print('    Effective surface mass is %.2f amu' % float(m_g/popt_soft[1]))
print('    Eint/Ei is %.2f' % float(popt_soft[2]/popt_soft[0]))
print('    Eint is %.2f kJ/mol' % float(popt_soft[2]))
print('Hard Sphere Fitting:')
print('    Effective surface mass is %.2f amu' % float(m_g/popt_hard[1]))
print('    Eint/Ei is %.2f, (should be close to zero)' % float(popt_hard[2]/popt_hard[0]))
print('    Eint is %.2f kJ/mol, (should be close to zero)' % float(popt_hard[2]))
##########------------------------------------------------------------##########



##########---------------------3. Data Processing---------------------##########
yE_transfer_Fit_soft_param = np.reshape(yE_transfer_Fit_soft_param,(int(Nfile+1),len(yE_transfer_Fit_soft_param[0]))).T
yE_transfer_Fit_hard_param = np.reshape(yE_transfer_Fit_hard_param,(int(Nfile+1),len(yE_transfer_Fit_hard_param[0]))).T

# Deleting parameters that are not important. If wish to debug, unfunction this line.
del Colors,E_i,i,m_g,Name,Nfile,pcov_hard,pcov_soft,popt_hard,popt_soft,T_liq,Theta_i,x,xAngle,xChi_Fit_temp,y,yE_SS,yE_transfer_Fit_hard_temp,yE_transfer_Fit_soft_temp
##########------------------------------------------------------------##########
```

Listing C.5: Python code to fit energy transfer parameters for scattering experiments using soft- and hard-sphere kinematic model.

This code outputs the energy transfer plot. Figure 4.10 uses data points fit by this code which are listed in variables `yE_transfer_Fit_soft` and `yE_transfer_Fit_hard` for soft- and hard- sphere model. Only the fitting results of total data points are used. Fitting results of E_{int} and μ can be find in `yE_transfer_Fit_soft_param` and `yE_transfer_Fit_hard_param`, while μ, as expected, can then be transfer to m_{surf}.

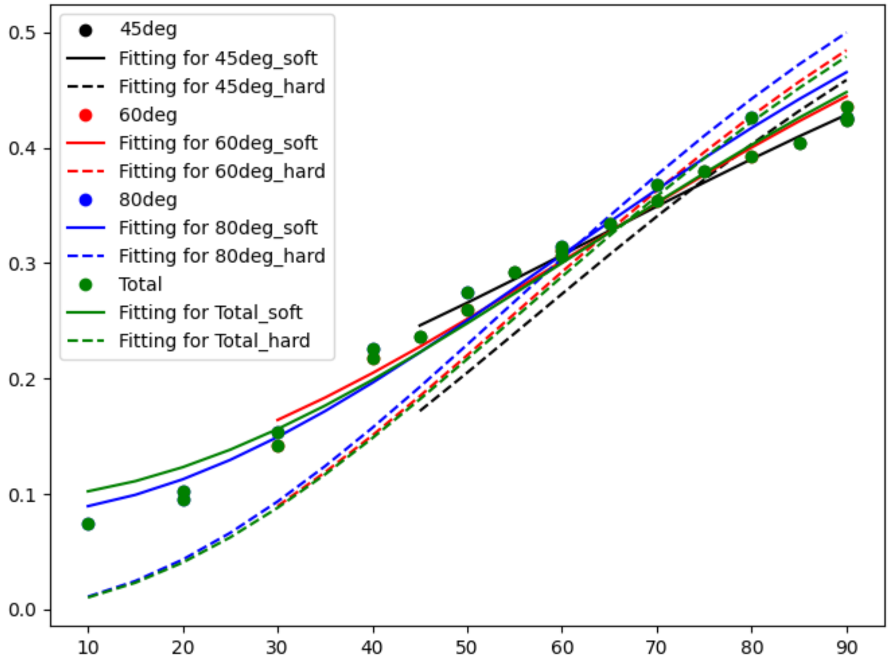


Figure C.9: Average fractional energy loss as a function of the deflection angle for impulsively scattered fast neon atoms from a dodecane flat jet. Incident angles at 45°(black), 60°(red) and 80°(blue) are shown, so as the total data point including all incident angles (green). Kinematic models of soft- (solid curves) and hard-sphere (dash curves) model are used to fit.

C.4 Concentration Plot

The gas concentration in a liquid with an impinging supersonic pulse is calculated with pre-setting parameters. The matrix to calculate beam-on and beam-off conditions shown in Section 5.1, where equations for beam-on is shown as Equation 5.8 and reprinted here as

$$\begin{bmatrix} (2a_1+1)-\frac{a_1 D}{D+k\Delta x} & -a_1 & 0 & \dots & 0 & 0 & 0 \\ -a_1 & 2a_1+1 & -a_1 & \dots & 0 & 0 & 0 \\ 0 & -a_1 & 2a_1+1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -a_1 & 2a_1+1 & -a_1 \\ 0 & 0 & 0 & \dots & 0 & -a_1 & 2a_1+1 \end{bmatrix} \begin{bmatrix} C_1^{t+1} \\ C_2^{t+1} \\ C_3^{t+1} \\ \vdots \\ C_j^{t+1} \end{bmatrix}$$

$$= \begin{bmatrix} C_1^t + \frac{kI_{beam}\Delta x a_1}{D+k\Delta x} \\ C_2^t \\ C_3^t \\ \vdots \\ C_j^t \end{bmatrix}$$

where I_{beam} is set to be the supersonic distribution using Equation 2.14 with $NFL = L$. As for beam-off condition, Equation 5.10 is reprinted here as

$$\begin{bmatrix} (2a_1+1)-\frac{a_1 D}{D+k\Delta x} & -a_1 & 0 & \dots & 0 & 0 & 0 \\ -a_1 & 2a_1+1 & -a_1 & \dots & 0 & 0 & 0 \\ 0 & -a_1 & 2a_1+1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -a_1 & 2a_1+1 & -a_1 \\ 0 & 0 & 0 & \dots & 0 & -a_1 & 2a_1+1 \end{bmatrix} \begin{bmatrix} C_1^{t+1} \\ C_2^{t+1} \\ C_3^{t+1} \\ \vdots \\ C_j^{t+1} \end{bmatrix} = \begin{bmatrix} C_1^t \\ C_2^t \\ C_3^t \\ \vdots \\ C_j^t \end{bmatrix}.$$

