

January 2012

# Experimental and Numerical Investigation of an Electrospray RF Ion Funnel

Eric Miguel Tridas

University of South Florida, [eric.m.tridas@gmail.com](mailto:eric.m.tridas@gmail.com)

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>



Part of the [American Studies Commons](#), and the [Nanoscience and Nanotechnology Commons](#)

## Scholar Commons Citation

Tridas, Eric Miguel, "Experimental and Numerical Investigation of an Electrospray RF Ion Funnel" (2012). *Graduate Theses and Dissertations*.

<http://scholarcommons.usf.edu/etd/4243>

This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact [scholarcommons@usf.edu](mailto:scholarcommons@usf.edu).

Experimental and Numerical Investigation of an  
Electrospray RF Ion Funnel

by

Eric Miguel Tridas

A thesis submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science in Mechanical Engineering  
Department of Mechanical Engineering  
College of Engineering  
University of South Florida

Co-Major Professor: Rudy Schlaf, Ph.D.  
Co-Major Professor: Rasim Guldiken, Ph.D.  
Muhammad Rahman, Ph.D.

Date of Approval:  
June 11, 2012

Keywords: ESI, CFD, rhoCentralFoam, SIMOIN, OpenFOAM

Copyright © 2012, Eric Miguel Tridas

## Dedication

For Papi, Mom, Maria and Catherine. I love you all and cannot thank you enough for your support through good times and bad. I am so fortunate to have such a wonderful family.

## Acknowledgments

I would like to tip my hat to all of the members of the Surface Science Laboratory who have made my lab experience fun and funny. I would also like to thank Dr. Rasim Guldiken for his support with all of my CFD and fluid dynamics questions and for sparking such an interest in the field as well as agreeing to be my co-advisor. Dr. Mark Anthony has helped immensely with the theory behind the ion optics concepts as well as made our weekly meetings humorous. Finally, I would like to offer a very special thanks Dr. Rudy Schlaf who has been a wonderful mentor throughout my entire college career. Without him my experience at USF would have been much, much less and I cannot begin to express the gratitude I have for the opportunities, support and encouragement he has given me over the last 7 years.

## Table of Contents

List of Tables .....	iii
List of Figures .....	iv
Abstract .....	vi
1 Introduction.....	1
2 Electrospray Ionization Background .....	2
2.1 Electrospray Ionization.....	2
3 Fluid Flow/Simulation Background .....	9
3.1 Fluid Flow Background .....	9
3.2 OpenFOAM and the Finite Volume Method .....	19
3.3 rhoCentralFoam .....	20
3.4 SIMION® and the Hard Sphere Collision Model .....	25
4 Physical Apparatus.....	27
5 Experimental Procedures .....	38
5.1 Procedure for Electrospray Measurements.....	38
5.2 Procedure for Simulation .....	40
5.2.1 Definition of Fluid Flow Problem.....	40
5.2.2 Creation of Model Geometry .....	42
5.2.3 Meshing the Model.....	44
5.2.4 Definition of Simulation Parameters .....	47
5.2.5 Running the Simulation .....	49
5.2.6 Post-Processing the Simulation .....	50
5.3 Development of LabVIEW Interface.....	51
5.3.1 Theory Behind the Interface.....	51
5.3.2 Object Definitions .....	54
5.3.3 Data Transfer Protocol .....	54
6 Results and Discussion .....	57
6.1 LabVIEW Interface.....	57
6.1.1 Front Panel.....	57
6.1.2 Block Diagram.....	61

6.2 Electrospray Current Measurements .....	78
6.2.1 Current versus Needle to Capillary Distance.....	79
6.2.2 Current versus Heater Temperature .....	84
6.2.3 Current versus Solution Flow Rate.....	88
6.2.4 Funnel Transmission versus Jet Disruptor Voltage .....	91
6.3 Fluid Flow Simulation Results .....	94
6.3.1 Flow Field Properties .....	94
6.3.2 Temporal Convergence.....	100
6.3.3 Mesh Convergence .....	103
6.4 SIMION Simulation Results .....	103
6.4.1 Standard Operating Conditions .....	103
6.4.2 No Electric Field Applied .....	106
6.4.3 Variation of Jet Disruptor Voltage.....	107
7 Conclusions .....	112
References .....	115
Appendices.....	118
Appendix A – OpenFOAM Simulation Configuration Files .....	119
Appendix B – Additional SIMION Simulation Results .....	134
Appendix C – Third Party Material Permissions.....	137

## List of Tables

Table 1: Mesh convergence of 20 mil capillary simulation (top) and 30 mil capillary simulation (bottom).....	103
--	-----

## List of Figures

Figure 1: Schematic of the electrospray process .....	3
Figure 2: Various forms of menisci when performing electrospray .....	4
Figure 3: Photograph of a droplet undergoing coulomb fission .....	6
Figure 4: Image of electrospray emitter with cone-jet and plume .....	7
Figure 5: Schematic diagram of an underexpanded jet through a converging nozzle .....	17
Figure 6: Locations of the sonic line for various nozzle geometries .....	18
Figure 7: Schematic of discretization methodology used in OpenFOAM between cell P and neighbor cell N (reprinted with permission from [23]).....	21
Figure 8: Numerical (top) and experimental (bottom) results of a underexpanded jet using the rhoCentralFoam solver (reprinted with permission from [23]) .....	25
Figure 9: Overall view of the macromolecular patterning system .....	28
Figure 10: Image of the heater block with capillary .....	29
Figure 11: Injection area for electrospray .....	30
Figure 12: Schematic (a) and side view (b) of the RF ion funnel.....	31
Figure 13: View of the funnel along the axial direction with the heater block/capillary assembly removed.....	33
Figure 14: Image taken from GMSH showing the axisymmetrical model used for the fluid flow simulations .....	44
Figure 15: Image of the meshed geometry from GMSH.....	45
Figure 16: Image of the front panel of the RF ion funnel controller GUI .....	58
Figure 17: Screen shots of the front panel of the private variable definitions for the numerical LabVIEW class (left) and Boolean LabVIEW class (right) .....	62
Figure 18: Block diagram used to check for and analyze any data incoming over the serial port from the Arduino to LabVIEW .....	64
Figure 19: Analysis of a Boolean packet .....	66
Figure 20: Analysis of a numerical packet.....	67
Figure 21: Front panel images of Boolean (left) and numeric (right) setValue methods .....	69
Figure 22: Block diagram images of the GUI front panel control sections used to update their associated objects for the Boolean (left) and numeric (right) objects.....	75
Figure 23: Block diagram image of the current time averaging and time history display function.....	77



Figure 24: Schematic indicating the measurement points on the funnel electrodes .....	79
Figure 25: Current measurements versus needle to capillary distance variation of 20 mil (top) and 30 mil (bottom) .....	81
Figure 26: Funnel transmission values versus needle to capillary distance of 20 mil capillary (top) and 30 mil capillary (bottom) .....	83
Figure 27: Funnel transmission values versus heater temperature variation of 20 mil capillary (top) and 30 mil capillary (bottom) .....	85
Figure 28: Current measurements versus heater temperature variation of 20 mil capillary (top) and 30 mil capillary (bottom) .....	87
Figure 29: Transmission values versus the solution flow rate variation of 20 mil capillary (top) and 30 mil capillary (bottom) .....	89
Figure 30: Results from Cloupeau et. al. showing how the variation of solution flow rate effects electrospray properties .....	90
Figure 31: Total funnel transmission versus jet disruptor voltage.....	92
Figure 32: Fluid flow field results.....	94
Figure 33: Closeup images of the velocity magnitude of the jet issuing from the capillary in the 20 mil (top) and 30 mil (bottom) cases. ....	97
Figure 34: Centerline Mach number values for the 20 mil (top) and 30 mil (bottom) capillary simulations.....	99
Figure 35: Temporal convergence graphs for 20 mil (top) and 30 mil (bottom) capillary simulations.....	102
Figure 36: Combined electrodynamic and CFD simulation results for 20 mil (top) and 30 mil (bottom) capillary .....	104
Figure 37: Coupled SIMION and CFD simulations of 20 mil (top) and 30 mil (bottom) capillaries with all electric fields disabled .....	106
Figure 38: Coupled SIMION and CFD simulations with an applied jet disruptor voltage of 170 V and 210 V for the 20 mil capillary (a, b) and the 30 mil capillary (c, d) .....	108
Figure 39: Experimental and simulated values of transmission when varying the jet disruptor voltage for the 20 mil (top) and 30 mil (bottom) cases .....	109
Figure 40: SIMION simulation results of 20 mil capillary with no fluid flow coupling.....	135
Figure 41: SIMION simulation results of 30 mil capillary with no fluid flow coupling.....	136

## Abstract

Using experimental techniques along with computational fluid dynamics and electrodynamic simulations the performance of the first of three focusing elements in an electrospray macromolecular patterning system was assessed. The performance of this element, the ion funnel, was analyzed by varying the parameters and electric field applied to the system including electrospray emitter to atmosphere-vacuum interface capillary distance, temperature of the desolvating heater, injection rate of solution and the voltage applied to the jet disruption element. Results indicated that processes involved in injecting larger droplets into the chamber resulted in a less effective transmission of the ions through the funnel. Droplet diameter was increased by increasing flow rate and was decreased by increasing the desolvation heater. Varying the voltage applied to the jet disrupting element indicated a peak transmission voltage, when using a 20 mil interface capillary, of 175 V and when using the 30 mil capillary of 180 V. Numerical simulations were in agreement with these values although the widths of these transmission curves were much narrower than the experimental curves.

## 1 Introduction

The focus of this study is the Electrospray Macromolecular patterning system located in the Surface Science Laboratory at the University of South Florida. This system is a deposition apparatus used to transmit macromolecules in solution from atmospheric conditions to high vacuum where they can be deposited or analyzed. The ions are removed from solution by the electrospray process which creates a spray of micron and submicron sized droplets as well as gas phase ions.

Specifically this study focuses on the first and lowest vacuum of three vacuum chambers of the system. This chamber contains the first focusing element called the radio frequency (RF) Ion Funnel. The goals of this study are:

- Develop a graphical user interface using LabVIEW to communicate between the electronics controlling the ion funnel and the computer
- Perform systematic variations on several physical parameters of the ion funnel and analyze its performance
- Simulate the airflow into the ion funnel chamber using computational fluid dynamics (CFD)
- Couple the CFD simulation previously described with an electrodynamic simulation modeling the funnel and compare the results with those obtained experimentally

## 2 Electrospray Ionization Background

### 2.1 Electrospray Ionization

Electrospray ionization (ESI) is a process used to isolate gas phase macromolecules from solution for analysis or deposition [1, 2] and is commonly used in mass spectrometry. The process was first developed by Dole in an effort to separate singular high polymers from solvent [3]. The process was further advanced by Fenn who was awarded the Nobel Prize in chemistry in 2002 for his work on electrospray mass spectroscopy (ESI-MS) [4]. The main benefits of the process compared to other ionization and deposition techniques is that it is able to produce singular ions from solution while at the same time maintaining the original functionality and physical structure.

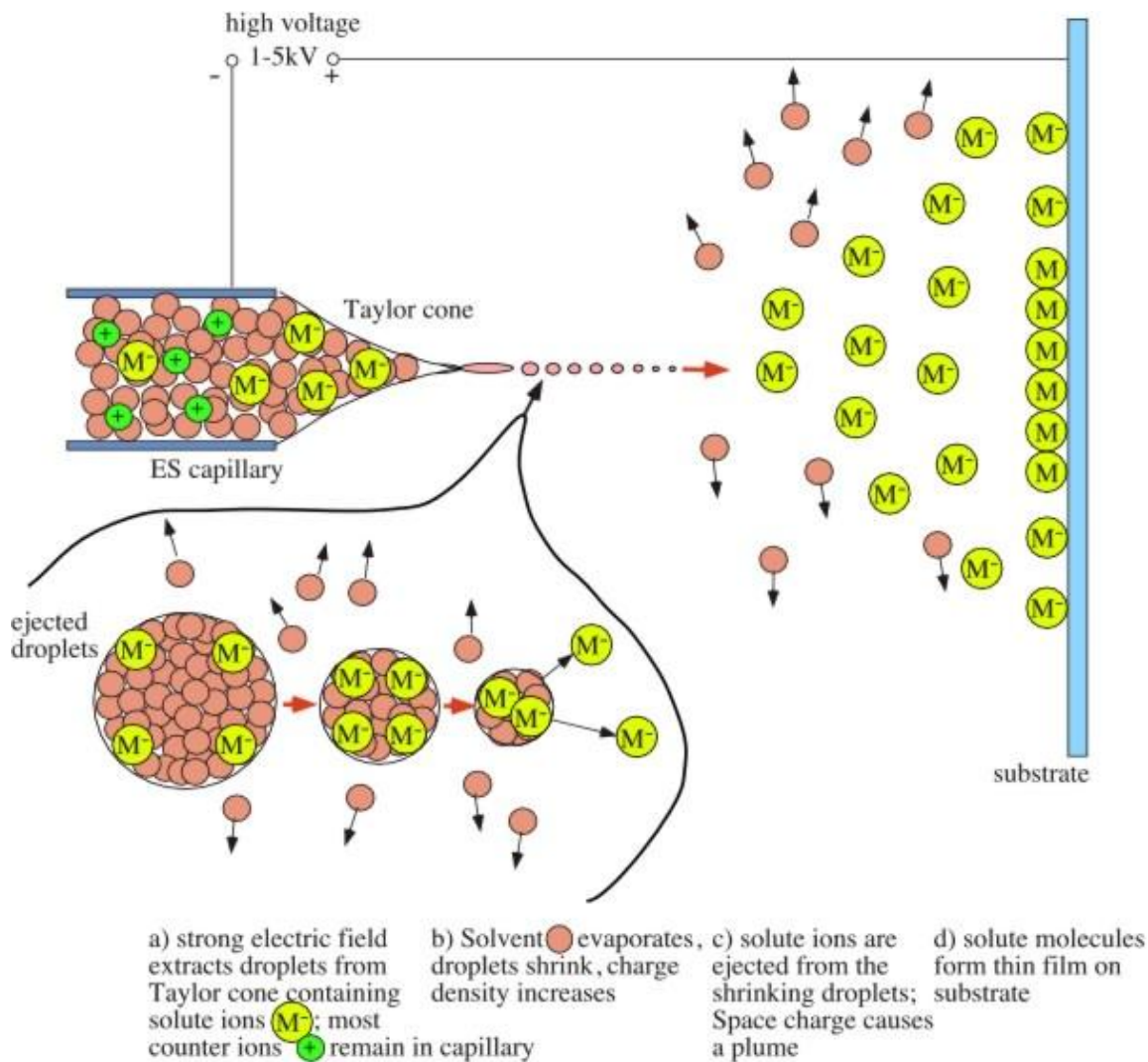


Figure 1: Schematic of the electro spray process. This image shows the charge separation in the electro spray emitter, the coulomb fission process and deposition of the solute ions (reprinted from [5]).

The process consists of applying a high electric field between an ESI emitter (a hypodermic needle in this case) and a counter electrode. Assuming a positive voltage is applied to the emitter, the positive ions within the needle tip will travel to the surface of the liquid whereas the negative ions will travel toward the emitter wall resulting in a partial charge separation [2]. The balance between surface tension of the solvent and force of the electric field on the ions on the

liquid surface creates a conical meniscus (Taylor Cone). If the field is strong enough, uniformly sized droplets will be emitted from the tip of this meniscus. Depending on the applied electric field several types of menisci can exist, with the previously mentioned Taylor Cone (or cone-jet) emitting the most stable and uniformly sized droplets. Figure 1: Schematic of the electro spray process. This image shows the charge separation in the electro spray emitter, the coulomb fission process and deposition of the solute ions (reprinted from [5]). shows the various forms of meniscus that can be present during electro spray.

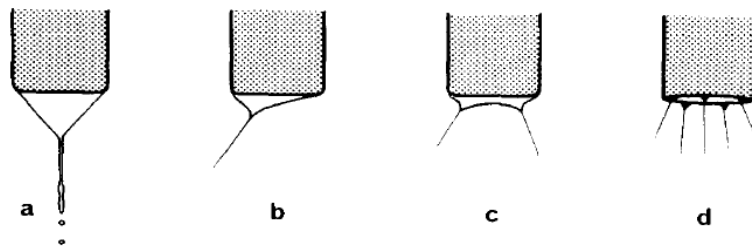


Figure 2: Various forms of menisci when performing electro spray. a, b) cone-jet mode; c,d) variants of this mode (reprinted with permission from [6])

After emission, these droplets travel in the direction of decreasing potential and the trajectories can be further modified by externally applied electric fields. The electric field necessary to generate the Taylor Cone at the needle tip is on the order of  $10^6 - 10^7$  V/m. The electric field at a needle and flat counter electrode is defined by the following equation:

$$E_c = \frac{2V_c}{\left[ r_c \ln \left( \frac{4d}{r_c} \right) \right]} \quad (1)$$

where  $E_c$  is the value of the electric field,  $V_c$  is the applied voltage,  $r_c$  is the outer radius of the capillary and  $d$  is this distance between capillary and counter electrode [7].

As the droplets travel through air they begin to evaporate. Upon progressive evaporation the radius of a droplet sphere will decrease but the charge within the droplet will remain the same. As a result the charge density within the droplet will increase until the Rayleigh Limit is reached whereupon Coulomb fission will occur. The Rayleigh Limit is the minimum radius of a charged droplet before it becomes unstable and is defined by the following equation:

$$q_R = 8\pi(e_0\gamma R^3)^{1/2} \quad (2)$$

where  $q_R$  is the charge,  $e_0$  is the permittivity of vacuum,  $R$  is the radius of the droplet and  $\gamma$  is the surface tension of the droplet fluid [8].

Coulomb fission is the process of droplet breakup in electrospray. The mutual repulsion of the ions on the surface of the droplet becomes greater as the droplet decreases in size. The force generated by this repulsion surpasses the surface tension of the solvent and smaller "satellite" droplets are emitted from the original droplet. This process repeats itself several times resulting in successively smaller droplets. The final product of several coulomb fissions is gas phase macromolecules.

There are currently two main theories on how the final gas phase ions are produced after the successive coulomb fissions; the Ion Evaporation Model (IEM) and the Charge Residue Model (CRM). According to the IEM, after several

coulomb fissions very small droplets on the order of 10 to 20 nm will remain. Spontaneous generation of gas phase ions will occur from these droplets. In the CRM the small “child” droplets emitted from the Coulomb fissions contain single ions. The solvent will evaporate from the droplet resulting in single ion residues without spontaneous emission of the ion from the droplet (unlike the IEM) [2].

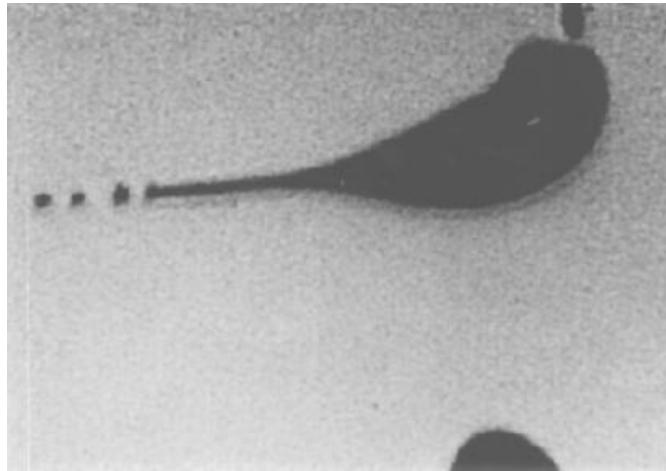


Figure 3: Photograph of a droplet undergoing coulomb fission. Notice the elongated conical meniscus and satellite droplets on the left of the figure (reprinted with permission from [8]).

The amount of current being generated at the ES emitter is a measure of the rate of ions leaving the emitter contained within the spray. This is the main measurable quantity obtained in the experiments of this study. This current can be calculated using the modified Hendricks equation shown below:

$$i_{ES} = H v_f^v \sigma_s^n E_c^e \quad (3)$$

where  $i_{ES}$  is the electrospray current,  $H$  is a constant that varies depending on the surface tension and dielectric constant of the solvent used,  $v_f$  is the fluid flow rate,  $\sigma_s$  is the specific conductivity, and  $E_c$  is the applied electric field [9].



Cloupeau [6] performed a systematic study to analyze the effect of the main parameters of an electrospray setup. These parameters include solution conductivity, needle diameter, applied voltage and solution flow rate. Some of the more significant findings of the study indicate that the measured current increases with increasing flow rate and solution conductivity and emitted droplet diameter increases with flow rate. Fernandez de la Mora and Loscertales found in their study that this current varies proportionally with the square root of flow rate and solution conductivity [10].

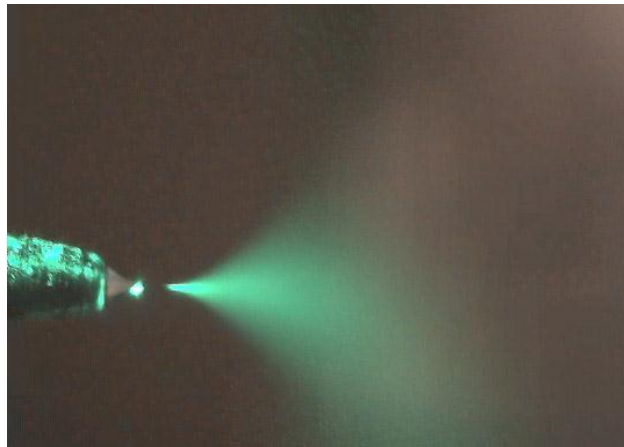


Figure 4: Image of electrospray emitter with cone-jet and plume. The image was illuminated with a green laser orthogonal to the direction of the camera (reprinted from [5]).

The necessary components for an electrospray apparatus are minimal and simple. A pumping device (in this study a syringe pump) is necessary to advance the solution at a rate of microliters per minute. A stainless steel hypodermic needle is used as the emitter. To this needle a high voltage, low current power supply is electrically coupled. A counter electrode (capillary)

serves as an interface between vacuum chamber and atmosphere and is placed on a lower potential than the needle.

### 3 Fluid Flow/Simulation Background

#### 3.1 Fluid Flow Background

Computational fluid dynamics (CFD) is a powerful tool for predicting complex flows for which no analytical solutions exist. Most real world cases consist of irregular geometries and fall under this category. Special consideration must be taken when performing these simulations as the common saying “garbage in garbage out” is quite valid. The two main considerations prior to performing CFD analysis where low pressures exist is whether the fluid can be treated as a continuum and whether the flow regime is laminar or turbulent.

For rarefied gases, statistical analysis and Monte Carlo simulations are common. Gas-wall interactions dominate this flow regime as opposed to continuum (or viscous) flow wherein gas-gas interactions dominate. When working with gases in a higher pressure regime the continuum theory holds and the bulk properties of the system can be computed without regard to the individual molecular components. To determine whether the continuum theory hold and viscous flow analysis can be used with regard to a system the Knudsen number must be calculated. The Knudsen number is a dimensionless quantity that compares the mean free path to the characteristic dimension of a flow and is defined as:

$$Kn = \frac{\lambda}{d} \quad (4)$$

where  $Kn$  is the Knudsen number,  $\lambda$  is the mean free path and  $d$  is a characteristic length of the flow. Based on this equation it is evident that as the flow becomes increasingly rarefied the Knudsen number will increase. Flows in which the  $Kn < 0.01$  can be regarded as continua [11]. All flows in this study fall well below this number and can therefore be analyzed as viscous fluid flows.

The Reynolds number is a measure of the forces due to momentum versus the viscous forces. The equation of the Reynolds number is given as:

$$Re = \frac{U\rho d}{\mu} \quad (5)$$

where  $Re$  is the Reynolds number,  $U$  is the flow velocity,  $\rho$  is the density and  $\mu$  is the dynamic viscosity. From a practical standpoint, the relative effect of viscosity serves to dampen the flow; therefore when the Reynolds number is low the viscous forces have a large effect resulting in a more stable flow. When the Reynolds number is high the momentum forces have a dominating effect. The result is less stabilization from the viscous forces and the potential for the onset of random fluctuation in flow parameters, commonly known as turbulence.

Estimation for the velocity of air through a tube should begin by taking into account whether or not the fluid becomes choked. Choked flow occurs when the fluid within a tube or orifice reaches the speed of sound and if the pressure at the low pressure outlet is reduced further the gas flow rate will cease to increase. This occurs because the fluid is travelling at the same rate that it would need to send the information back through to the high pressure inlet of the pipe to increase the flow. Because of this the, fluid flow will remain constant at the speed of sound (Mach number 1). For an orifice of zero length the ratio of the

low pressure outlet to the high pressure inlet for which air will be choked is 0.52 [11]. Although this value is valid only for an orifice it demonstrates that an outlet to inlet pressure ratio of only slightly more than half is low enough to choke the flow. This phenomenon is demonstrated through the simulations and is discussed in the simulation results section.

There are commonly accepted “critical” values of the Reynolds number that have been experimentally determined in which the onset of turbulence is likely. For a smooth tube with flow in the axial direction this value is 2100 – 2300. Inserting the minimum critical Reynolds number, using an approximated viscosity of air at 130 C and tube diameters of 20 and 30 mils critical velocities of 72 and 108 m/s were determined for the capillaries used in this study. This number is even lower as these critical Reynolds numbers are estimated for smooth tubes.

Because the flow is choked an estimation of the maximum velocity can be made and compared to the critical velocity for the laminar/turbulent flow regime discussed earlier. The speed of sound at 130 C is 402.49 m/s which far exceeds the critical value calculated previously. As a result, the flow regime was deemed turbulent for this study.

In practice, turbulence manifests itself at a point as a random fluctuation of a quantity as a function of time around a mean. The process of Reynolds decomposition mathematically represents this as:

$$\mathbf{u}(t) = \mathbf{U} + \mathbf{u}'(t) \quad (6)$$

$$p(t) = P + p'(t) \quad (7)$$

where  $\mathbf{u}(t)$  is the instantaneous velocity vector,  $\mathbf{U}$  is the mean value of the velocity vector,  $\mathbf{u}'(t)$  is the fluctuating part of the velocity vector,  $p(t)$  is the instantaneous pressure,  $P$  is the mean pressure value and  $p'(t)$  is the fluctuating part of the pressure. Inserting these ensemble averages into the Navier-Stokes equations results in additional stresses known as the Reynolds stresses. These are the product of the fluctuations of the three dimensional velocity components. The consequence of these additional stresses is that the system of equations is no longer closed and additional approximations must be made. This new system of equations is known as the Reynolds averaged Navier-Stokes (RANS) equations and is shown below.

$$\frac{\partial \bar{\rho}}{\partial t} + \text{div}(\bar{\rho} \tilde{\mathbf{U}}) = 0 \quad (8)$$

$$\begin{aligned} \frac{\partial(\bar{\rho} \tilde{U})}{\partial t} + \text{div}(\bar{\rho} \tilde{U} \tilde{\mathbf{U}}) = & -\frac{\partial \bar{P}}{\partial x} + \text{div}(\mu \text{grad } \tilde{U}) \\ & + \left[ -\frac{\partial(\overline{\rho u'^2})}{\partial x} - \frac{\partial(\overline{\rho u'v'})}{\partial y} - \frac{\partial(\overline{\rho u'w'})}{\partial z} \right] + S_{Mx} \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{\partial(\bar{\rho} \tilde{V})}{\partial t} + \text{div}(\bar{\rho} \tilde{V} \tilde{\mathbf{U}}) = & -\frac{\partial \bar{P}}{\partial y} + \text{div}(\mu \text{grad } \tilde{V}) \\ & + \left[ -\frac{\partial(\overline{\rho u'v'})}{\partial x} - \frac{\partial(\overline{\rho v'^2})}{\partial y} - \frac{\partial(\overline{\rho v'w'})}{\partial z} \right] + S_{My} \end{aligned} \quad (10)$$

$$\frac{\partial(\bar{\rho} \tilde{W})}{\partial t} + \text{div}(\bar{\rho} \tilde{W} \tilde{\mathbf{U}}) = -\frac{\partial \bar{P}}{\partial z} + \text{div}(\mu \text{grad } \tilde{W})$$

$$+ \left[ -\frac{\partial(\overline{\rho u' w'})}{\partial x} - \frac{\partial(\overline{\rho v' w'})}{\partial y} - \frac{\partial(\overline{\rho w'^2})}{\partial z} \right] + S_{Mz} \quad (11)$$

$$\frac{\partial(\overline{\rho \tilde{\Phi}})}{\partial t} + \text{div}(\overline{\rho \tilde{\Phi} \tilde{\mathbf{U}}}) = -\text{div}(\Gamma_{\Phi} \text{grad } \tilde{\Phi}) + \left[ -\frac{\partial(\overline{\rho u' \varphi'})}{\partial x} - \frac{\partial(\overline{\rho v' \varphi'})}{\partial y} - \frac{\partial(\overline{\rho w' \varphi'})}{\partial z} \right] + S_{\Phi} \quad (12)$$

where the tilde above the symbol represents the density-weighted form of the mean flow properties (i.e.  $\tilde{\mathbf{U}}$  represents the density-weighted average velocity),  $\Phi$  and  $\varphi$  represent a generic transported scalar,  $\Gamma_{\Phi}$  is the diffusion coefficient of this scalar and the  $S$  terms represent source terms.

To close this system of equations another approximation must be made. Boussinesq proposed that the mean rates of deformation are proportional to the Reynolds stresses. Mathematically this is represented by:

$$\tau_{ij} = -\overline{\rho u'_i u'_j} = \mu_t \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (13)$$

where  $k$  is the turbulent kinetic energy and  $\delta_{ij}$  is the Kronecker delta (when  $i = j$ ,  $\delta_{ij} = 1$ , when  $i \neq j$ ,  $\delta_{ij} = 0$ ). Similarly the transport of an arbitrary scalar value can be represented as:

$$-\overline{\rho u'_i \varphi'} = \Gamma_t \frac{\partial \Phi}{\partial x_i} \quad (14)$$

where  $\Gamma_t$  is the turbulent diffusivity. The RANS turbulence models approximate the turbulent viscosity using various constants or by introducing new transport equations for variables describing the turbulent action of the flow.

The two most commonly used RANS turbulence models are the standard k- $\epsilon$  model [12] and the Wilcox k- $\omega$  model [13]. Each of these models introduces new transport equations whose variables are used to calculate the turbulent fluctuations of flow parameters. Both models introduce the turbulent kinetic energy, mentioned previously and described by the following equation:

$$k = \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) \quad (15)$$

The quantity  $\epsilon$  is a measure of the dissipation of the turbulent kinetic energy by the eddies on the smallest scale from viscous stresses and units are  $m^2/s^3$ . It is represented by the following equation:

$$\epsilon = 2\nu \overline{s'_{ij} \cdot s'_{ij}} \quad (16)$$

where  $\nu$  is the kinematic viscosity and  $s'_{ij}$  is the component fluctuating component of the deformation rate tensor. The quantity  $\omega$  is the turbulence frequency with units  $s^{-1}$ . It is related to  $k$  and  $\epsilon$  by:

$$\omega = \frac{\epsilon}{k} \quad (17)$$



The approximations for the turbulent viscosity for the k-ε and the k-ω models are:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \quad (18)$$

$$\mu_t = \rho \frac{k}{\omega} \quad (19)$$

respectively.  $C_\mu$  is a dimensionless constant generally taken as 0.09.

Each of the two turbulence models mentioned has strengths and weaknesses. The k-ω model accurately predicts flows at low Re and solid surface boundaries but is very sensitive to the freestream value of ω. Conversely, the k-ε model is much less sensitive to the freestream value of ε but poorly predicts low Re flows and flows at solid boundaries. Because of this Menter developed the k-ω shear stress transport (SST) turbulence model which utilizes blending functions to employ the k-ω model near solid boundaries and the k-ε model away from it [14]. The Reynolds stresses and turbulent kinetic energy are calculated using the standard k-ω formulation. The extra transport equations for k and ω are:

$$\frac{\partial(\rho k)}{\partial t} + \text{div}(\rho k \mathbf{U}) = \tilde{P}_k - \text{div}[(\mu + \sigma_k \mu_t) \text{grad}(k)] - \beta^* \rho k \omega \quad (20)$$

$$P_k = 2\mu_t S_{ij} \cdot S_{ij} - \frac{2}{3} \rho k \frac{\partial U_i}{\partial x_j} \delta_{ij} \quad (21)$$

$$\tilde{P}_k = \min(P_k, 10 \cdot \beta^* \rho k \omega) \quad (22)$$

$$\begin{aligned}
& \frac{\partial(\rho\omega)}{\partial t} + \text{div}(\rho\omega\mathbf{U}) \\
& = \alpha\rho S^2 - \beta\rho\omega^2 + \text{div}[(\mu + \sigma_{\omega,1}\mu_t)\text{grad}(\omega)] \\
& + 2(1 - F_1)\rho\sigma_{\omega,2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad (23)
\end{aligned}$$

$$F_1 = \tanh \left\{ \left\{ \min \left[ \max \left( \frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega} \right), \frac{4\rho\sigma_{\omega,2}k}{CD_{k\omega}y^2} \right] \right\}^4 \right\} \quad (24)$$

$$CD_{k\omega} = \max \left( 2\rho\sigma_{\omega,2} \frac{1}{\omega} \frac{dk}{dx_i} \frac{d\omega}{dx_i}, 10^{-10} \right) \quad (25)$$

where  $\beta^* = 0.09$ . The constants  $\alpha$ ,  $\beta$ ,  $\sigma_k$ , and  $\sigma_\omega$  are made by blending the constants from the k- $\epsilon$  and k- $\omega$  models by:

$$C = C_1F + C_2(1 - F) \quad (26)$$

The values for the individual models are  $\alpha_1 = 5/9$ ,  $\beta_1 = 3/40$ ,  $\sigma_{k1} = 0.85$ ,  $\sigma_{\omega1} = 0.5$ ,  $\alpha_2 = 0.44$ ,  $\beta_2 = 0.0828$ ,  $\sigma_{k2} = 1$ ,  $\sigma_{\omega2} = 0.856$  and  $F$  is the blending function. The k- $\omega$  SST model has produced the best agreement with experimental results associated with capturing shock features when compared to several of the well-known turbulence models [15].

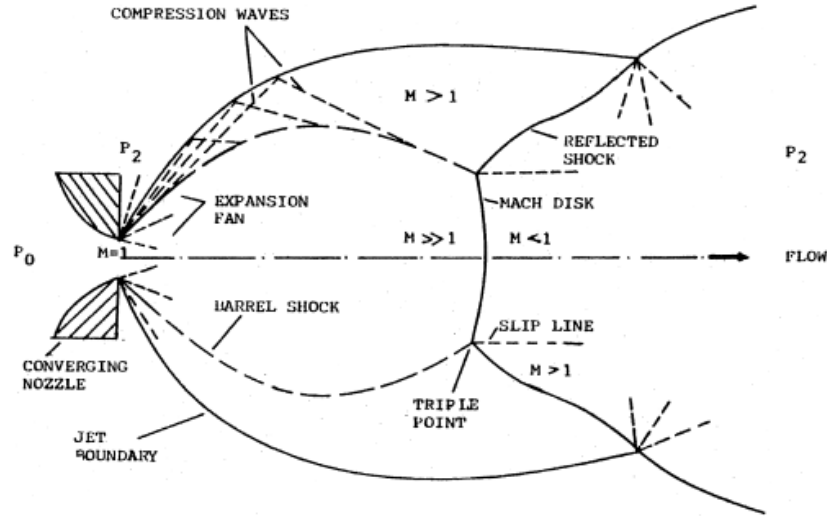


Figure 5: Schematic diagram of an underexpanded jet through a converging nozzle. Notable features are the Mach disk and barrel shock (reprinted with permission from [16]).

When high pressure ratios exist across an axisymmetric orifice or capillary interesting flow features arise due to the rapid expansion of the fluid. These features include the Mach disk, barrel shock, and zone of silence. These are characteristic features found in all underexpanded jets. Expansion waves travelling from the edge of the capillary will reflect off this boundary shifting to a compression wave. When they contact the expansion waves coming from the opposite edge of the capillary exit a shock is formed often referred to as the barrel shock [17]. The Mach disk is a normal shock that appears bound by the barrel shock at a distance defined by the following equation:

$$x_{MD} = \frac{2}{3} d \left( \frac{p_0}{p_2} \right)^{1/2} \quad (27)$$

where  $x_{MD}$  is the location of the Mach disk,  $d$  is the orifice diameter,  $p_0$  is the inlet pressure and  $p_2$  is the far field pressure. The equation for the Mach disk location assumes that the fluid is exiting from an orifice of zero length. In the case of this

study this is not true. Murphy [18] demonstrated numerically and experimentally that the sonic line which would be found at the exit of an orifice actually extends into the capillary by as much as one diameter length, causing a source of error when approximating the location of the Mach disk. The zone of silence is the region contained between the nozzle exit, barrel shock and Mach disk where the fluid flow expands radially as it travels along the axial direction. This is the area which is sampled when utilizing free jets for ESI-MS with skimmers [16].

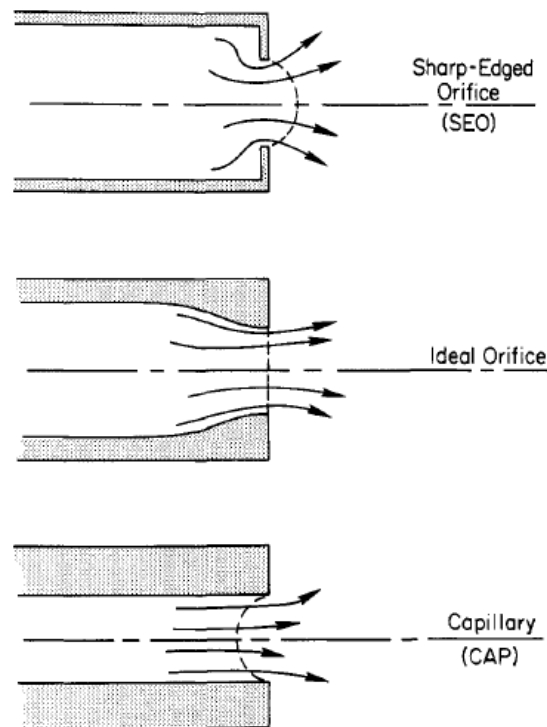


Figure 6: Locations of the sonic line for various nozzle geometries. Of importance to this study is the capillary nozzle (from [18]).

These shock features are often observed repeating downstream from the initial jet. This oscillation of the jet boundary is a result of the attempt to reach an equilibrium pressure by the flow. The expansion of the jet causes an undershoot

of the pressure. When this occurs the higher background pressure will force the flow to reverse inward toward the jet axis. The flow will then overshoot the background pressure resulting in a second expansion. This process will repeat several times downstream but due to mixing at the jet boundary will dissipate in energy.

### 3.2 OpenFOAM and the Finite Volume Method

The software suite used in this study was compiled from the Open Source Field Operation and Manipulation (OpenFOAM) C++ libraries. The source code is accompanied by a myriad of preprogrammed executable applications that include continuum mechanics solvers (over 80) and utilities for mesh creation, data manipulation and post-processing tasks (over 170) [19]. The software has a large online user community that contributes to the development of the software. The syntax required to solve the partial differential equations encountered in continuum mechanics is easily programmed using the OpenFOAM libraries. For example, to solve the equation:

$$\frac{\partial(\rho\mathbf{U})}{\partial t} + \text{div}(\rho\mathbf{U}\mathbf{U}) - \text{div}(\mu \text{grad } \mathbf{U}) = \text{grad}(P) \quad (28)$$

one would use the code:

```
solve
(
    fvm::ddt(rho, U)
  + fvm::div(phi, U)
  - fvm::laplacian(mu, U)
  ==
  - fvc::grad(p)
);
```

This example illustrates the simplicity with which the OpenFOAM solvers are programmed as well the ability to be understood by the end user.

OpenFOAM also supports parallel processing of cases using Message Passing Interface (MPI) using domain decomposition. Domain composition breaks the domain of a model into separate entities. The governing equations of fluid flow are then solved for each domain on a separate processor allowing for parallel computation of each. OpenFOAM provides a utility to perform the actual decomposition as well one to reconstruct the decomposed domain after the parallel processing has been completed.

The finite volume method is used by OpenFOAM to discretize the governing equations of fluid flow in a domain. The finite volume method approximates partial differential equations as algebraic equations. The physical domain over which these equations are solved for is broken into individual volumes (or cells) which are centered around nodes. These discretized equations are solved for at each node and can be used to approximate the continuous function found in the physical system.

To construct the discretized equations for each cell the differential equation being solved must be integrated over the control volume. The divergence and gradient terms can then be transformed into integrals over the surface of the face of the cell using Gauss's (divergence) theorem. The fluxes of the properties being calculated are required at these faces and are obtained by interpolation [20].

### **3.3 rhoCentralFoam**

The solver used in the present study is "rhoCentralFoam" which was first included in the OpenFOAM 1.5 release. The solver was developed by

Christopher J. Greenshields and utilizes the flux interpolation methods of Kurganov and Tadmor [21] and Kurganov, Roelle and Petrova [22]. The solver is a density based compressible solver capable of implementing turbulence models and utilizes an implicit Euler time discretization scheme. It was chosen for this study due to its ability to accurately capture shock features [23].

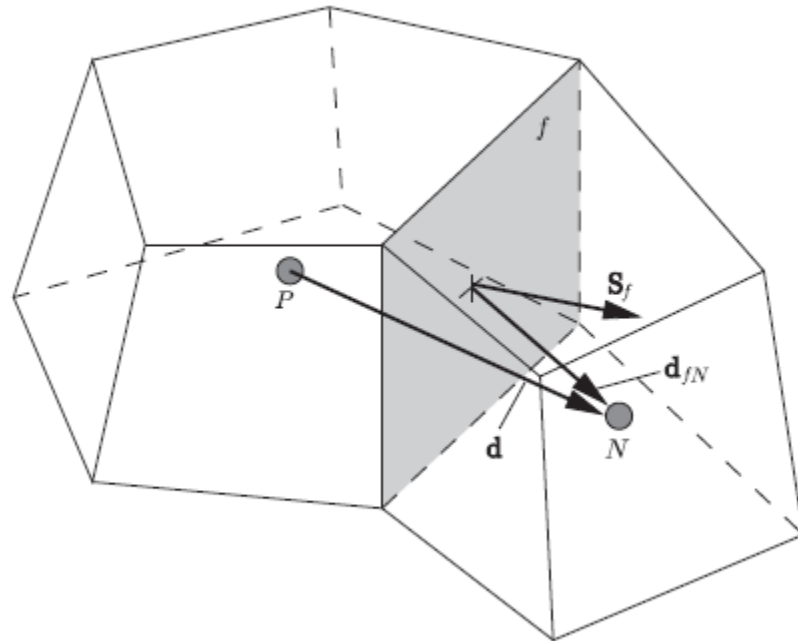


Figure 7: Schematic of discretization methodology used in OpenFOAM between cell P and neighbor cell N (reprinted with permission from [23]).

The flux schemes by Kuganov and Tadmor and Kurganov, Roelle and Petrova account for the fact that in compressible flows information about the properties of the flow are not only transported by the bulk fluid flow but also by wave propagation. For this study the flux scheme by Kurganov, Roelle and Petrova was used exclusively and will be discussed here. The discretization of the fluxes of convective terms at the faces of the cells is as follows:

$$\sum_f \phi_f \Psi_f = \sum_f [\alpha \phi_{f+} \Psi_{f+} + (1 - \alpha) \phi_{f-} \Psi_{f-} + \omega_f (\Psi_{f-} - \Psi_{f+})] \quad (29)$$

where:

$$\Psi_f = \omega_f \Psi_P + (1 - \omega_f) \Psi_N \quad (30)$$

where  $\Psi_P$  is the value of the flow property at point  $P$ ,  $\Psi_N$  is the value of the flow property at point  $N$ ,  $\phi_f$  is the volumetric flux through face  $f$ ,  $\alpha$  is a weighting factor and  $\omega_f$  is the diffusive volumetric flux. The + and – signs on the subscripts represent flux leaving or entering the face  $f$ . The volumetric fluxes across the faces associated with the local propagation speeds are defined and are used to define alpha:

$$\psi_{f+} = \max(c_{f+} |\mathbf{S}_f| + \phi_{f+}, c_{f-} |\mathbf{S}_f| + \phi_{f-}) \quad (31)$$

$$\psi_{f-} = \max(c_{f+} |\mathbf{S}_f| - \phi_{f+}, c_{f-} |\mathbf{S}_f| - \phi_{f-}) \quad (32)$$

$$c_{f\pm} = \sqrt{\gamma R T_{f\pm}} \quad (33)$$

$$\alpha = \frac{\psi_{f+}}{\psi_{f+} + \psi_{f-}} \quad (34)$$

where  $c$  is the speed of sound  $R$  is the specific gas constant and  $\gamma$  is the ratio of specific heats. This results in an upwind biasing of the fluxes and can be termed as a central upwinding scheme. The diffusive volumetric flux  $\omega_f$  is calculated by:

$$\omega_f = \alpha(1 - \alpha)(\psi_{f+} + \psi_{f-}) \quad (35)$$

where  $\gamma$  is the ratio of specific heats and  $R$  is the ideal gas constant.

Similarly, gradient terms are discretized as:

$$\sum_f S_f \Psi_f = \sum_f [\alpha \mathbf{S}_f \Psi_{f+} + (1 - \alpha) \mathbf{S}_f \Psi_{f-}] \quad (36)$$



Symmetrical, total variation diminishing (TVD) flux limiting schemes are also employed with this solver to prevent the appearance of oscillations around shocks. For this study the van Leer limiter was chosen [24].