```python
from IPython import get_ipython
get_ipython().magic('reset -sf')

import numpy as np
import matplotlib.pyplot as plt
from scipy import special

### Assuming a supersonic pulse, this script simulate the concentration changing on the liquid surface.
### No chopper in the way. Arrival time is a sum of velocity and desorption time.

def Pdes(t,x,D,k): # from 2000 Ringeisen's thesis (3.2.14) or 2004 Muenter's thesis (3.1.2), (3.4.2)
```

```
    temp = special.erfc(x/(4*D*(t*10**-6))**0.5)-np.exp(k*x/D)*np.exp(k**2*(t*10**-6)/D)*special.erfc(k*((t*10**-6)/D)**0.5+x/(4*D*(t*10**-6))**0.5)
    return temp

L1=0.1716       # unit: m;  distance between chopper and ionizer
L2=0.036            # unit: m;  distance between interaction region and chopper
L=L1+L2

def MB(t, m, T, A, v0):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6)-v0)**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

def Convolution(main, window):
    temp=np.convolve(main,window)
    NofOnes=window.tolist().count(1)
    if NofOnes > 1:
        Dist=temp[int((len(window)-1)/2):len(temp)-int((len(window)-1)/2)]
    else:
        t_max_idx=np.argmax(window)
        Dist=temp[t_max_idx:len(temp)-(len(window)-t_max_idx)+1]
    Dist=Dist/max(Dist)*max(main)
    return Dist

##########-----------------------PARAMETERS---------------------##########
D=1*10**(-12)                         # unit: m**2/s
tau=1*10**(-6)                        # unit: s
k=(D/tau)**0.5

t_pul=500                             # unit: us
t_step=200
dt=t_pul/t_step                       # unit: us

x_end=10*(4*D*(t_pul*10**-6))**0.5    # unit: m
x_step=200
dx=x_end/x_step                       # unit: m

a1=D*(dt*10**-6)/(dx**2)

m=20 # unit: amu
T=30 # unit: K
v0=1460 # unit: m/s

# MB parameter for Gaussian fitting
mean_intensity=t_pul/2                  # unit: us, set at the center of t_pulse
```

```
sigma_intensity=15
mean_velocity=t_pul/2                    # unit: us, set at the center of t_pulse
sigma_velocity=8

t=np.array([float(i*dt) for i in range(0,t_step)])           # unit: us
t[0]=10**(-20)

xlayer = [5,10,50,100]     # what x do you want to show?
##########------------------------------------------------------##########



##########--------------------1. Calculation--------------------##########
### Generate molecular beam profile
MB_intensity=np.exp(-(t-mean_intensity)**2/(2*sigma_intensity**2))
MB_velocity=np.exp(-(t-mean_velocity)**2/(2*sigma_velocity**2))
MB_raw=Convolution(MB_intensity,MB_velocity)  #MB profile

### Generate operating matrix.
Matrix=np.zeros((x_step,x_step))
for i in range(x_step):
    for j in range(x_step):
        if i == j:
            Matrix[i,j]=2*a1+1
        elif abs(i-j) == 1:
            Matrix[i,j]=-a1
Matrix[0,0]=Matrix[0,0]-a1*D/(D+k*dx) #no unit

### Calculate concentration distribution by the solving matrix.
Conc=np.zeros((x_step,t_step))
for i in range(1,t_step):
    Conc_curr=np.copy(Conc[:,i-1])
    Conc_curr[0]=Conc_curr[0]+a1*k*dx/(D+k*dx)*MB_raw[i]
    Conc_next=np.linalg.solve(Matrix,Conc_curr)
    Conc[:,i]=Conc_next

Conc0=np.array([MB_raw[i]*k*dx/(D+k*dx)+D/(D+k*dx)*Conc[0,i] for i in range(t_step)])
##########------------------------------------------------------##########



##########----------------------2. Plotting---------------------##########
plt.figure()
plt.plot(t,MB_raw,label="Molecular Beam")
plt.plot(t,Conc0,label="x={}m".format((0)*dx))
#plt.plot(t1,Pdes(t1,0,D,k),’--’,label="x={}um".format(0*dx*10**6))  #The curve for numerically calculated Conc should be the same as analytical solution Pdes
```

```
for i in range(len(xlayer)):
    xlayer_temp=xlayer[i]
    plt.plot(t,Conc[xlayer_temp-1,:],'-',label="x={:.2e}um".format(xlayer_temp*dx*10**6))
    #plt.plot(t1,Pdes(t1,xlayer_temp*dx,D,k),'--',label="x={:.2e}um".format(xlayer_temp*dx*10**6))
plt.xlabel('Time (us)')
plt.ylabel('Concentration (arb)')
plt.gca().set_title("Supersonic Pulse (D={:.0e} m**2/s, tau={:.0e} s)".format(D,tau))
plt.legend(framealpha=1,frameon=True)
##########-------------------------------------------------------##########
```

Listing C.6: Python code to calculate concentration for a supersonic beam impinging onto a liquid surface.

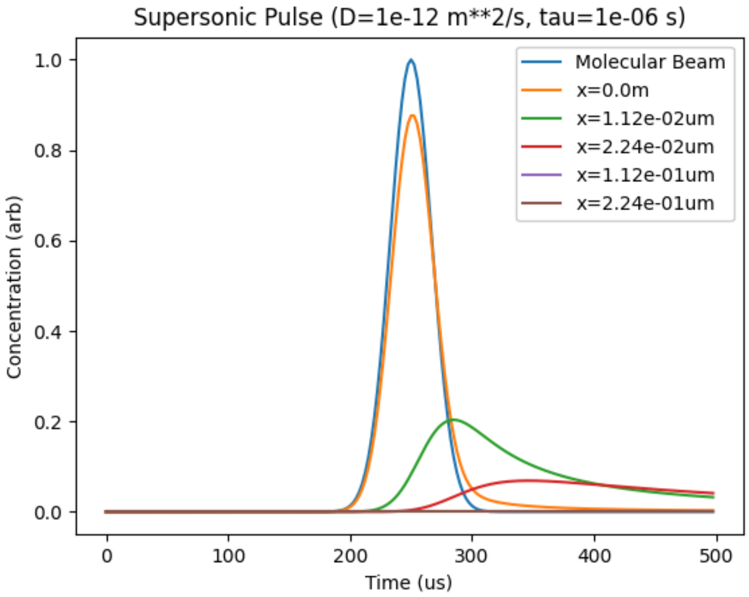


Figure C.10: Concentration plot for a supersonic beam impinging the liquid then desorb. Concentration of several layers are shown.

C.5 Chopped Scattering TOF Simulation