The solution procedure for the solver employs equation splitting to first solve for the inviscid property equations and then correct their values using diffusive correctors. First the inviscid momentum equation is solved:

$$\left(\frac{\partial \hat{\mathbf{u}}}{\partial t}\right)_I + \text{div} \cdot [\mathbf{u} \hat{\mathbf{u}}] + \text{grad}(p) = 0 \quad (37)$$

$$\mathbf{u} = \frac{\hat{\mathbf{u}}}{\rho} \quad (38)$$

The inviscid velocity terms solved for in the inviscid prediction are inserted explicitly into the stress tensor of the viscous momentum correction equation and the resulting viscous momentum equation is solved:

$$\left(\frac{\partial(\rho \mathbf{u})}{\partial t}\right)_V - \text{div} \cdot (\mu \text{grad}(\mathbf{u})) - \text{div} \cdot (\mathbf{T}_{\text{exp}}) = 0 \quad (39)$$

The inviscid energy equation is then solved and the inviscid temperature is calculated using the inviscid energy predictor and the previously calculated velocity:

$$\left(\frac{\partial \hat{E}}{\partial t}\right)_I + \text{div} \cdot [\mathbf{u} (\hat{E} + p)] + \text{div} \cdot (\mathbf{T} \cdot \mathbf{u}) = 0 \quad (40)$$

$$E = \frac{\hat{E}}{\rho} \quad (41)$$

The temperature is then updated with:

$$T = \frac{1}{c_v} \left( \frac{\hat{E}}{\rho} - \frac{|\hat{\mathbf{u}}|^2}{2} \right) \quad (42)$$

A diffusive temperature correction is then performed to update the temperature value:

$$\left(\frac{\partial(\rho c_v T)}{\partial t}\right)_v - \text{div} \cdot (k \text{ grad}(T)) = 0 \quad (43)$$

The rhoCentralFoam solver assumes that the fluid acts as a calorically perfect gas where:

$$p = \rho RT \quad (44)$$

$$e = c_v T = (\gamma - 1)RT \quad (45)$$

$$\gamma = \frac{c_p}{c_v} \quad (46)$$

Where  $c_p$  and  $c_v$  are the specific heats at constant pressure and constant volume respectively. In addition, to model the transport properties of the fluid Sutherland's formula for the viscosity as a function of temperature was used and is given by:

$$\mu = A_s \frac{T^{1.5}}{T + T_s} \quad (47)$$

where  $A_s$  and  $T_s$  are the transport coefficients whose values for air are  $1.458 \times 10^{-6}$  Pa s/K<sup>0.5</sup> and 110.4 K respectively.

The solver has been verified against several benchmark tests including shock tube, supersonic forward facing step, supersonic biconic wedge and underexpanded free jet [25]. The results of the free jet comparison are shown in Figure 8 and show that rhoCentralFoam predicts the notable flow features with good accuracy.

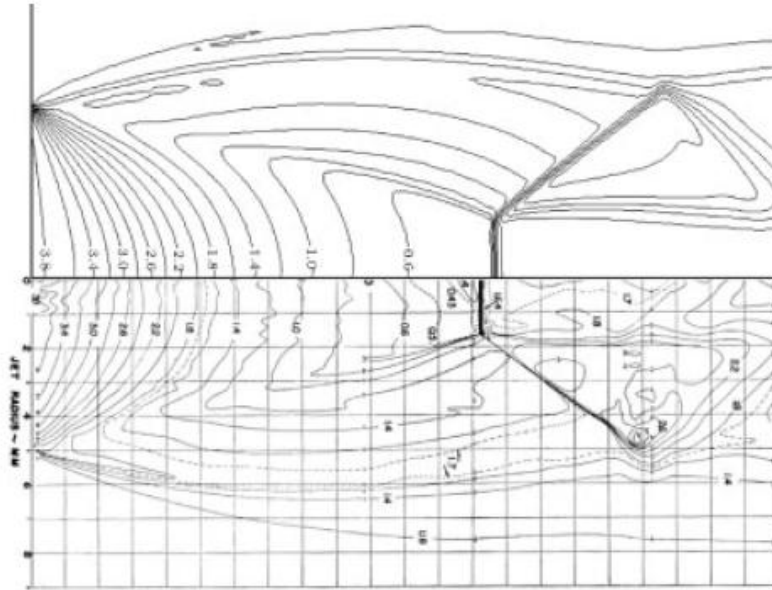


Figure 8: Numerical (top) and experimental (bottom) results of a underexpanded jet using the rhoCentralFoam solver (reprinted with permission from [23]).

### 3.4 SIMION® and the Hard Sphere Collision Model

SIMION® is a simulation software “suitable for a wide variety of systems involving 2D or 3D, static low-frequency (MHz) RF fields: from ion flight through simple electrostatic and magnetic lenses to particle guns to highly complex instruments, including time-of-flight, ion traps, quadrupoles, ICR cells, and other MS, ion source and detector optics [26].” SIMION uses finite difference methods over a Cartesian grid containing the geometry representing the electrodes of a system. The Laplace equation is solved to calculate the electric field in the area between these electrodes. Using Runge-Kutta calculations, ion flight paths can be calculated. The software has been used to predict the path of ions in the RF ion funnels of other groups [27]. The development of the base electrodynamic-only SIMION simulations in this study was performed by Dr. Mark Anthony.

The hard sphere collision model was employed to introduce the effects of gas-ion interactions in this study. Gas molecules and ions are idealized as rigid spherical objects that cannot occupy the same space as another particle. Individual collisions between gas and ions are simulated and are assumed elastic. The mean free path of the background gas is derived from the kinetic theory of gasses and provides a length scale between which these collisions will occur. This value is a function of the pressure and temperature of the surrounding gas. The vibrational velocities of the gas molecules are a function of temperature and follow a Maxwell-Boltzmann distribution. SIMION provides a built in collision model entitled "Collision Model HS1" which allows for the definition of bulk gas parameters over the entire domain. For this study modifications to this model's code were made to allow for the introduction of arrays of fluid flow parameters whereby the mean free path, vibrational velocities of the background gas and bulk velocity of the background gas could be calculated and allowed to affect the ions as a function of position. After performing the fluid flow analysis of the domain the flow property arrays could be combine with the simulations performed by Dr. Mark Anthony and the combined viscous and electrodynamic effects observed.

#### 4 Physical Apparatus

The electrospray chamber consists of three differential pumping stages. Each stage is under progressively higher vacuum in an effort to remove the impurities of the air and solvent prior to deposition. The first chamber (and focus of this study) is the “funnel chamber” and is held under the lowest vacuum. The pressure in this chamber ranges from 1 to 10 Torr depending on the capillary used for the atmospheric interface (discussed later). The second chamber, the “collision cell”, contains an RF quadrupole and is held at a vacuum pressure of 1 to 30 mTorr. The final chamber, the “deposition chamber” contains hardware for focusing and rastering the ion beam as well as a sample holder. This chamber is held at a pressure of  $10^{-5}$  to  $10^{-6}$  Torr. Each of the stages has a roughing pump attached to it. The ratings of these pumps are 21, 19 and 19 cubic feet per minute for the funnel chamber, collision cell, and deposition chamber respectively. The collision cell and deposition chamber each have turbovacuum pumps attached to them and these pumps are both rated at 230 liters per second. The entire system is generally held under vacuum at all times unless some internal hardware is being changed.

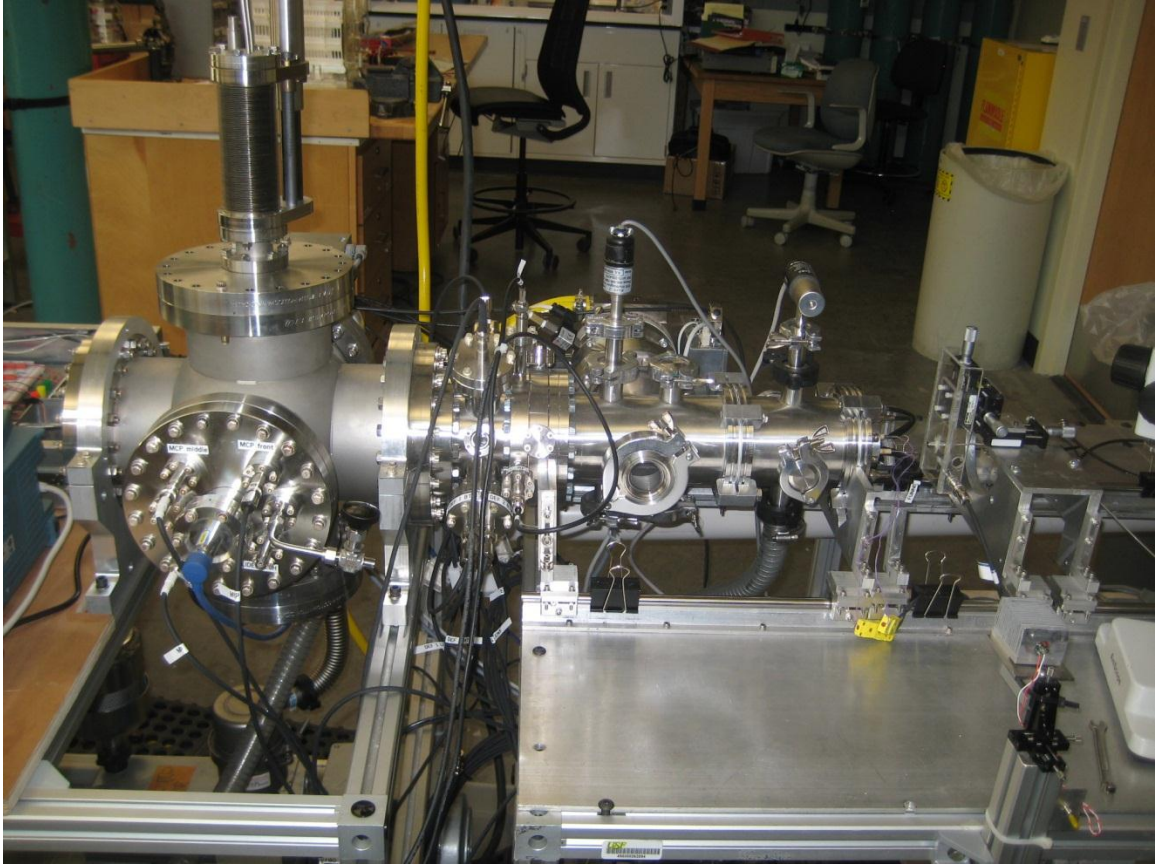


Figure 9: Overall view of the macromolecular patterning system. Starting from right to left: injection area, RF ion funnel chamber, collision cell chamber, deposition chamber.

The interface between atmospheric pressure and the first differential pumping stage is a  $1/16^{\text{th}}$  inch OD capillary. Two capillary IDs are used in this study and the results of these different diameters will be further discussed later in the paper. Generally a capillary of length 6.4 cm with an inner diameter of 20 mil is used. This provides a good compromise between throughput of ions and resultant pressures in the chambers. This capillary is situated within a heater block to increase the desolvation of droplets via IR radiation. The heater block is constructed of solid aluminum and is held within a Teflon sheath. A rubber O-ring is connected to the sheath to provide a seal between the edge of the heater

block and the flange of the chamber. To heat the block, two 35 W cartridge heaters are placed within holes inside of the heater block. To electrically isolate these cartridges, ceramic tubes are used as sheaths. A thermocouple, also in a ceramic sheath, is placed in another hole within the heater block. The cartridge heaters are connected to a solid state relay whose control leads are connected to an Omega PID heater controller. This controller is used to maintain a constant temperature on the heater block.

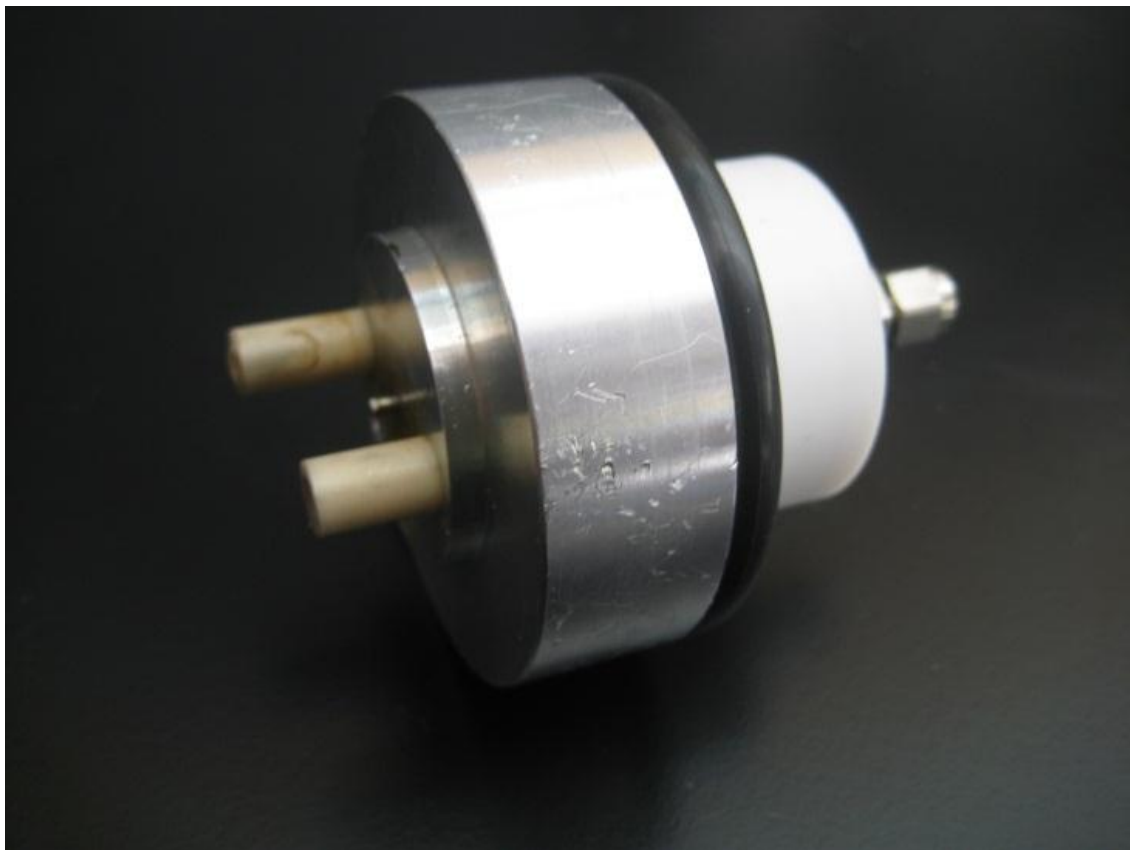


Figure 10: Image of the heater block with capillary. The ceramic sheaths that hold the heater cartridges are on the left and the Teflon sheath with o-ring can be seen on the right.

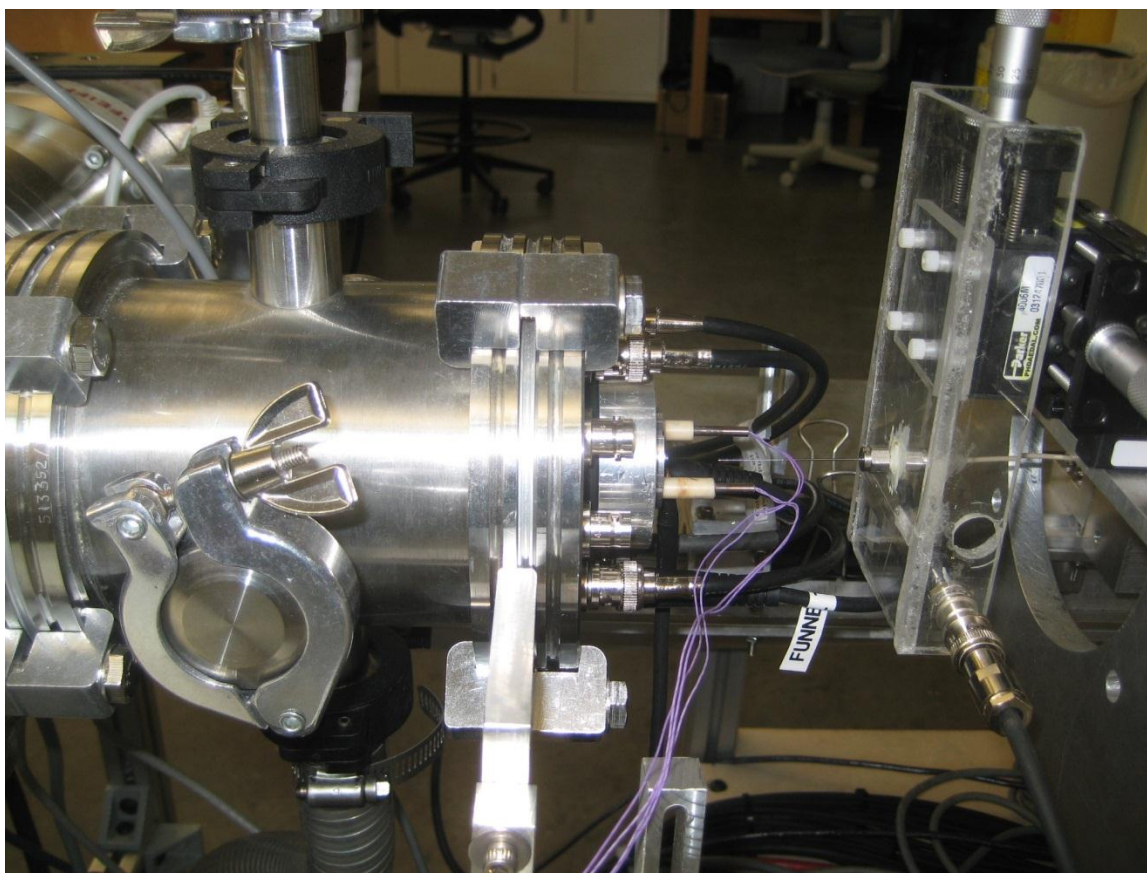


Figure 11: Injection area for electrospray. The needle can be seen in the area where the purple wires attached to the heating cartridges meet the chamber.

As stated previously, the main focus of this paper is on the ion funnel and funnel chamber. This type of ion focusing device was first proposed by Shaffer [27]. The ion funnel is used to progress the ions injected from atmosphere into the following chambers of the system using a DC voltage gradient. An RF voltage applied to the lenses helps overcome space charge issues of the similarly charged ions to compress them into a narrow beam. The RF voltage effectively creates a sharp voltage gradient at the inner edges of the lenses while creating a nearly field-free region along the axis of the funnel, thus forcing the ions toward it.



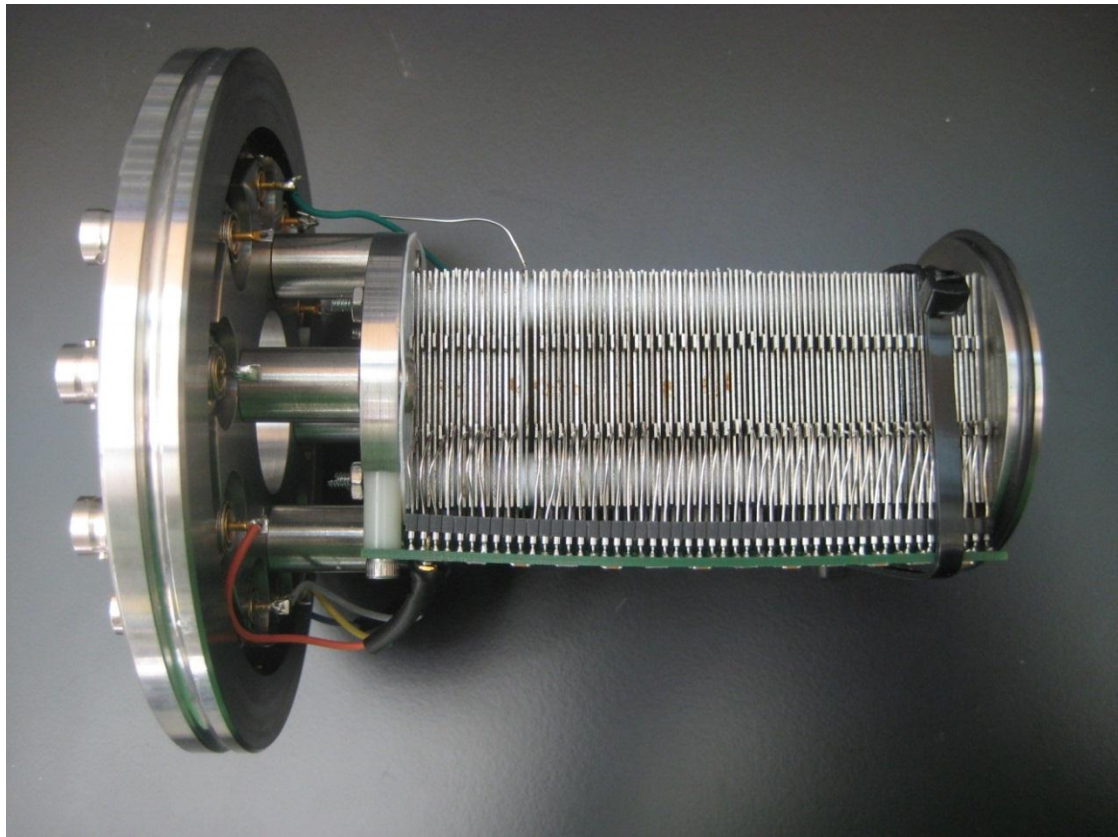
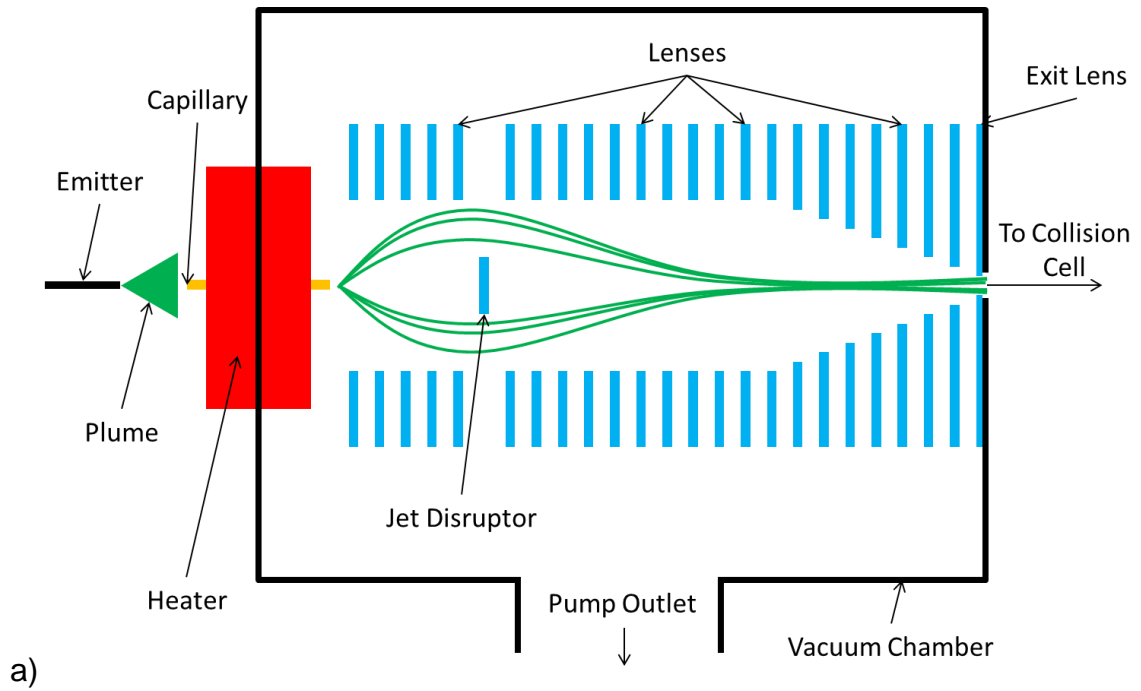


Figure 12: Schematic (a) and side view (b) of the RF ion funnel.

The physical layout of the funnel consists of 98 square stainless steel plates (lenses) with concentric holes in them. The side length of the outside of the lenses are 1  $\frac{3}{4}$ " and the thickness of each lens is 0.5 mm. The diameter of the hole in the first 55 lenses is 1" with the last 45 reducing from this size to ~2 mm. These lenses are supported by ceramic rods that run through holes in the four corners of the lenses. The lenses are separated and electrically insulated by 0.5 mm thick Teflon washers.

The lenses that would normally be located at positions 20 and 21 have been removed to accommodate another hardware feature of the ion funnel. In the location between where these lenses would be expected to be located the "jet disruptor" (JD) has been placed. The JD has the same dimensions as one of the first 55 lenses but instead of having a standard hole cut in it a small disk of 0.256" diameter has been left, supported by small beams coming from the inner diameter of the lens. The purpose of the JD is to reduce the number of large droplets that are emitted from the capillary and to allow the single ions to flow around it.



Figure 13: View of the funnel along the axial direction with the heater block/capillary assembly removed. The jet disruptor is clearly visible in the center of the funnel. Also visible are the BNC connectors used to supply the various voltages to the device.

As stated previously, to focus the ion cloud that enters the chamber both DC and RF voltages are applied to the lenses of the funnel. Two RF signals are generated at the funnel controller (the controller will be discussed later). These signals are sinusoidal waves and are 180 degrees out of phase with each other. Every other lens is connected in parallel through a capacitor with a value of 10 nF to one of the two signals so that two adjacent lenses will have inverse sine waves on them at any given time. The jet disruptor is not connected to the RF signal.

In addition, the lenses are connected resistively in series with their adjacent lenses with 510 k $\Omega$  resistors. In positive ion mode, the first lens will have a high potential applied to it (referred to as “funnel top” or FT) and the last lens will have a lower potential (referred to as “funnel bottom” or FB). Because the resistors connected in series with the lenses are all of the same value a linear voltage drop across the length of the funnel is generated. This provides a driving force for the ions to move in the direction toward the end of the funnel.

The jet disruptor is connected to an independent power supply and is set to a potential that approximates that of the lenses it has replaced. This provides a repulsive force for the ions travelling towards it. Because of the higher momentum of the liquid droplets, the jet disruptor will capture these and push the light single ions away from it. The jet disruptor acts as a mechanical filter for the usable, focusable current.

The final component of the funnel is the “exit lens” (EXT). This is a lens placed after the final funnel lens that is set to its own potential and isolated from the RF signal. This lens is always set to a lower potential than FB and is used to draw the ions out of the chamber. All of the voltages on the components of the funnel are of the same polarity as the needle and capillary voltages.

All of the active components of the ion funnel are connected via BNC cables. The funnel itself is mounted to the flange on the front of the chamber where the heater block is also situated. On the outside of this flange there are six BNC plugs; one for each DC voltage and two for the RF signals. These BNC cables are connected to the funnel controller where they can be used for either

outputting the required voltage or measuring the current discharge on the component. The measurement process will be reviewed later in the paper.

The ion funnel is regulated with an in-house built controller that contains power supplies for each DC component, an RLC circuit for generating the RF signals and relays to route the voltages when the funnel is on and the currents when measuring from the funnel. The DC power supplies contain a DC to DC converter controlled by a DAC and OP amp as well as four relays. These converters are rated for 0 to 300 volts in both positive and negative polarities. The actual minimum and maximum absolute values for the DC to DC converters are around 10 to 260 V. To overcome the minimum voltage limitation a relay was added that allows switching to a voltage divider which will use the entire range of output voltages of the DC to DC converter to generate the lower voltages when necessary. A second relay is included that switches the voltages from positive to negative polarity. A third relay grounds the output of the DC to DC converter to prevent any unwanted power surges when using the BNC connectors as inputs. The final relay switches the BNC connector from output mode to input, routing it to a single BNC connector where it can be connected to a picoammeter for current measurement.

To generate the RF signal, an RLC circuit was created with the resistive and inductive components located within the controller box. The capacitive component of the system is the funnel. A DAC controlled frequency generator is used to create a square wave of the desired output frequency. Using a DAC and an OP amp in combination with a linear voltage regulator a higher voltage can be

supplied to a gate driver which amplifies the signal from the frequency generator. The output of the gate driver is connected to a flip-flop which inverts the signal. These become the two inverted signals that are passed to adjacent lenses within the funnel.

The next section of the apparatus is the collision cell. As stated earlier this contains the RF quadrupole. The purpose of this chamber is to slow the ions down to a uniform speed in an effort to give them the same amount of kinetic energy. This chamber consists of four stainless steel rods whose axes are parallel to the direction of the ion beam. The rods are placed in a square pattern around the beam. As in the ion funnel, two 180 degree phase shifted RF signals are passed through these rods with adjacent rods utilizing opposite signals. The collision cell is not energized or used for its focusing properties in this study; instead it is used to measure the current that passes through the funnel chamber which will be discussed later.

The deposition chamber consists of two extraction lenses (one electrode each), a transfer lens (three electrodes) a condenser lens (three electrodes) an aperture (one electrode), and an objective lens (three electrodes each) as well as rastering plates. These are stainless steel cylinders with holes in the center. Voltages on the order of several kilovolts are applied to these elements and are polarized opposite the funnel voltages. The rastering plates are two sets of four plates placed in two square shapes with the center of the square on the beam axis. The two squares are placed one after the other with opposite voltages used on each set. When a voltage is applied to either the top or side plates the beam

passing through will be deflected vertically or horizontally. When the first plate deflects the beam it travels skew from its original axis. However since there are two sets of plates with exactly opposite voltages the second set will correct the skewness and place the beam parallel to its original axis. In this way the beam can be scanned over an area and a pattern can be created. This chamber will not be discussed in detail as it is not used for this study.

## 5 Experimental Procedures

### 5.1 Procedure for Electrospray Measurements

The first step to perform electrospray measurements is the creation of the solution. In this study the molecule Cytochrome C is used. This is a protein used in a cell's metabolism process and is found in the inner membrane of the mitochondria. Each batch of solution is produced in 20 mL volumes. First, 10 mg of Cytochrome C is dissolved into 1.8 mL water. This solution is stirred until no particulate Cytochrome C is visible. Next 18 mL methanol is added. Finally 200  $\mu$ L acetic acid is added to the solution. The result is an approximate 9:1 methanol to water ratio with 1% acetic acid. The purpose of the acetic acid is to increase the available  $H^+$  in solution and ensure the positive charge of the ions.

Subsequently, the solution is loaded into a 1 mL Hamilton glass syringe. The syringe is then connected to a PEEK tubing line that terminates in a stainless steel Hamilton 7785-01 hypodermic needle (ID 474  $\mu$ m). The solution is pumped through the tubing to ensure that no air remains in the line and the syringe is then placed in a Harvard Apparatus Pico Plus syringe pump. The mount for the needle is on a rail and must be slid into place before beginning electrospray. The tip of the needle is typically placed within several millimeters of the capillary entrance.



The third step is to set to the desired temperature using a Eurotherm 2132 PID controller. Two Stanford Research Systems PS 5350 high voltage power supplies are used to energize the needle and capillary/heater block assembly. These voltages are then set to a few kilovolts and a few hundred volts respectively and the electrospray is induced. To monitor the status of the spray a long working distance microscope is placed above the needle. A green laser diode is shown through the space in between the needle tip and capillary in an effort to illuminate the plume of droplets being emitted from the Taylor cone.

Once a stable spray has been established measurements can be obtained. The measureable quantity with electrospray is the current discharged at different points within the system. Since the goal is to pass as much current through the funnel as possible under the desired conditions the ratio of current out to current in is generally calculated. Using the controller and software developed in LabVIEW, measurements for the current into the chamber can be taken individually or in any combination of the active ion funnel components using the picoammeter. While a measurement is not being taken on one component yet on another, the potential of the component not being measured is set to ground. Generally, the total current discharging on all of the funnel lenses, jet disruptor and exit lens is taken as well as the current being discharged solely through the jet disruptor. The rationale for this procedure will be discussed further in the results and discussion section. To measure the current passing through the funnel the picoammeter is connected to the rods of the quadrupole. Using the

LabVIEW program measurements can be taken over time and the average of several measurements along with the standard deviation can be obtained.

To measure the current flowing out of the chamber the funnel must be energized and current must be measured on the rods of the collision cell. The rods are connected in parallel and routed to the picoammeter. Values for the potentials of funnel top, jet disruptor, funnel bottom and exit lens are then set. In positive ion mode each of these values is of a lower potential than the previous one in order to create a voltage gradient promoting the flow of ions out of the funnel. The frequency and voltage of the gate driver are set to create the RF signal on the lenses. Generally the resonant frequency of the RLC system is used (~610 kHz) and a gate driver voltage corresponding to a peak-to-peak voltage greater than or equal 100 V (>14 V) is used. The current on the collision cell rods can be measured in the same way as the current on the funnel components.

## **5.2 Procedure for Simulation**

### **5.2.1 Definition of Fluid Flow Problem**

The first step in development of the fluid flow model for this project was to define the scope of the problem. The area of interest included the region of flow from the exit of the capillary to the exit of the first differential pumping stage into the collision cell chamber. As the ID of the capillary is 0.020" the expected jet and shock features were expected to be on the same order as the capillary ID [16]. As stated earlier the IDs of the lenses at the entrance of the funnel are 1" and taper down to ~2 mm at the end of the funnel 18.3 mm from the capillary exit

and the jet disruptor, is 100.5 mm from the capillary exit. The edge length of the side of the outside of the square funnel lenses is 1  $\frac{3}{4}$ ". Because the size of these geometrical features is much larger than those of the associated flow features, many of these features were omitted in the fluid flow model. The main area of interest lies in the region between the inner edges of the lenses and the centerline of the funnel. Features outside of this area were assumed to not have a great impact on the flow in the area and for this reason were not included in the geometrical model. The areas included in the model were decided to be a small volume outside of the atmospheric side of the capillary, the entire length of the capillary, the volume of the funnel chamber from the centerline to 1.6 cm radially outward (including portions of the lenses), the jet disruptor and a small volume after the last lens of the funnel. The initial volume on the atmospheric side of the capillary was chosen to ensure that the fluid would have the correct velocity and pressure values as they entered the capillary, and not to rely on some approximation. The volume after the last lens of the funnel was added for the same reason as the atmospheric volume; to ensure that there were no approximations of these characteristics as they exited the system to the next chamber.

Given that the inner edges of the capillary, inner edges of the lenses, jet disruptor and funnel chamber exit orifice are circular and concentric around the centerline of the funnel length the geometrical dimensions of the simulation were chosen to be axisymmetric. The four support beams for the jet disruptor are 0.020" thick and extend from the edge of the jet disruptor to the inner edge of the

lenses. Due to the small area normal to the flow direction these supports they were assumed to have minimal effect on the fluid flow and were omitted in an effort of maintain the axial symmetry of the model.

OpenFOAM contains solvers for various fluid flow conditions including incompressible, compressible, laminar and turbulent flow regimes. Because of the great variation in pressures (atmospheric to approximately 1000 pa) over a short physical distance the flow was deemed compressible. The Reynolds number was taken to determine whether to treat the fluid as laminar or turbulent.

The system to be simulated would need to make use of a compressible solver that was able to employ turbulence models. Because of the strong difference in pressure and early conductance estimations there was a suspicion that the fluid may actually reach sonic and supersonic speeds as it exited the capillary into the funnel chamber. A solver with the ability to capture shock features was also a requirement of the simulation. Time resolved solutions were not necessary for this study, as the only time of interest is when the system has become steady. OpenFOAM offers various compressible, turbulent, steady-state solvers. Unfortunately none of these steady solvers are able to capture shock features and converge to an acceptable solution. Because of the limitations of the software available the transient solver “rhoCentralFoam” was chosen.

### **5.2.2 Creation of Model Geometry**

The next step in the development of the fluid flow simulation of the funnel chamber was the creation of a computerized model of the system. The main challenge was to ensure that the dimensions of the model used for the fluid flow

simulation were identical to those used in the SIMION model. This would allow for the most accurate coupling between the programs. SIMION uses a Cartesian mesh with a grid spacing of 0.1 mm therefore all physical entities within the fluid flow model would be accurate to this scale. Although the geometry contains many elements, the overall shape is relatively simple. Because of this, the software GMSH [28] was chosen to model and mesh the model without the use of external CAD software. This simplified the process two fold; first it reduced the need to convert from the CAD software file type to an acceptable GMSH file type, second it eliminated inherent rounding and approximation errors caused when this conversion occurs, causing the OpenFOAM developed model to not line up with the SIMION model.

To create the model the basic GMSH CAD engine was used. Because only cross-sectional areas of cylindrical elements were being drawn the resulting model was limited to straight lines. Details of the funnel electrode spacing can be found elsewhere in this document. After the two dimensional cross sections of the system were drawn the model was rotated around its centerline axis and extruded in the opposite direction so that the center of the volume was straddling the plane being modeled. OpenFOAM has specific requirements when modeling axisymmetric systems which provides the rationale for doing this procedure. Because the software is strictly finite volume, no two dimensional meshes are permitted. An extruded wedge of the desired modeling plane, with an angle of 0.08 rad (approximately 5 degrees) around the centerline axis, is required. When

this model is meshed the surface meshed of the two sides of the wedge must be the same and there must only be one element between them.

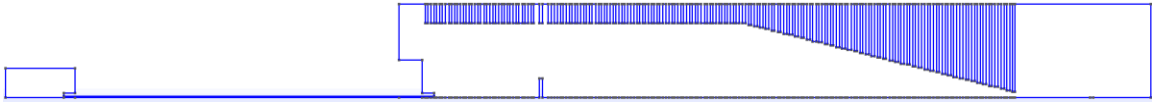


Figure 14: Image taken from GMSH showing the axisymmetrical model used for the fluid flow simulations.

The final step in creating the digital model of the system in GMSH is to define the boundary surfaces. After the extrusion of the symmetry wedge new surfaces are created where the boundary edges were which can then be selected as “Physical Groups”. The native file format of GMSH is an ASCII file which can be easily edited. Once these groups have been selected the actual name of the group can be edited using a text editor. Due to OpenFOAM convention, all surfaces must have a physical group definition as well as the volume. Each of the sides of the wedge must be labeled as its own physical group because of way in which OpenFOAM treats axisymmetric models.

### 5.2.3 Meshing the Model

As previously mentioned GMSH served as both the CAD software and meshing utility. Because of OpenFOAM’s requirements for creating axisymmetric meshes, only one element width between the two sides of the model wedge is permitted. GMSH’s built in extrude utility easily facilitates this. Using this utility, the geometry and surface mesh of the two dimensional cross section is copied and rotated.

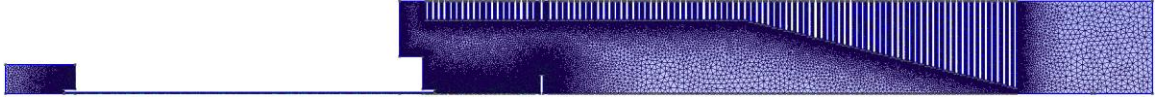


Figure 15: Image of the meshed geometry from GMSH. Note the high density of elements around the capillary exit, jet disruptor, heater and lenses.

The automated meshing portion of GMSH uses the Netgen algorithm [29] which generates a triangular surface mesh and tetrahedral volume mesh. The software also supports several recombination algorithms that will combine tetrahedra into hexahedra. Only the surface meshing capabilities of the software were utilized in this project as the volume mesh was just one element wide and comprised of links connecting the identical surface meshes of the wedge sides. The main challenge when creating an effective mesh for any type of numerical simulation is a tradeoff between accuracy and computation efficiency. As a result, it is preferable to place as few elements as possible while still maintaining the desired accuracy of results. Areas where very strong gradients exist are the most important locations to create a high mesh density.

The Knudsen number of a flow is a measure used to determine whether the fluid can be treated as a continuum or as discrete particles and is defined as the ratio of the characteristic length of the flow over the mean free path. At Knudsen numbers less than 0.01 the fluid can be treated as a continuum and special properties can be attributed to the flow. The most important of these properties is the “no-slip” condition that exists at solid boundaries within a fluid system. This property states that at a solid boundary the fluid particles in immediate contact with the wall move at the same velocity as the wall. This states that for stationary walls, such as the ones in this system, the fluid is not

moving when in contact with the walls. Very strong gradients are produced at an increasing distance from the wall due to the viscous forces causing the fluid to move at different speeds, going from zero velocity to the bulk flow velocity at a far distance from the wall.

Generally, prismatic boundary layer meshes are placed on the walls that grow geometrically in size normal to the wall. GMSH provides very little support for such meshes but allows for the creation of tetrahedral elements with similar growth properties. The “Boundary Layer” utility included in GMSH can create mesh with specific element sizes normal and tangential to the wall with a specified normal growth ratio. The tangential and normal element sizes for the heater, jet disruptor, lenses, and chamber walls sections are 0.0001 and 0.0001 and the growth ratio is 1.1. The tangential and normal element sizes for the capillary wall 0.00005 and 0.00003 and the growth ratio is 1.05. The element size far from the wall is 1 mm. GMSH defines its element sizes by ensuring that the element will fit within a circumscribed circle with a diameter of the specified element size.

After the model has been meshed GMSH outputs the file as a \*.msh file type. OpenFOAM uses its own meshing file system so the GMSH mesh must be converted. The standard OpenFOAM package includes a utility, “gmshtofoam” that converts this mesh into the native OpenFOAM format. The output of this utility labels all boundaries as the standard “patch” type. When working with symmetry planes, axisymmetric models or boundaries that require the use of turbulent wall functions, the boundary ASCII file within the OpenFOAM mesh file



system must be edited to ensure that each boundary is properly classified. In this case the planes that make up the wedge must be classified as the “wedge” boundary type. The capillary walls, chamber wall, lenses and jet disruptor must be classified as the “wall” type.

#### **5.2.4 Definition of Simulation Parameters**

The OpenFOAM framework does not provide any graphical user interface to modify or initiate the solution parameters. Instead, users must modify ASCII files, called dictionaries that contain all of the information required to run the simulation. Within each time folder there are dictionaries that identify the value of the internal field of the model as well as the values or conditions on each boundary for each physical property that is calculated in the simulation. To initialize the simulation the files in the “0” folder are adjusted to the initial conditions of the system. For a turbulent simulation velocity, pressure and temperature must be defined as well as turbulent viscosity, thermal diffusivity, turbulent kinetic energy, and a turbulent frequency. Each boundary must be listed in each of the property dictionaries and the condition associated with this boundary, as well as supplemental parameters for these conditions will be placed in this file. The only unusual boundary types that are associated with this system are the wedges. They must be labeled as the “wedge” boundary type in each one of the property files to ensure that the simulation will run as an axisymmetric model.

The next step in initializing a simulation in OpenFOAM is to determine which numerical schemes will be used. This will determine how the software will

numerically perform derivatives, including gradient, divergence and laplacian, interpolation and how time is discretized. To modify these parameters the “fvSchemes” dictionary must be edited. There are six subdictionaries within this file “interpolationSchemes”, “snGradSchemes”, “gradSchemes”, “divSchemes”, “laplacianSchemes”, “timeScheme” and “fluxRequired”. For each specific term in the finite volume equation to be solved a scheme must be selected under the appropriate subdictionary. A default scheme can also be selected for all terms of a specific type. The best practices for the “rhoCentralFoam” solver have been previously established [23].

As in the previous step, there exists another dictionary that is required to determine how OpenFOAM will solve the systems of equations as they are generated using the numerical schemes. Within this directory each property of the flow that is to be solved for will be assigned an appropriate linear solver. For steady simulations the convergence criteria are normally listed here. As noted earlier, best practices have already been established for the “rhoCentralFoam” solver.

The final step in the pre-processing phase of the simulation is to establish the time and data input and output control. This is again done with a dictionary entitled “controlDict”. Here the user will define what the time step of the simulation will be and when the data will be written to the time directories. Other parameters entered here include the precision of the time and output data. To establish the time step users have the option of using a fixed time step or basing the time step on the Courant number. For this system the time step was limited

to ensure that there was a maximum Courant number of 0.5. It was found that using a higher Courant number caused the system to become unstable and led to numerical errors. A write interval of  $5.0 \times 10^{-6}$  s was chosen as it allows observation the development of the flow with reasonable resolution while not using excessive hard disk space. All of the control files for the simulation can be found in Appendix A – OpenFOAM Simulation Configuration Files

### **5.2.5 Running the Simulation**

To solve the simulation the system was run in parallel on the CIRCE computing cluster. Parallel processing in OpenFOAM is done using a method called domain decomposition. In domain decomposition the geometry is broken up into parts and each of these parts is associated with a processor. Each sub-domain is then solved its own processor allowing for parallel computation of several parts of the domain.

The first step in performing the parallel computation of the simulation is to decompose the domain. OpenFOAM provides a standard utility that automatically performs this operation, “decomposePar”. The parameters for the domain decomposition are modified in the associated dictionary “decomposeParDict”. Because of the relative simplicity of the geometry of the domain in this system the “simple” decomposition method was chosen. This method decomposes the domain according to the number of splits in the x, y and z directions specified in the “decomposeParDict” dictionary. The simulation was run on 64 processors and the domain was decomposed accordingly. Since the

model was much larger in the y-direction than in the x-direction the model was simply split into 64 even parts in the y-direction.

The “decomposePar” utility was then run on the case. This created 64 directories, each of which contained a folder with the “0” time and a folder for the mesh part associated with the decomposed section of the geometry. These folders, along with the “controlDict”, “fvSchemes” and “fvSolution” folders were then uploaded to the cluster servers and run using MPI.

### **5.2.6 Post-Processing the Simulation**

The standard OpenFOAM package comes with several post-processing utilities as well as a data visualization application, ParaView. ParaView allows the user to graphically display the data from all calculated fields of the simulation. Images, videos as well as plotted data can be extracted using this application. The utilities incorporated in OpenFOAM for this project include the “sample” utility and the “reconstructPar” utility.

Visualization of a decomposed case can be done in one of two ways. The first option allows the use of the ParaView application on an individual processor domain, which only displays the field data for that domain. The second option, the one chosen for this project, was employs the use of the “reconstructPar” utility which takes the data from all of the processor domains and recombines them into a single domain. In this way all of the data for the entire domain can be visualized and processed in ParaView. The simulations were reconstructed prior to visualization in this study.

To import the data from OpenFOAM into SIMION the “sample” utility was used. As stated previously, the SIMION mesh is a Cartesian grid with spacing of 0.1 mm therefore the data taken from OpenFOAM would need to be taken at this scale. The “sample” utility allows users to provide a list of points from which to sample in the “sampleDict” dictionary. A simple script written in gawk (a text processing language for Linux) was developed to generate the 208,000 points over the domain used in SIMION. The SIMION mesh contains all points within the bounding box of the domain; therefore there were certain areas that would not output any data from the “sample” utility (i.e. in areas where solid surfaces existed in the OpenFOAM mesh). To overcome this limitation, another script was written in gawk which would fill in the “missing” data for output to SIMION. The script would scan through the data and whenever a missing data point existed the previous value would be inserted. In this way “dummy” data could be inserted into the areas of the lenses to ensure that the same number of data points existed between the two sets of data (a requirement of SIMION).