```
from IPython import get_ipython
get_ipython().magic('reset -sf')

from IPython import get_ipython
get_ipython().magic('reset -sf')

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import math as math
import os

### Assuming a supersonic pulse, this script simulate the concentration changing on the liquid surface.

L1=0.1716             # unit: m;  distance between chopper and ionizer
L2=0.036                        # unit: m;  distance between interaction region and chopper
L=L1+L2

def LoadData(filePath, parameter):
    fileName = os.listdir(filePath)
    yData = []
    for i in range(len(fileName)):
        file = os.path.join(filePath , fileName[i])
        yData.append(np.loadtxt(file))
    return fileName, yData

def MB(t, m, T, A, v0):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6)-v0)**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

def MW(t, m, T, A, L):
    temp = (1/(t*10**-6)**4)*np.exp(-m/1000/(2*8.314*T)*(L/(t*10**-6))**2)
    Dist = temp/np.max(temp)
    Dist = A*Dist
    return Dist

def Width(opening, speed, diameter):
  time=opening/(math.pi*diameter*10)*(1/speed)*10**6
  if (round(time) % 2)==1:        # length must be odd number
    width=np.ones(round(time)+2)
  else:
    if time > round(time):
      width=np.ones(round(time)+3)
    else:
      width=np.ones(round(time)+1)
  width[0]=0
  width[-1]=0
  return width

def Convolution(main, window):
    temp=np.convolve(main,window)
    NofOnes=window.tolist().count(1)
    if NofOnes > 1:
        Dist=temp[int((len(window)-1)/2):len(temp)-int((len(window)-1)/2)]
    else:
        t_max_idx=np.argmax(window)
        Dist=temp[t_max_idx:len(temp)-(len(window)-t_max_idx)+1]
    Dist=Dist/max(Dist)*max(main)
    return Dist

##########-----------------------PARAMETERS---------------------##########
D=1*10**(-12)                         # unit: m**2/s
tau=1*10**-5                          # unit: s

t_pul=600                             # unit: us, only use interger
t_step=600                            # unit: us, use t_pul=t_step so that the step is 1us, same as MW
dt=t_pul/t_step                       # unit: us

x_end=10*(4*D*(t_pul*10**-6))**0.5    # unit: m
x_step=200
dx=x_end/x_step                       # unit: m

# MB and MW parameter
m=20                                  # unit: amu
m_ion=20                              # unit: amu
T_MB=27                               # unit: K
T_MW=300
v0=1477                               # unit: m/s

# MB parameter for Gaussian fitting
mean_intensity=t_pul/2                  # unit: us, set at the center of t_pulse
sigma_intensity=40
mean_velocity=t_pul/2                   # unit: us, set at the center of t_pulse
sigma_velocity=20
mean_total=t_pul/2
sigma_total=40

DetectorAperture = 3            # unit: mm; default:3mm(big hole)
ChopperSlit = 200             # unit: mm; default:1.65mm
WheelSpeed = 200              # unit: Hz; default:200Hz
```

```
WheelDiameter = 17              # unit: cm; default:17cm

t_delay = 40                    # unit: us; pulse valve delay, + sige means later delay
t_MCS = 1250-1225.626

filePath = r"C:\Users\chinl\Box\Chin\Research\Latimer D10\Data\Y2021\092021_chop_AV_fast Ne scat from dodecane_60deg_flat\Test_tres" #input data needs to be intensity calibrated
xDelay = [t_delay]
##########--------------------------------------------------------##########

### Load yData and prepare xData
fileName, yDataRaw = LoadData(filePath, xDelay)
yData = yDataRaw

t_offset=(-6 + 19.5 + 4.8*(m_ion)**0.5)
NxData = len(yData[0])
t_step_MW=NxData
xData = np.array([i - t_offset for i in range(NxData)])

Nfile = len(xDelay)     # int

t_MB=np.array([float(i*dt) for i in range(0,t_step)])  # unit: us; time zero at reaction center
t_MB[0]=10**-20