### **5.3 Development of LabVIEW Interface**

#### **5.3.1 Theory Behind the Interface**

National Instruments LabVIEW is a software program that allows users to easily create graphical user interfaces (GUIs) and front end controls for hardware. Little to no coding is required for a functional interface as the “programming” is done by linking functions and operators using wires as the signal flow. The software suite provides easy access to serial communications which many modern pieces of hardware use to interface with the computer. For

these reasons LabVIEW was chosen as the method used to interface the developed hardware with the computer.

For this project the funnel controller was developed using the Arduino development platform. This platform consists of a ready-to-use ATmega-328P microprocessor, programmer and prototyping board in one that uses USB as a serial interface to the computer. The software provided to program the board is its proprietary Arduino programming language, which is very similar to C++ and shares many of the important features, including object oriented programming. The environment used to program the Arduino programming language is based on Processing. The hardware contained in the Arduino development board can be easily replicated and was done so in the later versions of the funnel controller while still using the Arduino programming language to develop the firmware contained on the processor.

LabVIEW provides a toolkit to interface with the Arduino development platform called the LabVIEW Interface for Arduino (LIFA). Included in this toolkit are additional VISA drivers for serial communication, several virtual instruments (VIs) for sending and receiving data between LabVIEW and the Arduino and a script (sketch) to be placed on the Arduino. These special VIs can be placed on the LabVIEW interface and are used to send communication packets to the Arduino via the serial port. These packets contain information on the specific commands the Arduino is to perform, additional data these commands will utilize as well as some validation criteria to ensure that the packet was properly sent. This communication protocol will be discussed later in this document.

The LIFA toolkit was designed so that the majority of the processing was occurring on the LabVIEW side of communication. All of the decisions would be made via LabVIEW then sent to the Arduino to perform the desired task. When the Arduino receives the packet from LabVIEW it processes the packet and selects one of several preprogrammed commands from a case structure for the Arduino to perform. These are mostly low level microprocessor functions such as setting a digital pin or reading an analog pin. The communication between the two was also synchronous. The only time LabVIEW had the ability to receive data from the Arduino was if the Arduino was polled by LabVIEW for information. For this application it was more desirable to have the Arduino do the majority of the processing and use LabVIEW solely as a terminal with which to communicate to the Arduino. This required some modification of the LIFA sketch on the Arduino, an additional communication protocol to packetize information from the Arduino to LabVIEW, and the addition of some VISA components into the LabVIEW block diagram to handle the asynchronous communication.

To simplify the task of managing the information contained within each object on the front panel of the LabVIEW interface object oriented techniques are employed on both LabVIEW and the Arduino. For every object placed on the LabVIEW front panel there is an analogous object created in the Arduino sketch. These objects assist in the management of the data contained within each of the front panel LabVIEW objects and allow for the creation of methods that automatically include the packetizing and sending of the information to the opposite party.

### 5.3.2 Object Definitions

All objects within the LabVIEW front panel used in this project fall under two categories: Boolean objects and numerical objects. Boolean objects include switches and LEDs and numerical objects include numerical inputs, sliders, and error message objects. Each object had a specific ID associated with it. This ID is kept constant between LabVIEW and the Arduino and is used to direct the two programs to perform specific actions to the associated objects from case structures when receiving data.

In addition to the ID all objects contain a “state” member variable that contains the value or condition of the object. For Boolean objects the state is either “on” or “off”, here represented by “1” or “0”. For the numeric objects all states are stored as a single precision float. LabVIEW objects contain several more private member variables than their Arduino counterparts. These member variables are the “Arduino resource” and “Arduino error” and contain the information necessary to transmit serial data from LabVIEW to Arduino from the methods of the objects. The methods common to both objects are setting and obtaining the ID and state of the object. The LabVIEW objects contain four extra methods to set and get the “Arduino resource” and “Arduino error”. The “setState” method, used to update the value of the object internally, also contains the serial data transfer procedures.

### 5.3.3 Data Transfer Protocol

A standardized way to pass information over the serial bus asynchronously between LabVIEW and the Arduino is necessary. As previously



described, the LIFA package contains a special VI that allows for the transmission of data to the Arduino where it will wait for a response if necessary. The data sent is first composed into an array (packet) of 15 unsigned char. The first byte of this packet, called the “header byte”, is 255. This is sent as an indication of the start of the packet. The next byte in the packet is the “command byte”. The “command byte” contains either the ID of an object or a different command to be performed. Depending on the type of object that the command is updating either one or four more bytes will be sent. If the object is of the Boolean class the next byte will be a “1” or “0” depending on the state of the object. If the object is of the numerical class the floating point value of the new state of the object will first have to be cast into its 4 byte representation. In this way there is no loss of precision or any estimation when sending floating point numbers. The fifteenth byte is the checksum byte, which is included to ensure that the data transmitted within the packet is correct. The checksum will add together all of the values of the packet but because it is represented by an unsigned char it will only contain the most significant byte. This packetizing system was modified from the LIFA base and is included in the “setState” method within the Arduino objects. Any time the state of the object is changed on either LabVIEW or the Arduino this method is performed and sends the corresponding information to the opposite party.

To receive the information the packet is read and first checked for the “header byte”. Next, the sum of all the bytes within the packet, excluding the checksum, is performed. If both the “header byte” exists and the checksum is

correct the read procedure would continue. The “command byte” is then read and fed into a case structure. This case structure contains all of the commands for the controller. Each case in the structure is associated with a LabVIEW and Arduino object. In most cases the command byte is the ID of the object and the specific procedure associated with that object (i.e. turning the RF power supply on) is contained. Other commands that are not ID specific are updating all object values and initializing the values upon startup. Finally, the data that is sent after the “command byte” is read. If the object is of the Boolean class the value can be taken directly, otherwise the value is recast from the four char array into a floating point number.

## 6 Results and Discussion

### 6.1 LabVIEW Interface

#### 6.1.1 Front Panel

The LabVIEW interface was created in an effort to control the DC voltages applied to the lenses, jet disruptor, exit lens, and the applied RF signal, as well as allowing for visual output of the picoammeter and current measurement statistics. Figure 16 shows the overall front panel layout of the LabVIEW interface. There are five main sections of the front panel: the funnel current measurement section, the DC voltage control section, the RF control section, the polarity control section and the graphical current output.

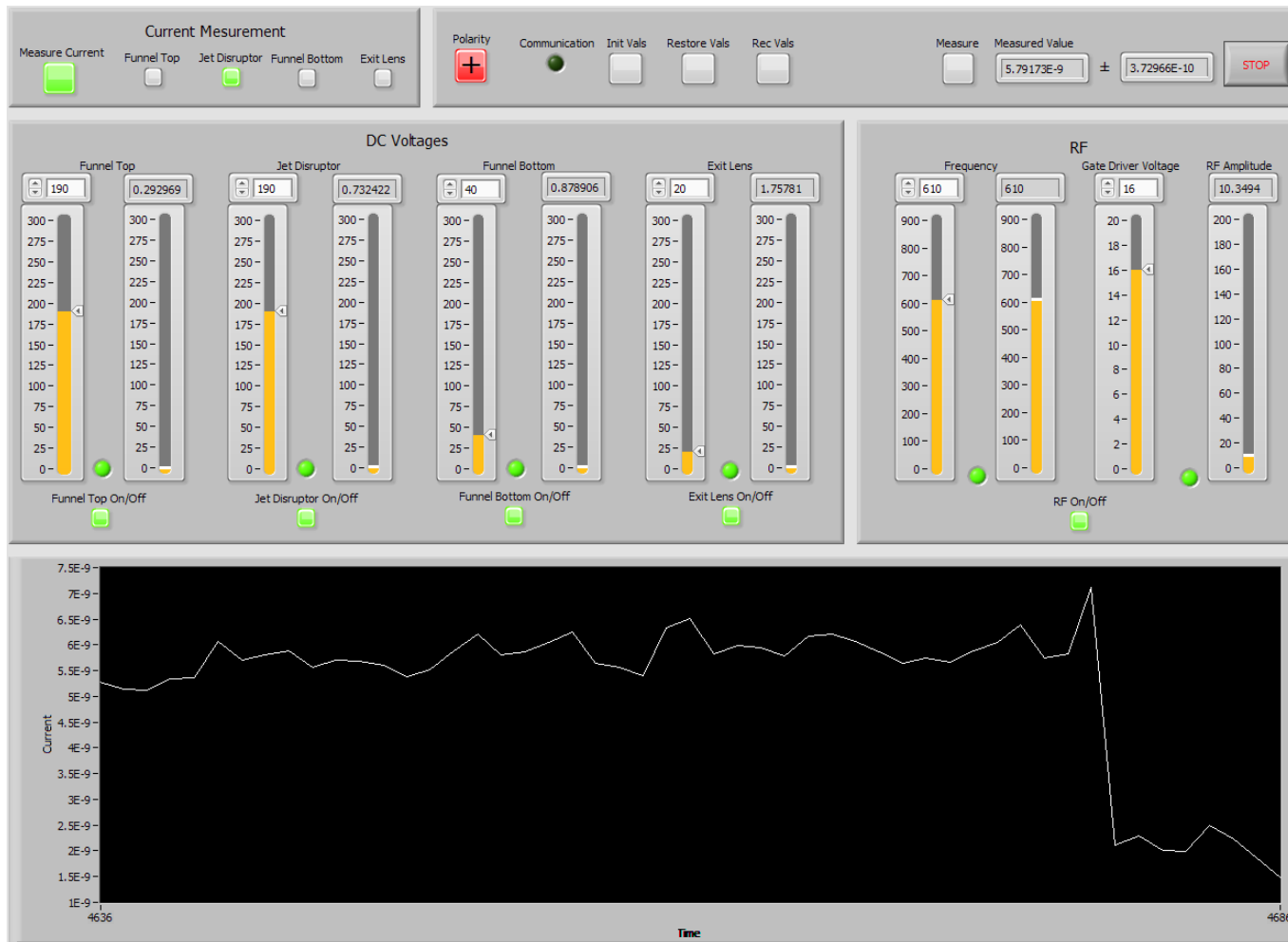


Figure 16: Image of the front panel of the RF ion funnel controller GUI. Controls for all voltages, RF signal and measuring techniques are available to the user.

For the funnel current measurement section a user can control the measurement of the current as it discharges through the lenses of the funnel (the large-right switch). The user then has the option to select any combination of the sections of the funnel on which to measure the current (the four smaller buttons). As soon as the “measure” button is pressed, the relays connected to the selected components will be switched within the controller routing the current discharged on them to a BNC plug attached to the picoammeter. If a switch is not selected the relay connected to that component will be grounded to ensure that there are no electrically floating components where a potential charge buildup and resulting electric field opposing the ions can occur. When the “measure” button is pressed any of the currently applied DC voltages as well as the RF signal are all deactivated as an applied signal interferes with the desired measurement. When the switch is deactivated the RF signal and any voltages that are selected to be on will be reactivated

The section dedicated to adjusting the DC voltages applied to the funnel allows the user full voltage control from the less than 1 V to up to 260 V. Each of the DC voltage supplies has its own control section with on/off switch, adjustment slider, digital input and feedback slider. The RF control section contains slider and digital controls for the frequency (in kHz) and amplitude (in volts applied to the gate driver) of the sinusoidal RF voltage applied to the lenses. The feedback indicators of these sections display the output of a system monitor chip contained in the controller. For the DC voltages the output is scaled using a voltage divider and sent to the monitor chip. A peak detection circuit measures the output of the

RF signal and scales it using a voltage divider. These values are measured once every 0.25s and sent via the serial port from the Arduino to LabVIEW.

The polarity control section allows the user to reverse the polarity of the DC voltages applied to the lenses. This is performed by first turning off any active voltages and then switching relays at each DC to DC converter to the appropriate output polarity pin. Also included in this section are buttons to instruct the Arduino to put all DC voltages and RF signals to the initialization state (all voltages, amplitude and frequency to zero). Although currently unimplemented, the record and restore values buttons will be able to save the current state of the controller and apply these properties when pressed. The measure section in the right of this section is used to take measurements over time. When the “measure” button in this section is pressed an array is created that stores the measured current at every time step until the button is pressed again. Using built in LabVIEW functions the average and standard deviation of this array is calculated and displayed. This function is necessary as the measured electrospray signal often fluctuates around a mean value. The graphical current display section allows a user to view the history of the current measurement over the past 50 seconds. This is especially useful as there is often a large discharge with a long time constant associated with measuring the current on the funnel and stability of the measurement is often not immediately evident.

### 6.1.2 Block Diagram

The LabVIEW block diagram is organized as a flat sequence to ensure that specific tasks are performed sequentially. The first task is to initialize the Arduino and Keithly Picoammeter objects. To initiate these objects one must specify the COM port over which communication takes place. The output of these specific initialization VIs is a resource wire that contains all the information about these objects to be used by the functions of the hardware.

The next section of the sequence instantiates all of the objects that have associated Arduino counterparts. As stated previously object oriented programming techniques were implemented to ease the scalability, data manipulation and transfer of information with both the LabVIEW interface and Arduino program. LabVIEW provides an easy interface to create classes and member functions of these classes within an existing project. Once created these classes can be instantiated as objects and the member functions associated with them used like VIs in a block diagram. To instantiate the objects an object block must first be placed in the block diagram. All of the private variables belonging to the object are contained in the wire coming from the object block. In LabVIEW the instantiated object is the bundled wire or object identifier. This wire can only be unbundled within the methods of the function therefore allowing the data to be manipulated by the methods of the object itself. For each object instantiated an ID must be set corresponding to the specific case in the case structure containing the commands on the Arduino. In addition, Arduino

resource and Arduino error must also be set for each object to allow communication within the methods.

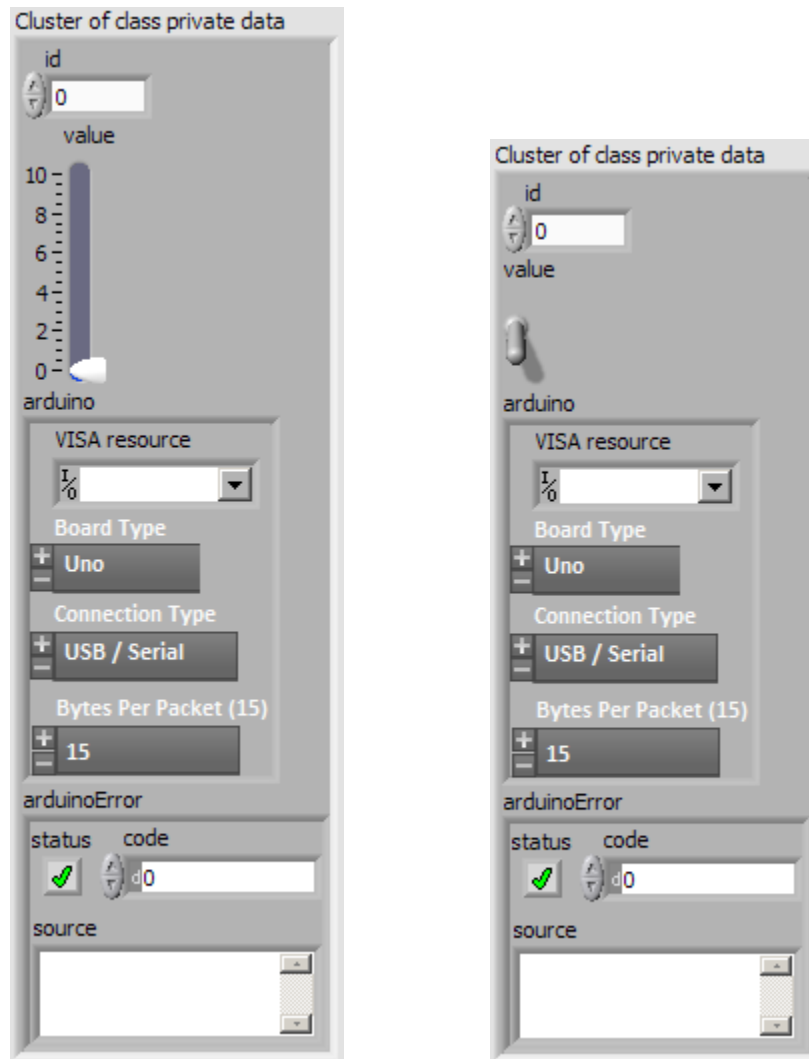


Figure 17: Screen shots of the front panel of the private variable definitions for the numerical LabVIEW class (left) and Boolean LabVIEW class (right).

Figure 17 show how classes are created in LabVIEW. Once the class file has been created a user adds the indicators representing the private data desired to be stored within the class onto the front panel of the class. This part of the class has no block diagram as any processing of this data must be done using methods (as in traditional object oriented programming). LabVIEW



provides an automated method of creating methods to fetch data from and write data to an object.

The final procedure in the flat sequence contains the main loop of the program. There are three sections of the main loop, two of which are contained within a second flat sequence and another that is a function performed at regular intervals. The first portion of this flat sequence ascertains if there is any data on the serial port and analyzes and processes this data if present. The second section of the sequence determines if there has been any change in the front panel controls and if so, updates the value of the associated objects. Outside of this loop there is a function to measure the data from the picoammeter every 1.5 seconds as well as average the data when the “measure” button is pressed.

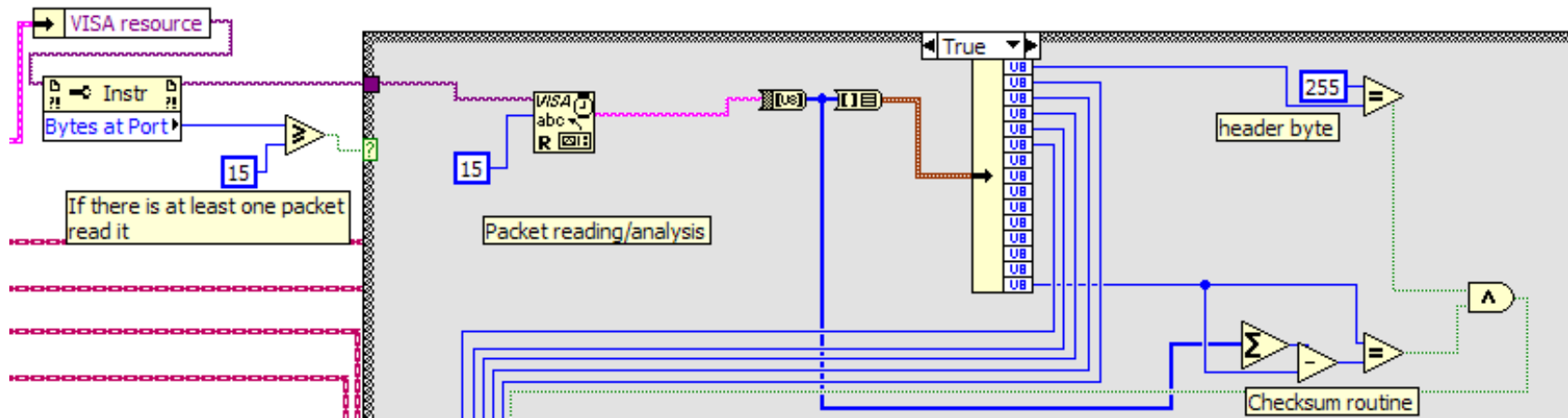


Figure 18: Block diagram used to check for and analyze any data incoming over the serial port from the Arduino to LabVIEW. First, a check is done to see if there is any data at the port. Next, a packet is read. Third, the data is split up into individual bytes. The header byte and checksum byte are then checked. If both are valid the analysis of the packet continues.

Figure 18 shows the block diagram of the function used to check for and analyze incoming data. This process is based on the data transfer protocol described earlier. The first section unbundles the Arduino resource to obtain the VISA resource by itself. Next, there is a check to determine how many bytes are present in the serial port buffer. If this number is equal to or greater than the size of one packet, one packet will be read from the buffer. This data is read as a string and must be converted into a byte array and unclustered to obtain the individual bytes of the array. If the first byte is the header byte (255) and the most significant digit of the sum of all of the values of the array matches the checksum, the communication was properly sent and the analysis of the packet data can proceed.

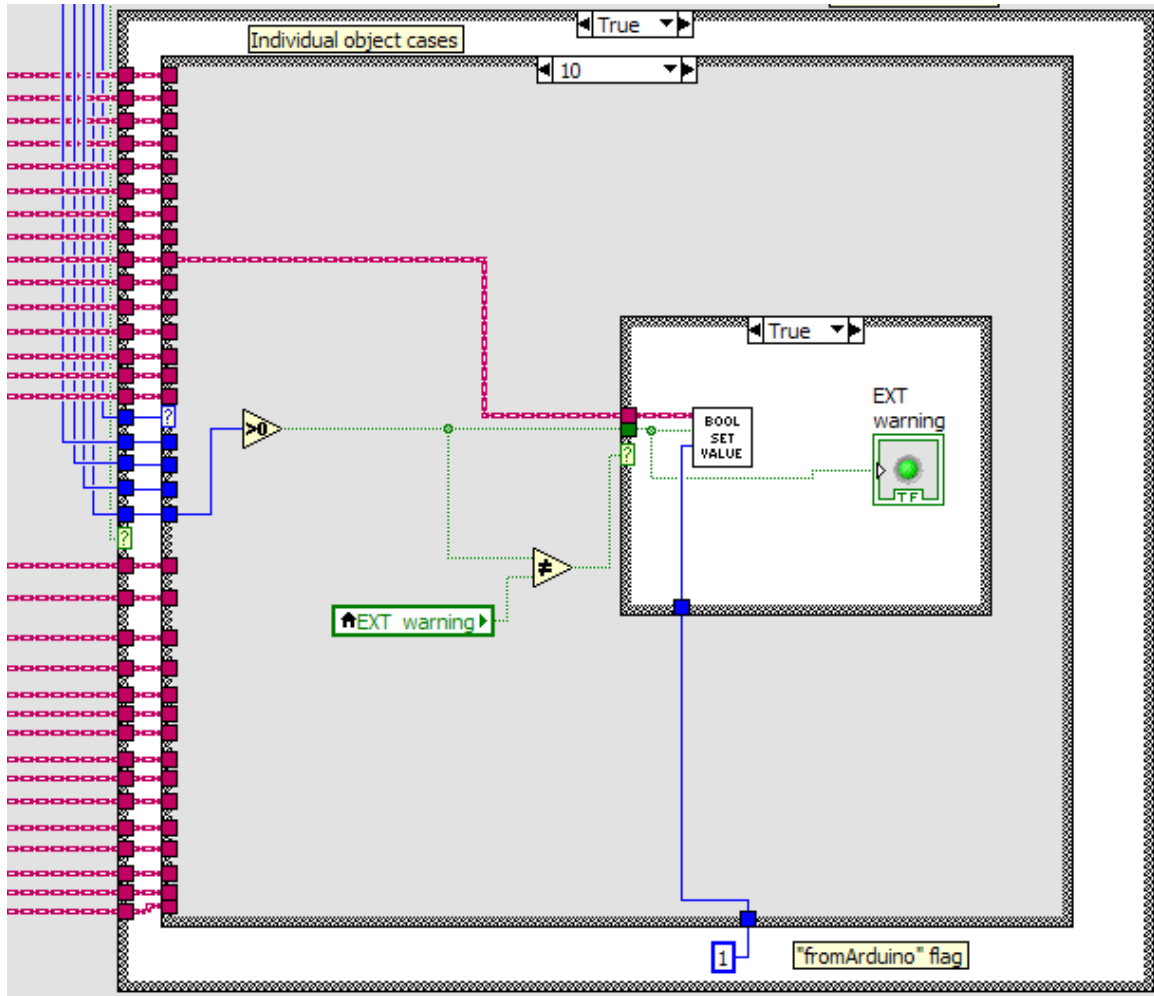


Figure 19: Analysis of a Boolean packet. The first data byte defines which of the case structures will be entered. The data byte is then sent into the appropriate structure and the value is set to the corresponding Boolean object and front panel control/indicator.

Figure 19 shows an example of the case structure used when analyzing the packet data of a Boolean object. The second byte of the packet contains the ID of the object that is being referenced and the third byte contains the binary value of the Boolean object. This binary value is converted to a LabVIEW Boolean type and compared with the current value of the object. If the values are different a new value is set using the “setValue” method of the object and the indicator or control corresponding to the object is also updated. When setting the

value of an indicator the component is updated directly. When setting the value of a control, a local variable must be used to reference the control.

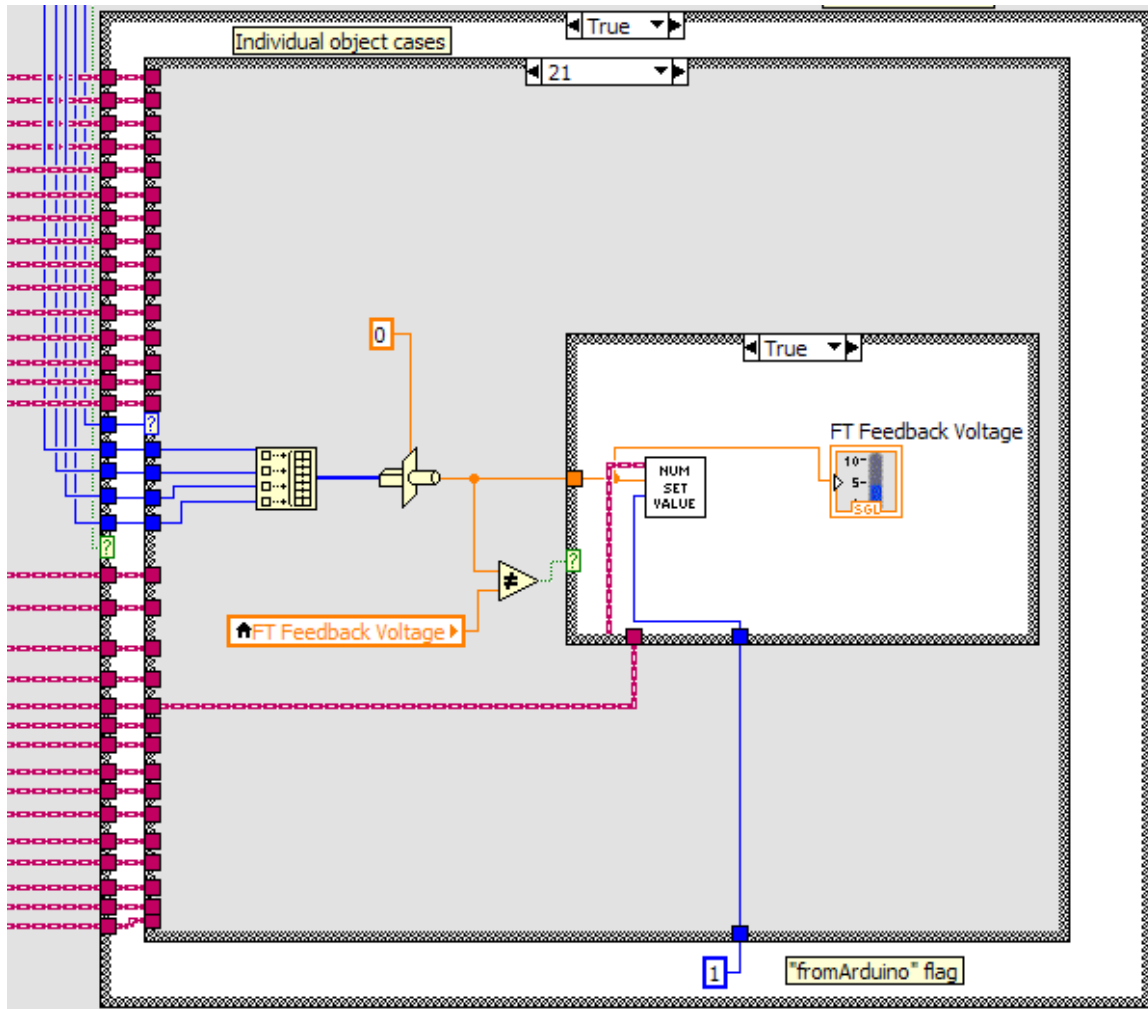


Figure 20: Analysis of a numerical packet. As in the Boolean case the first data byte directs the data to the appropriate case structure. Next, the data is recast from a four byte array into a floating point number and the appropriate numerical object and front panel control or indicator is set.

Figure 20 shows the process for analyzing a packet used to set the value of a numerical object. The main difference from the Boolean object procedure is that there are four bytes sent from the Arduino which correspond to a single precision floating point number. These values must then be converted using the

type case function. When setting values sent originally from the Arduino, the “fromArduino” flag must be 1 to prevent a transmission back to the Arduino. Each case has the object identifier wire corresponding to the object to be updated being routed into the appropriate case of the case structure which is then connected to the “setValue” function.

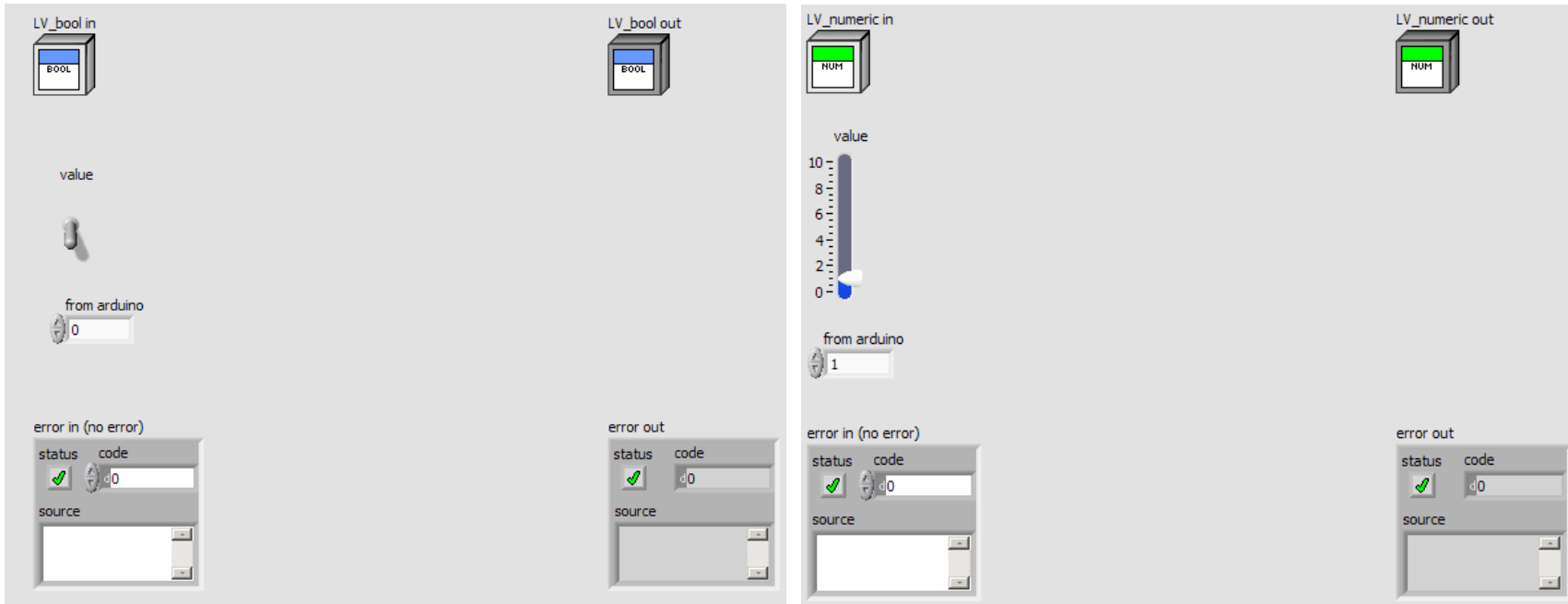


Figure 21: Front panel images of Boolean (left) and numeric (right) setValue methods.

Figure 21 shows what the front panel of a member function used to set the value of an object created automatically using LabVIEW looks like. Normally this method only updates the private variable of the object but because of the desired transmission to the Arduino the operations to perform this task were also included. In this way when an object is updated in LabVIEW it not only stores the new private variable but also transmits the value of this variable to the Arduino.



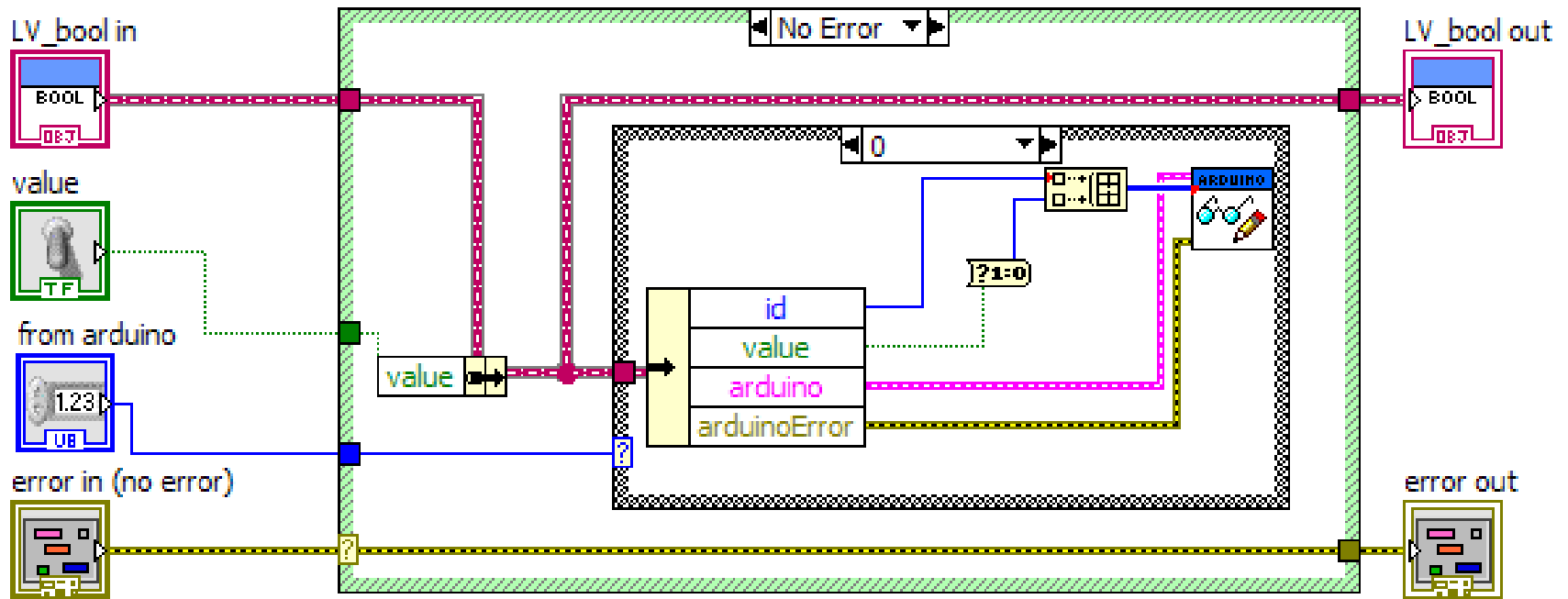


Figure 22 Block diagram images of Boolean (top) and numeric (bottom) setValue functions. Notice the functionality included to automatically send the value of the object to the Arduino.

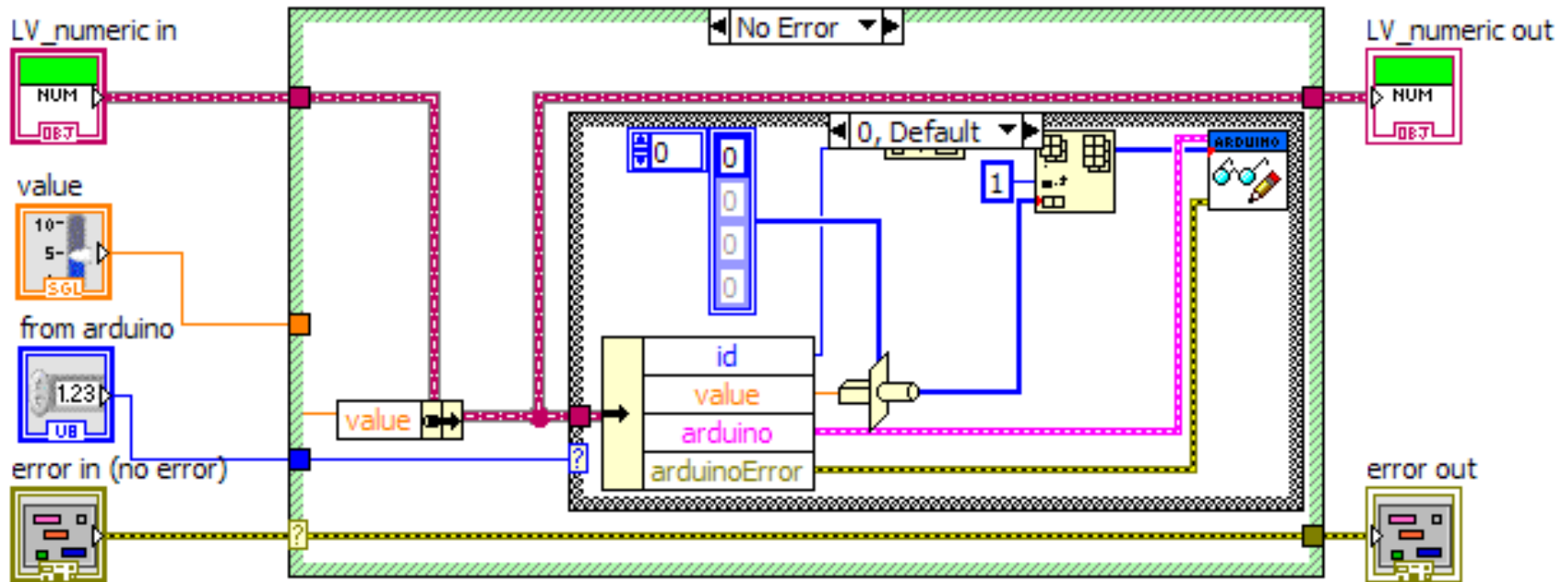


Figure 22 (Continued)

Figure shows the “setValue” function with transfer operations included. As stated previously these functions can be used like VIs on the block diagram of the program. The modules on the left are the inputs for the “setValue” method and the modules on the right are the outputs. The user must supply the object identifier, which identifies the instantiated object for which the variables will be modified, the value, the “fromArduino” flag, and the object error. The “fromArduino” flag identifies whether the value was set from the Arduino (flag set to 1) or from LabVIEW (flag set to 0). Because a corresponding “setValue” function exists on the Arduino this flag ensures that an infinite loop of “setValue” transmissions does not occur as the values are set from LabVIEW to the Arduino. The outputs of the method are the object identifier and the object error handler.

The process of setting the object value first requires using the bundle by name LabVIEW function. This allows for a bundled wire to be broken down into its individual components where the value of those components can be modified and rebundled. This bundle is then routed to the output object identifier of the method. To perform the transmission to the Arduino the object identifier bundle is routed in parallel to an unbundle by name LabVIEW function. Unlike the bundle by name function which can take an existing bundle, break it into its components and output the same bundle, the unbundle by name function solely breaks up the bundle into its constituents.

Using this function the object’s ID, value, Arduino resource and Arduino error variables can be obtained. With the Arduino resource (which identifies the

serial port information about the Arduino) and Arduino error wires the program can now transmit the new values using the LIFA supplied “read/write” module. This module takes an array containing the command number and relevant data for that command, packetizes the data and sends it to the Arduino. The ID of the object is used to identify which command to perform on the Arduino end of the communication within a case structure. For Boolean transmission the true/false value is converted to a binary one or zero and a two byte data array is created. The single precision floating point numbers must first be type cast into a four byte array. A five byte array is created for these transmissions.

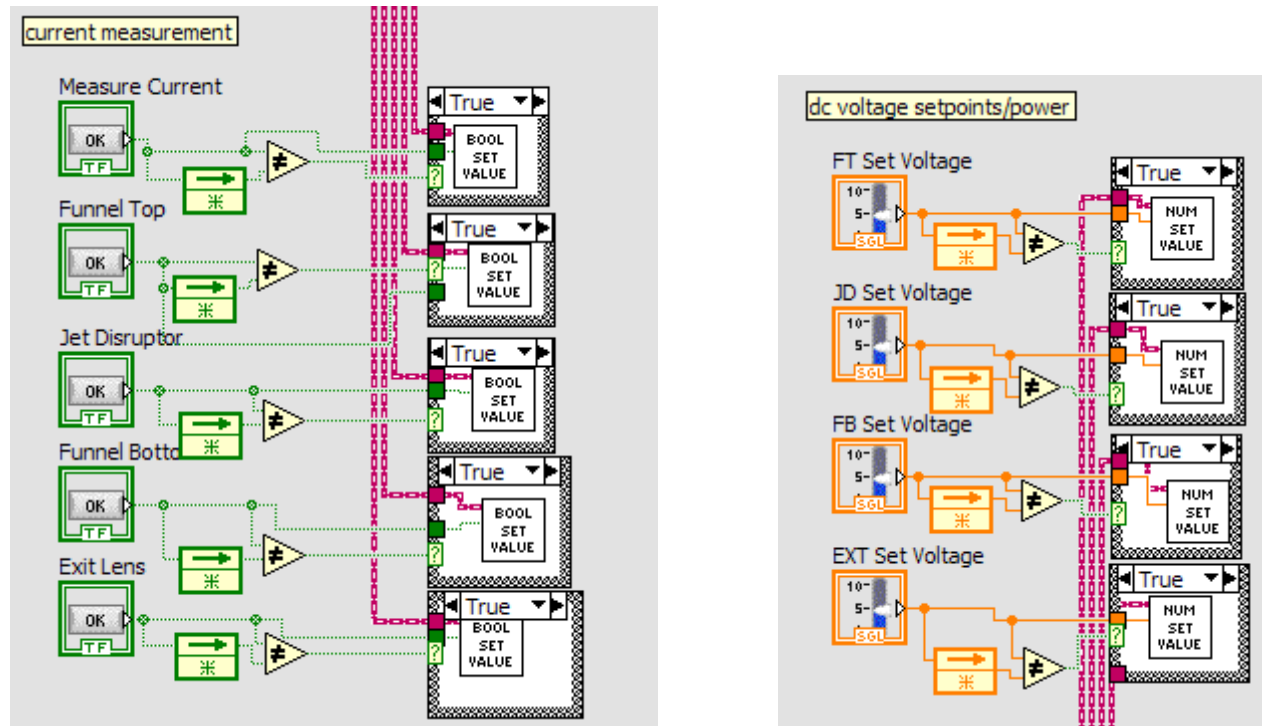


Figure 22: Block diagram images of the GUI front panel control sections used to update their associated objects for the Boolean (left) and numeric (right) objects.

After the check is performed to determine whether there are any packets to be analyzed at the serial buffer, the main loop can proceed and update values from controls on the front panel if any changes were made. Figure 22 shows the block diagram code to check if any changes were made to the controls and if so set the corresponding value to the object (which will simultaneously send the data to the Arduino). A feedback node (represented by the arrow block) is used to compare the previous value of the control with the current one. If this value is different the “setValue” method will be performed with this updated value.

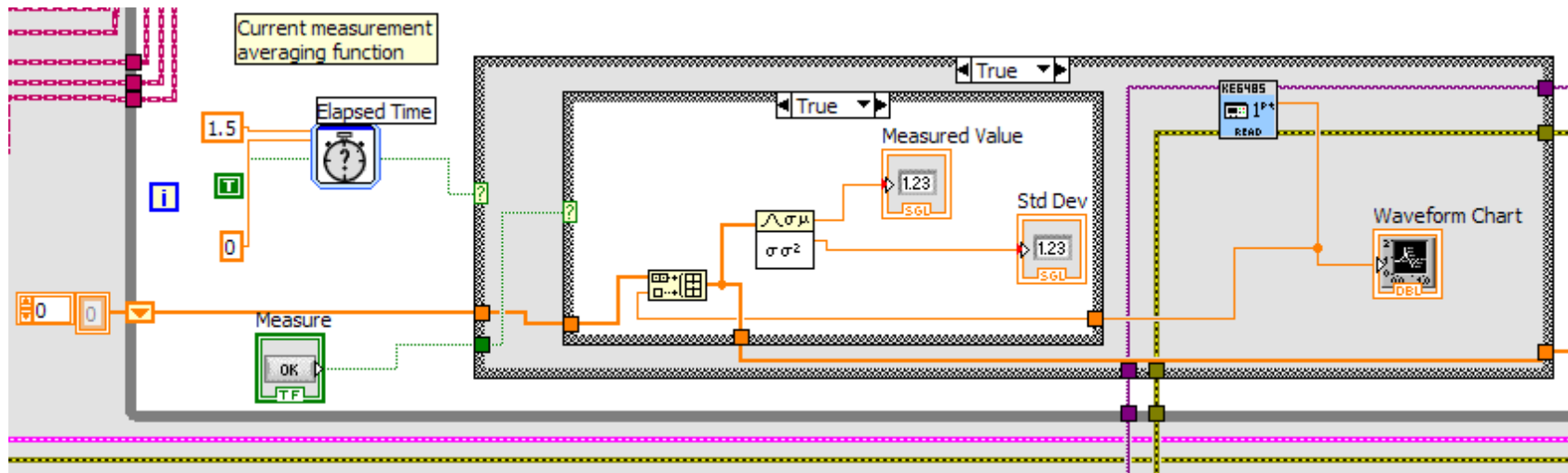


Figure 23: Block diagram image of the current time averaging and time history display function.

Figure 23 shows the function used to average the current measurements as well as display the ion current history on the graphical current display. A timer function outputting a “true” value when the specified time has been reached controls when the measurement will take place. Using the VI supplied by the Keithley Picoammeter the current measurement can be obtained. The “measure” button controls whether ion current data will be added to an array which is then averaged and the standard deviation calculated. Using shift registers this array is passed along for every iteration of the loop allowing for current data to be measured over time. These calculated values are then displayed on an indicator on the front panel. The final steps of the program are to close the connections to the Arduino and the Keithley Picoammeter. When the “stop” button is pressed the main loop is aborted and these procedures are performed.