### Generate molecular beam profile
#MB_intensity=np.exp(-(t_MB-mean_intensity)**2/(2*sigma_intensity**2))
#MB_velocity=np.exp(-(t_MB-mean_velocity)**2/(2*sigma_velocity**2))
#MB_raw=Convolution(MB_intensity,MB_velocity)  #MB profile
MB_raw=np.exp(-(t_MB-mean_total)**2/(2*sigma_total**2))  #MB profile

def MW_Chop(t, D, tau, A, t_delay):
    k=(D/tau)**0.5
    a1=D*(dt*10**-6)/(dx**2)

    t_MW=np.array([float(i*dt+t[0]) for i in range(0,t_step_MW)])       # unit: us; time zero at reaction center

    ### Generate operating matrix.
    Matrix=np.zeros((x_step,x_step))
    for i in range(x_step):
        for j in range(x_step):
            if i == j:
                Matrix[i,j]=2*a1+1
            elif abs(i-j) == 1:
                Matrix[i,j]=-a1
    Matrix[0,0]=Matrix[0,0]-a1*D/(D+k*dx)

    ### Calculate concentration distribution by the solving matrix.
    Conc=np.zeros((x_step,t_step))
    for i in range(1,t_step):
        Conc_curr=np.copy(Conc[:,i-1])
        Conc_curr[0]=Conc_curr[0]+a1*k*dx/(D+k*dx)*MB_raw[i]
        Conc_next=np.linalg.solve(Matrix,Conc_curr)
        Conc[:,i]=Conc_next

    Conc0=np.array([MB_raw[i]*k*dx/(D+k*dx)+D/(D+k*dx)*Conc[0,i] for i in range(t_step)])



    ### Setting up sliced Maxwell-Boltzman Distribution
    ### Calculate opening time
    ChopperWidth=Width(ChopperSlit,WheelSpeed,WheelDiameter)
    ApertureWidth=Width(DetectorAperture,WheelSpeed,WheelDiameter)

    ### Generate Maxwell-Boltzmann Distribution
    MW_raw=MW(t_MW,m,T_MW,1,L)

    MW_Conc0_chop=np.zeros([len(MW_raw)+len(MB_raw)])
    t_MB_shift=np.array([t_MB[i]-t_MB[np.argmax(MB_raw)] for i in range(len(t_MB))])
    for i in range(len(t_MB_shift)):
        for j in range(len(t_MW)):
            t_temp=L2/(L/t_MW[j])+t_MB_shift[i]
            t_chop_1=t_MCS+t_delay-(len(ChopperWidth)-2)/2-t_MB_shift[i] # unit: us
            t_chop_2=t_MCS+t_delay+(len(ChopperWidth)-2)/2-t_MB_shift[i] # unit: us
            if t_temp>t_chop_1 and t_temp<t_chop_2:
                MW_Conc0_chop[i+j]=MW_Conc0_chop[i+j]+MW_raw[j]*Conc0[i]
    MW_Conc0_chop=MW_Conc0_chop/max(MW_Conc0_chop)

    MW_Conc0_chop_con=Convolution(MW_Conc0_chop,ApertureWidth)
    MW_Conc0_chop_con=MW_Conc0_chop_con[np.argmax(MB_raw):len(MW_raw)+np.argmax(MB_raw)]
    MW_Conc0_chop_con=A*(MW_Conc0_chop_con/max(MW_Conc0_chop_con))
    return MW_Conc0_chop_con

MW_raw=MW(xData,m,T_MW,1,L)

plt.figure()
plt.plot(xData,MW_raw,label="MW")
MW_chop=[]
MW_chop_parameters=[]
for i in range(Nfile):
    x=xData
    y=yData[i]
    t_delay=xDelay[i]
    popt_MB,pcov_MB = curve_fit(MB,x,y,bounds=([m,0,0,0],[m+0.0001,500,4000,2000])) #pre fit
    popt,pcov = curve_fit(MW_Chop,x,y,p0=[10**-12,10**-4,popt_MB[2],t_delay],bounds=([10**-12,10**-8,0,t_delay],[1.1*10**-12,1,4000,t_delay+0.00001])) #pre fit
    MW_chop_temp=MW_Chop(xData,*popt)
    plt.plot(x,y,label="Data_{}us".format(t_delay))
    plt.plot(x,MW_chop_temp,label="Simulation at {}us, tau={:.0e}".format(t_delay,popt[1]))

    MW_chop.append(MW_chop_temp)
    MW_chop_parameters.append(popt)