## **6.2 Electropray Current Measurements**

The main purpose of the electropray experiments is to determine where the losses of ions occur during operation. One of the main concerns when focusing the electropray plume is the separation of the usable gas phase ions from neutral residues and ions contained in solvent droplets. The focus of these experiments is to vary the parameters that change the ratio of gas phase ions to the unusable spray components. Cloupeau, et. al [6] systematically varied many of these parameters and indicated their effect on the droplet size. By measuring the total current into the funnel in comparison with the current on the jet disruptor alone conclusions can be drawn about the amount of material being lost by impact with the jet disruptor. The transmission of the useable current (i.e. the



ions making it around the JD) through the funnel can be quantified by dividing the transmitted current by the total current into the funnel minus the jet disruptor current.

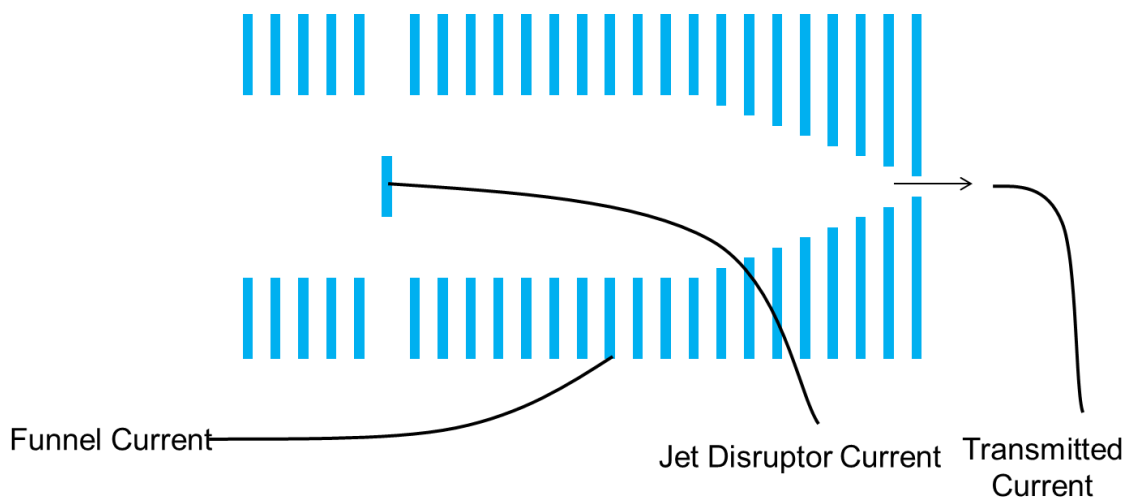


Figure 24: Schematic indicating the measurement points on the funnel electrodes.

### 6.2.1 Current versus Needle to Capillary Distance

Several experiments were performed to determine the effect of the needle to capillary distance on the various currents and ratios of those currents. The flow rate was kept at a constant  $1.667 \mu\text{L}/\text{min}$ . This value was previously established as the minimum stable flow rate at which cone-jet mode electro-spray could be performed. The initial spray was performed at 1 mm and the distance was increased by 1 mm for each successive measurement up to 7 mm. The voltage applied to the capillary was +210 V and the voltage applied to the needle was varied from +1800 V to +2600 V as the distance was increased. The heater was kept at a constant  $130 \text{ }^\circ\text{C}$ . Current measurements were taken over all of the lenses of the funnel, the jet disruptor, the exit lens and on the jet disruptor alone.

The voltages applied to the funnel were +200 V for FT, +190 V for JD, +40 for FB, and +20 for EXT. The RF frequency was 610 kHz and the amplitude was 125 V<sub>peak-to-peak</sub>. The solution used was the 0.5 mg/ml Cytochrome-C solution described previously. The DC voltages applied to the funnel, the values for the RF frequency and amplitude and the solution were the same for all the successive experiments described in this section. The capillary IDs used were 30 and 20 mil and both were 6.4 cm long. The measurements were repeated three times for each capillary ID.

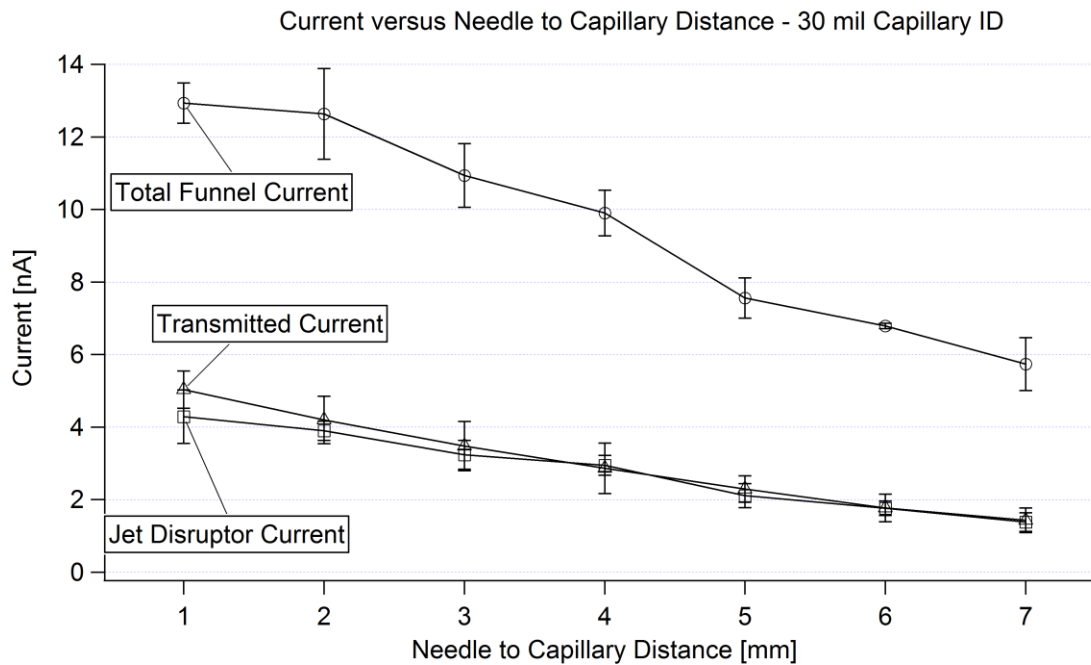
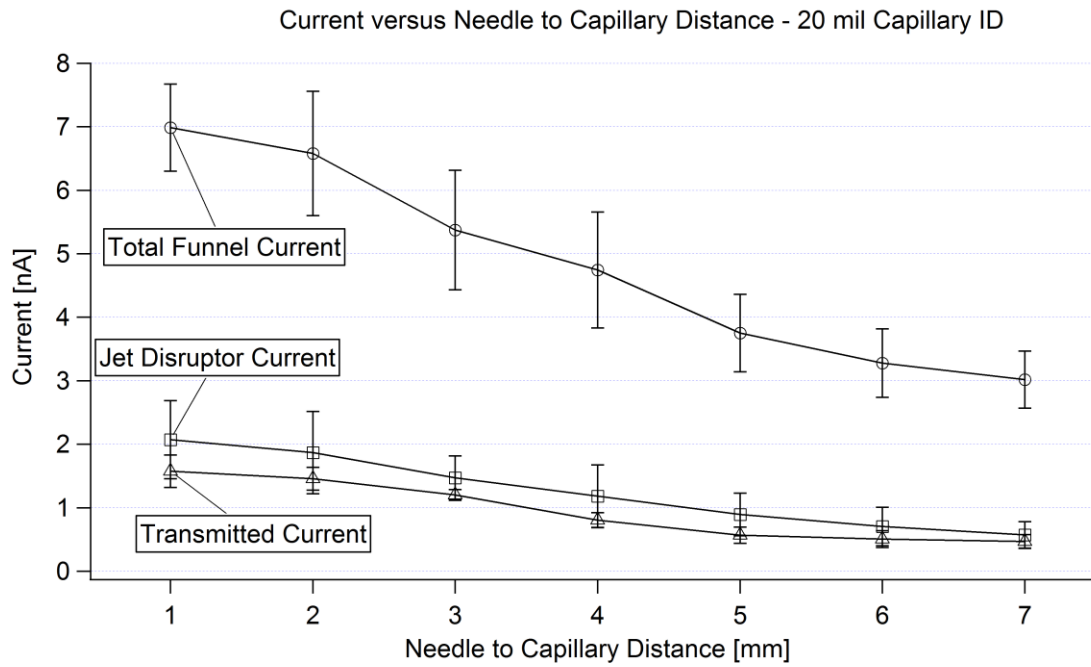


Figure 25: Current measurements versus needle to capillary distance variation of 20 mil (top) and 30 mil (bottom). There is a clear downward trend as the distance between the two is increased. This is mainly due to an expansion of the plume away from the needle while sampling from a constant size inlet.

Both of the graphs for these experiments show similar negative correlation between the needle to capillary distance and measured currents on and through

the funnel. The main reason for this is the expansion of the electrospray plume. As demonstrated in Figure 4 there is a large expansion of the droplets after the tip of the Taylor cone. This is due to the mutual repulsion of the similarly charged droplets emitted during electrospray. When the ESI emitter tip is located very close to the inlet capillary the viscous forces of the airflow into the vacuum chamber dominate and nearly the entire plume is collected into the capillary. Beginning at a distance of around 3 mm the entire plume ceases to be drawn into the capillary and much of the material escapes or collides with the edge of the capillary. The result is that a much smaller portion of the spray is being sampled therefore resulting in a much lower current to be measured.

The transmission efficiencies through both capillaries also decrease with increasing distance. This can be interpreted as the spray entering the chamber having a lower quality (i.e. contains more and/or larger un-transmittable droplets). This could be explained by the larger droplets size as they have more mass and are less affected by the repulsive space charge effects of their smaller counterparts. As a result, these droplets will tend to travel along the centerline for a longer distance due to their higher momentum and therefore are more likely to be sampled by the capillary into the chamber. As the emitter is positioned farther away the space charge effect has more time to act on the droplets, and more of the larger droplets will be sampled. It should be noted that the first data point of the graphs shown correspond to the “standard” operating conditions of the ESI patterning system.

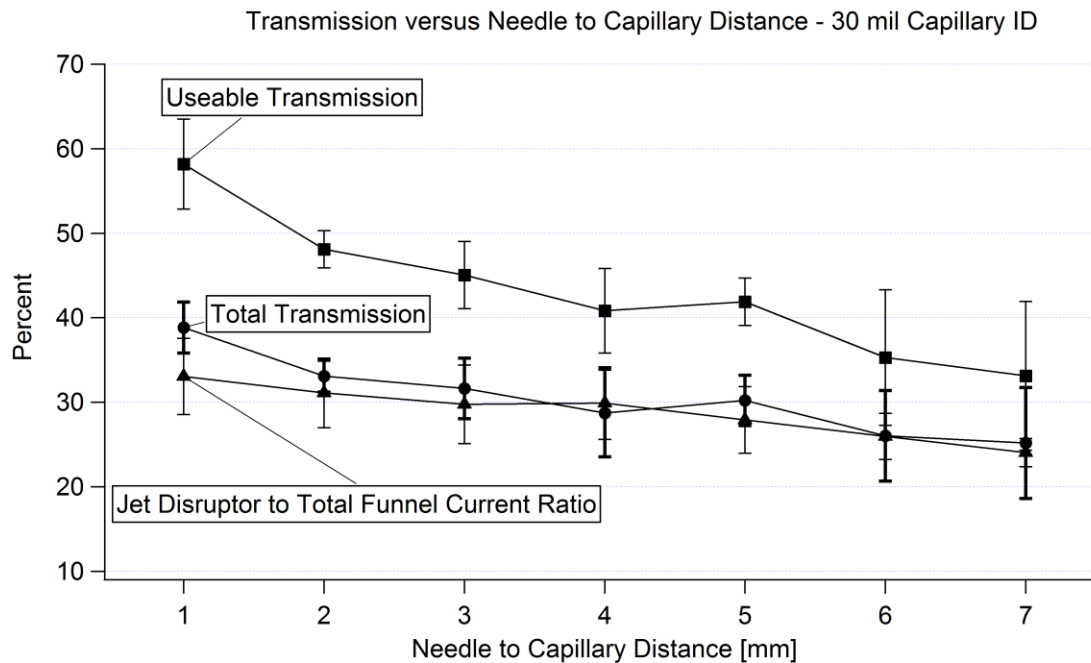
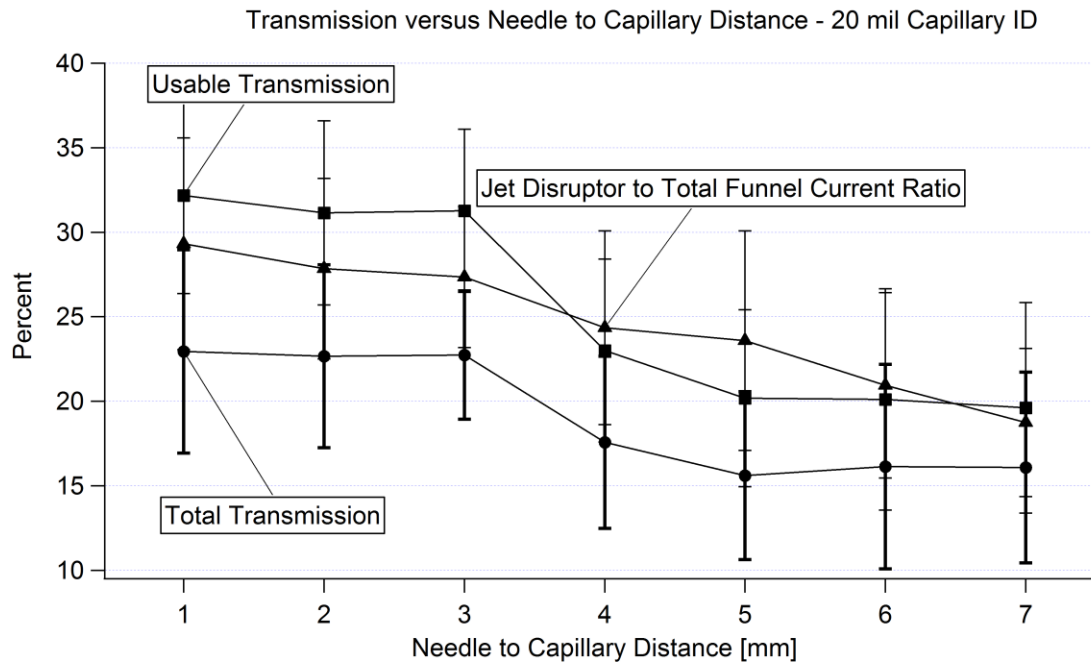


Figure 26: Funnel transmission values versus needle to capillary distance of 20 mil capillary (top) and 30 mil capillary (bottom). There is a downward trend as the needle to capillary distance is increased. This is most likely due to the fact that higher momentum droplets are preferentially sampled by the capillary over smaller, focusable droplets resulting in a poorer quality spray.

### 6.2.2 Current versus Heater Temperature

The heater serves to assist in removing the solvent from injected droplets. Increasing this temperature is thought to accelerate the rate of desolvation as the spray passes through the capillary into the funnel chamber thereby decreasing the droplet diameter and increasing the likelihood of coulomb fission of gas phase ions. For these experiments the heater was first started at room temperature (21 °C), increased to 40 °C and then increased by 20 °C for each successive step until reaching a final temperature of 180 °C. The method for measuring the currents was the same as in the needle to capillary distance experiments. The needle was positioned 1 mm from the capillary inlet and a needle voltage of +1800 V was applied. A solution flow rate of 1.667  $\mu\text{L}/\text{min}$  was used. These experiments were performed three times for the 20 and 30 mil ID capillaries.

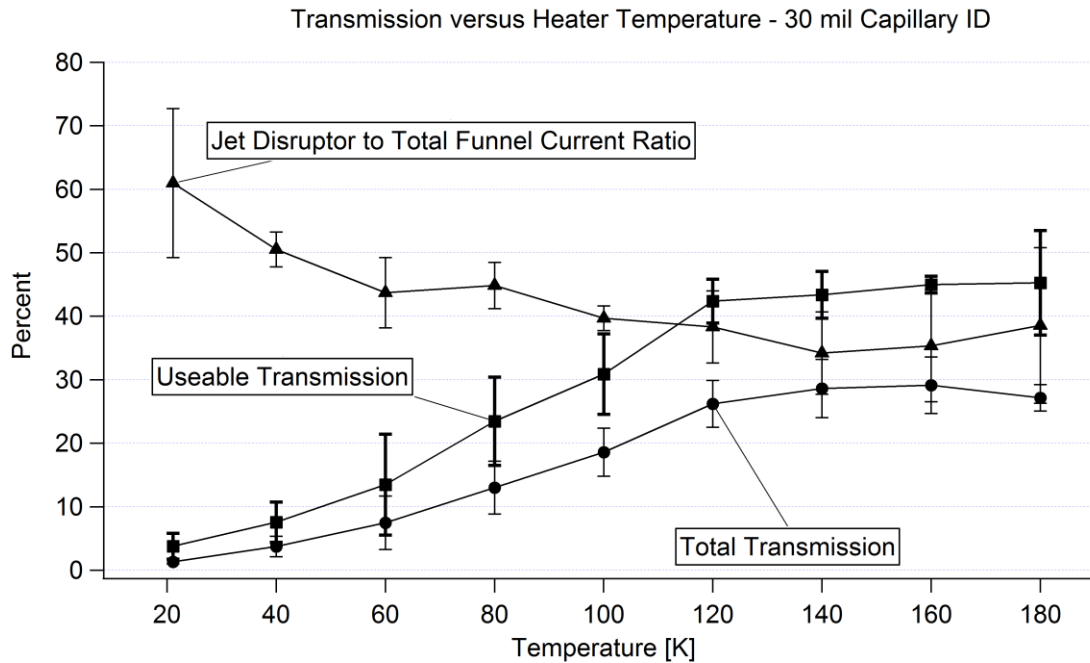
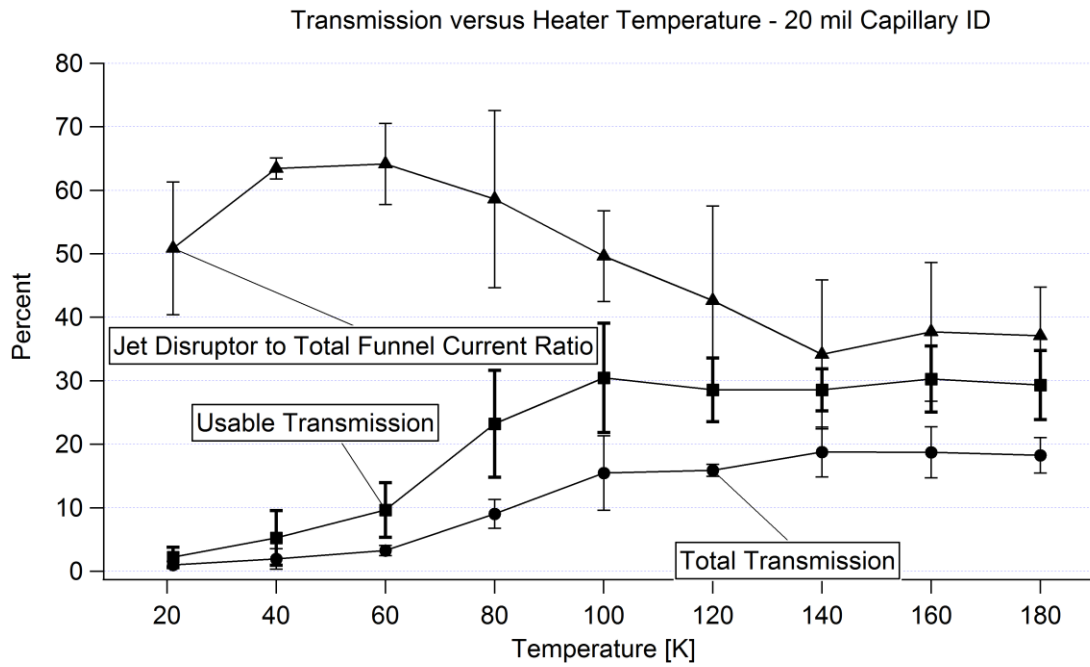


Figure 27: Funnel transmission values versus heater temperature variation of 20 mil capillary (top) and 30 mil capillary (bottom). As the heater temperature increases the jet disruptor to total funnel current decreases and the transmitted currents increase. This is due to an increase in the desolvation of the droplets and therefore a higher focusable to unfocusable spray ratio.

The findings indicate that the two capillaries show similar trends as the temperature of the heater is increased. When the temperature is very low the

evaporation of the solvent of the droplets becomes minimal. This results in a very “wet” spray and it is hypothesized that many of the losses occur due to condensation within the capillary. The droplets that do come through are larger and therefore have greater momentum, which drives them into the jet disruptor as is evident from the high jet disruptor to total funnel current ratio. Very little of the current is transmitted through the funnel and it is believed that this is due to the smaller effect the electric field has on the large droplets.

As the temperature increases the quality of the spray improves. The jet disruptor to total funnel current ratio begins to decrease while at the same time the transmitted currents increase. The temperature increases the rate of evaporation of the droplets, allowing them to reach the Rayleigh limit at a faster rate and therefore produce more gas phase ions. As the temperature is increased to approximately 140 °C and beyond the transmissions as well as the ratio of jet disruptor current to total funnel current reaches a plateau. As shown in Figure 28 the currents entering the funnel continue to increase with increasing temperature. These data suggest that the rate of desolvation reaches a limiting value preventing any increase in spray quality. Page et. al. observed a similar peak in transmitted current occurring around 140 °C [30].