#,p0=[10**-12,10**-6,popt_MB[2],t_delay]
plt.xlim(0,1000)
plt.xlabel('Time (us)')
plt.ylabel('Concentration')
plt.legend(framealpha=1,frameon=True)
```

Listing C.7: Python code to simulate scattering TOF with chopped scattered signal.

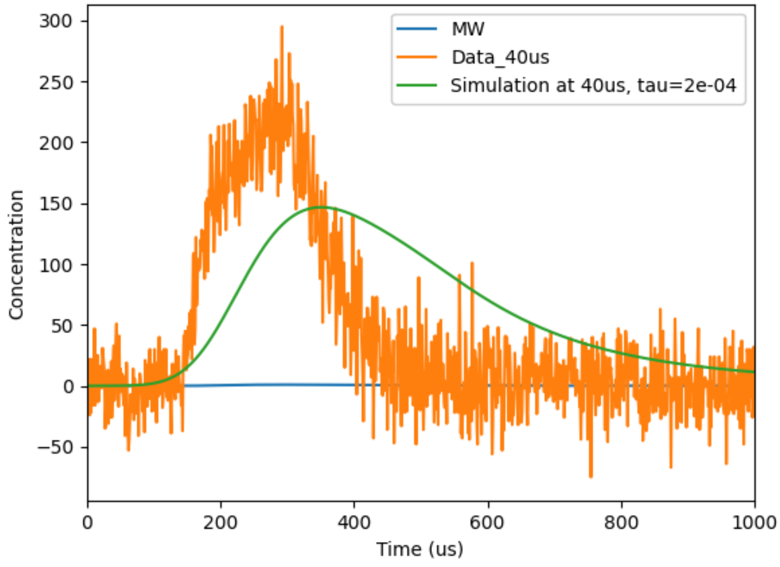


Figure C.11: Simulated chopped scattering signal. Parameters for the simulated profile are set at $t_{delay} = 40\,\mu s$, which is same as the experimental setup.

Appendix D

Publications Resulting from Graduate Work

1. W. Yang*, C. Lee*, S. Saric, M. N. Pohl, D. M. Neumark "Collisions of protic and aprotic gases with dodecane liquid surface" (in preparation, *equal authorship).

2. C. Lee, M. N. Pohl, I. A. Ramphal, W. Yang, B. Winter, B. Abel, D. M. Neumark "Evaporation and molecular beam scattering from a flat liquid jet" *J. Phys. Chem. A*, **126**, *3373* (2022).

3. M. N. Pohl, S. Malerz, F. Trinter, C. Lee, C. Kolbeck, I. Wilkinson, S. Thürmer, D. M. Neumark, L. Nahon, I. Powis, G. Meijer, B. Winter, U. Hergenhahn "Photoelectron circular dichroism in angle-resolved photoemission from liquid fenchone" *Phys. Chem. Chem. Phys.*, **24**, *8081* (2022).

4. P. E. Mason, H. C. Schewe, T. Buttersack, V. Kostal, M. Vitek, R. S. McMullen, H. Ali, F. Trinter, C. Lee, D. M. Neumark, S. Thürmer, R. Seidel, B. Winter, S. E. Bradforth, P. Jungwirth "Spectroscopic evidence for a gold-coloured metallic water solution" *Nature*, **595**, *673* (2021).

5. S. Thürmer, S. Malerz, F. Trinter, U. Hergenhahn, C. Lee, D. M. Neumark, G. Meijer, B. Winter, I. Wilkinson "Accurate vertical ionization energy and work function determinations of liquid water and aqueous solutions" *Chem. Sci.*, **12**, *10558* (2021).

6. S. Malerz, F. Trinter, U. Hergenhahn, A. Ghrist, H. Ali, C. Nicolas, C.-M. Saak, C. Richter, S. Hartweg, L. Nahon, C. Lee, C. Goy, D. M. Neumark, G. Meijer, I. Wilkinson, B. Winter, S. Thürmer "Low-energy constraints on photoelectron spectra measured

from liquid water and aqueous solutions" *Phys. Chem. Chem. Phys.*, **23**, *8246* (2021).

7. I. A. Ramphal, C. Lee, D. M. Neumark "Photodissociation dynamics of the methylsulfinyl radical at 248 nm" *Mol. Phys.*, **117**, *3043* (2019).