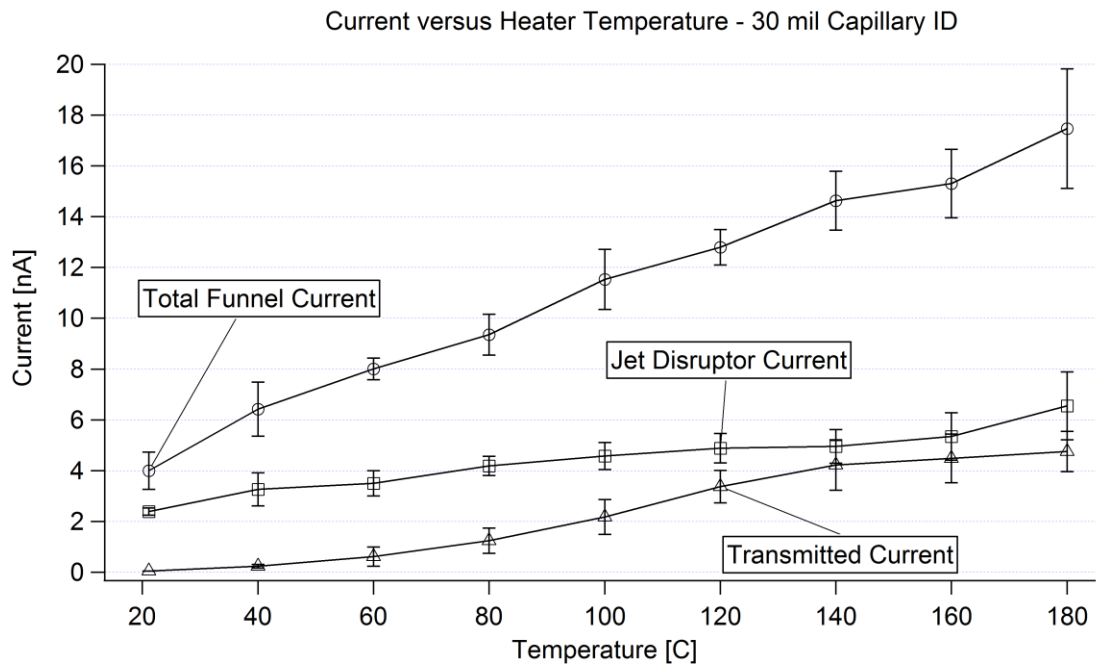
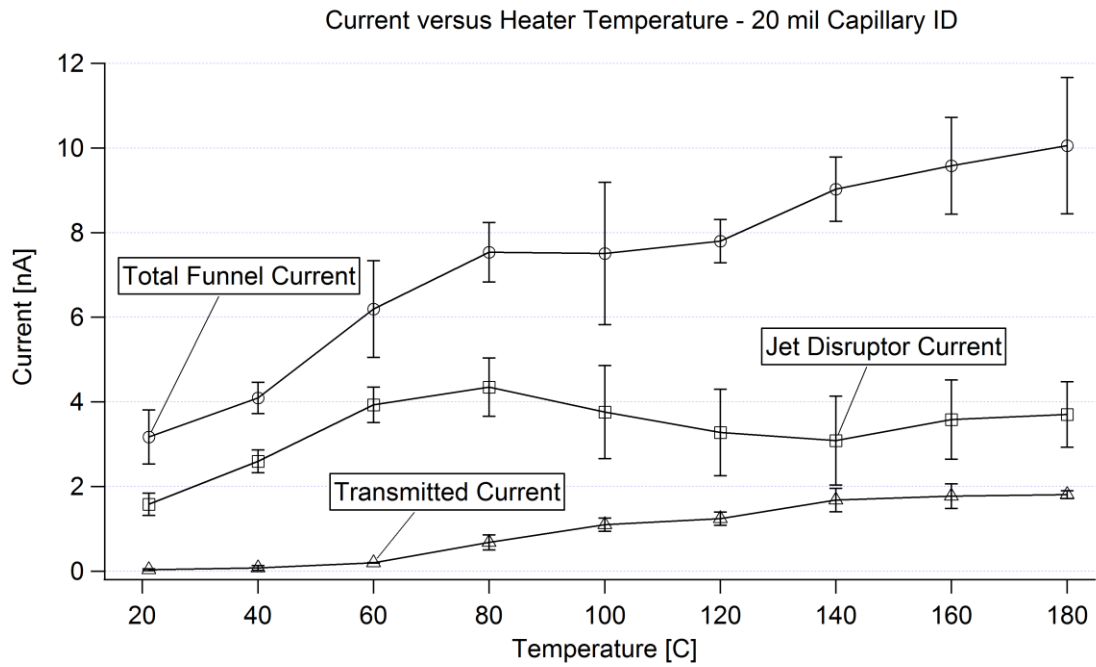


Figure 28: Current measurements versus heater temperature variation of 20 mil capillary (top) and 30 mil capillary (bottom). As the temperature is increased the current entering the chamber also increases. Plateau values are reached around 140 °C which is in agreement with the literature.

### 6.2.3 Current versus Solution Flow Rate

Experiments were performed by increasing the flow rate of the solution while keeping the other parameters constant. Since the standard flow rate used is the minimum stable value the first point in this set of experiments was 2  $\mu\text{L}/\text{min}$ . This was increased by 1  $\mu\text{L}/\text{min}$  until reaching a final solution flow rate of 8  $\mu\text{L}/\text{min}$ . The heater temperature was kept at 130  $^{\circ}\text{C}$ , the needle to capillary distance was 1mm, and the applied needle voltage was +1800 V. Current measurements were taken in the same way as the previous two sets of experiments and three sets of experiments were conducted for each capillary ID.

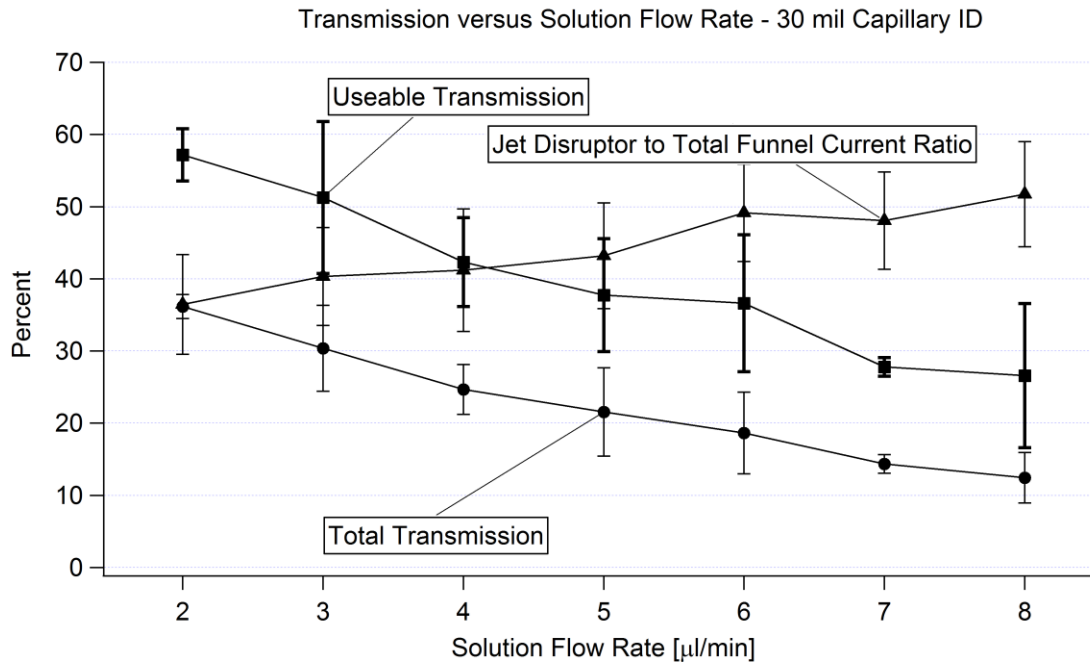
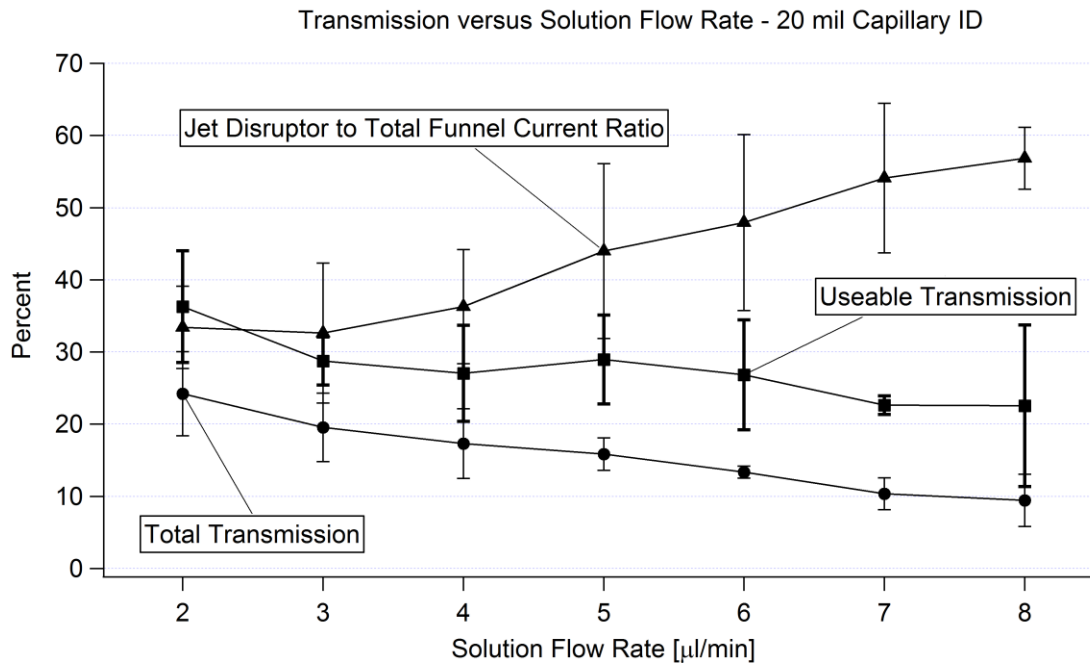


Figure 29: Transmission values versus the solution flow rate variation of 20 mil capillary (top) and 30 mil capillary (bottom). Increased flow rate results in a larger droplet production thereby increasing the ratio of current in the jet disruptor to the total funnel current and decreasing the transmission of the funnel.

As it was found in the previously described experiments, the two capillary sizes yielded similar trends with increasing solution flow rate. Increasing values

of flow rate yielded opposite trends to that of the heater temperature experiments. This finding supports the conclusions drawn from both experiments that larger droplets will yield a poorer quality spray and therefore result in losses on the jet disruptor. Cloupeau demonstrated that the emitted droplet size is proportional to the jet diameter issuing from the tip of the Taylor cone [6]. In Figure 30 from that study it can also be seen that with increasing flow rate the corresponding jet diameter also increases and consequently the subsequent emitted droplet size, as well.

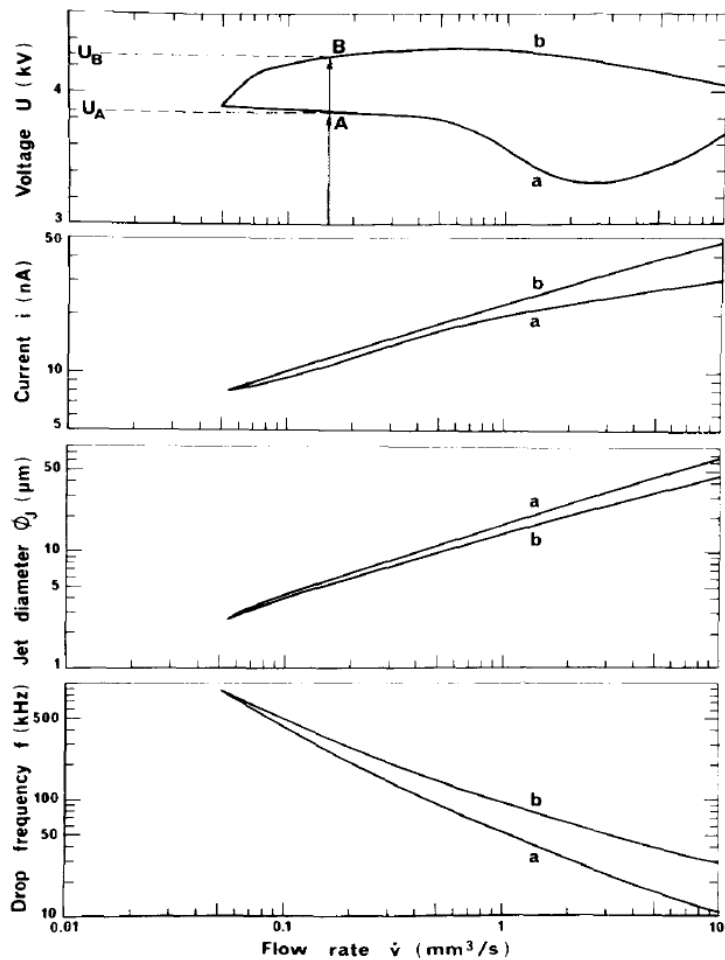


Figure 30: Results from Cloupeau et. al. showing how the variation of solution flow rate effects electro-spray properties.

#### **6.2.4 Funnel Transmission versus Jet Disruptor Voltage**

These experiments were performed in order to determine the effect of the jet disruptor voltage on the transmission of the funnel. Previous testing had determined that the transmission values were very sensitive to this voltage and that there existed a maximum value. Three experiments were performed by varying the voltage applied to the jet disruptor and measuring the ratio of the transmitted current to the total current entering the chamber. The average of three experiments was taken. The standard flow rate, heater temperature, needle to capillary distance and funnel voltages (aside from the jet disruptor) were used.

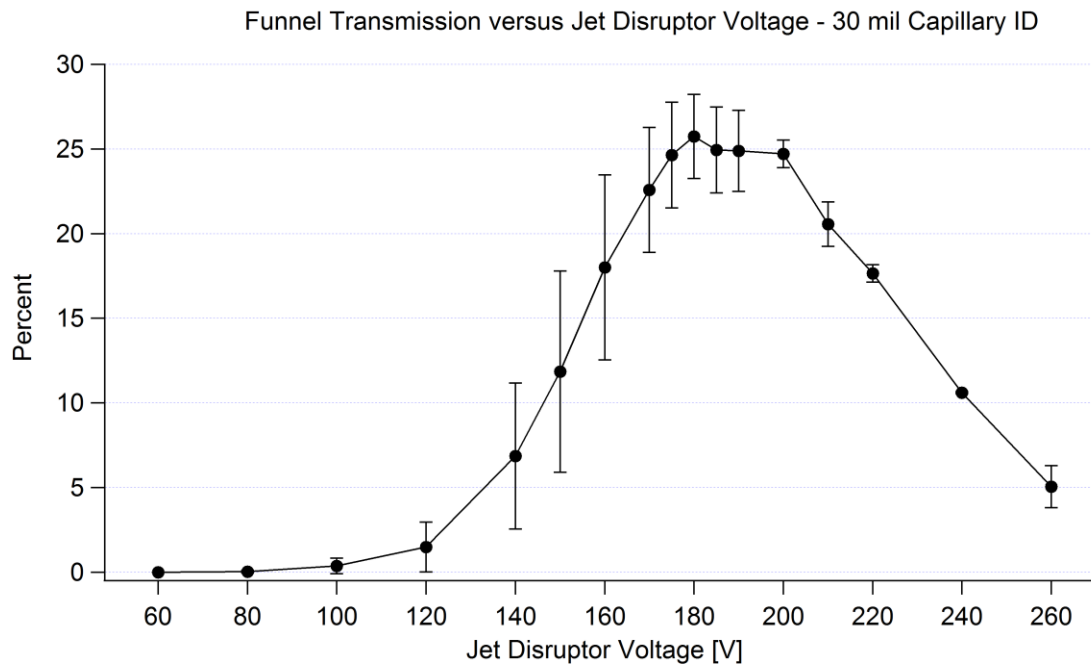
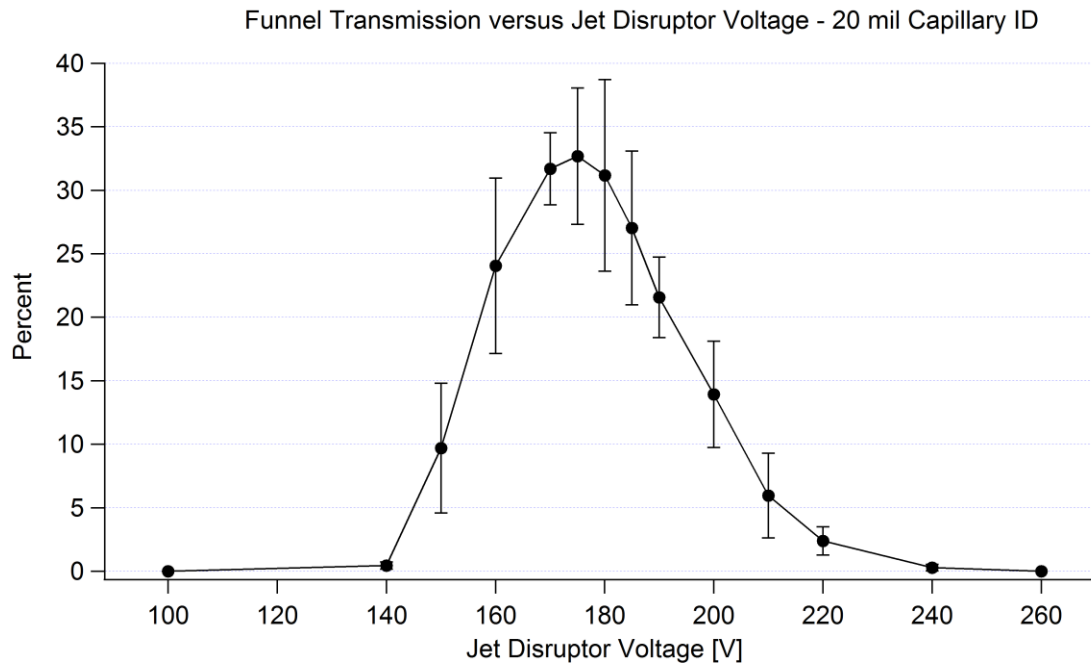


Figure 31: Total funnel transmission versus jet disruptor voltage. These images indicate that there is an optimum voltage which the jet disruptor should be set to. For the 20 mil capillary (top) the voltage is around 175 V for the 30 mil capillary (bottom) the voltage is 180 V.

Figure 31 shows the results of these experiments. The transmission peaks at 175 V and 180 V for the 20 and 30 mil capillaries respectively. Both graphs show similar trends with a sharp onset of transmission before the peak and a linear tapering off afterward. The 20 mil capillary results indicate that the use of this capillary is more sensitive to the voltage applied to the jet disruptor. There is a higher onset and lower maximum voltage that will give adequate transmission through the funnel. In contrast, the 30 mil capillary data demonstrates that the onset of transmission begins much earlier (albeit very poorly). The maximum jet disruptor voltage could not be determined due to the maximum voltage limitations of the funnel controller.

The shape of the curve is the result of the electric field created between the exit of the capillary and the jet disruptor. When the jet disruptor voltage is low a strong attractive electric field exists between the capillary and the jet disruptor. The ions are therefore drawn toward the jet disruptor and this force dominates the fluid flow force and gradient of the funnel. At very high jet disruptor voltages a strong repulsive force develops pushing the ions away from it. The peak exists at the optimum point where the repulsive force prevents the ions from contacting the jet disruptor but is weak enough for the fluid flow, RF field and voltage gradient to still have a positive effect. The difference in the widths of the peaks may be attributed to the higher pressure resulting in more ion-gas collisions when using the 30 mil capillary. The fluid flow has a more dominant effect resulting in less influence from the jet disruptor.

## 6.3 Fluid Flow Simulation Results

### 6.3.1 Flow Field Properties

Fluid flow simulations were performed using both the 20 and 30 mil capillary IDs. The standard temperature of 130 °C was applied to the wall of the capillary. Measured pressures from within the funnel chamber, when using the different capillary sizes, were applied as the boundary conditions in between the lenses. The pressure of the collision cell chamber was applied as the boundary condition to the outlet after the last lens.

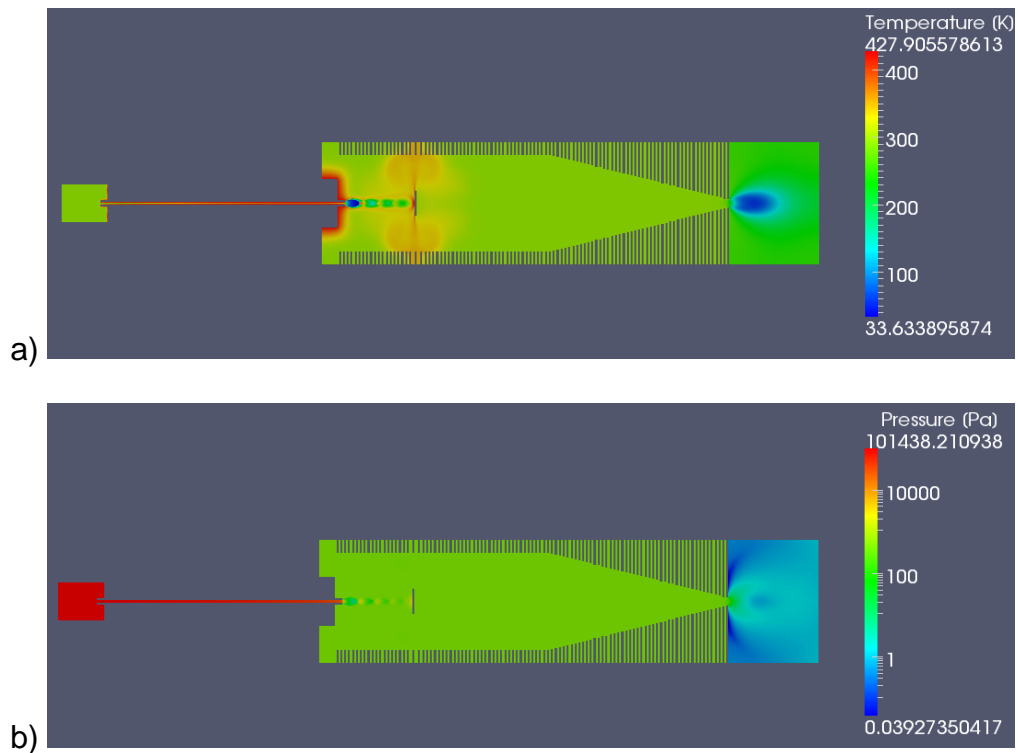


Figure 32: Fluid flow field results. The temperature, pressure and velocity fields of the 20 mil capillary (a, b, c) and 30 mil capillary (d, e, f) are shown.



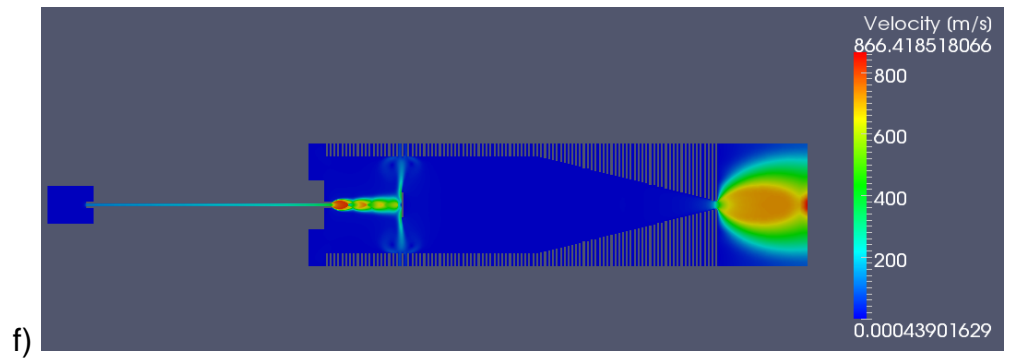
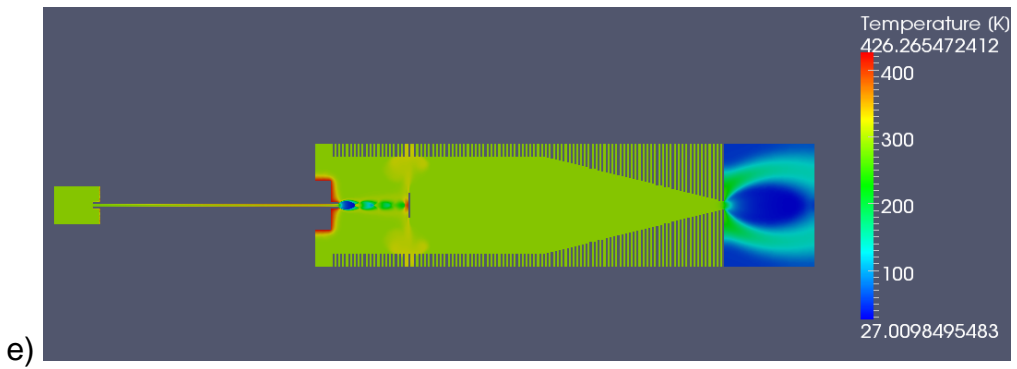
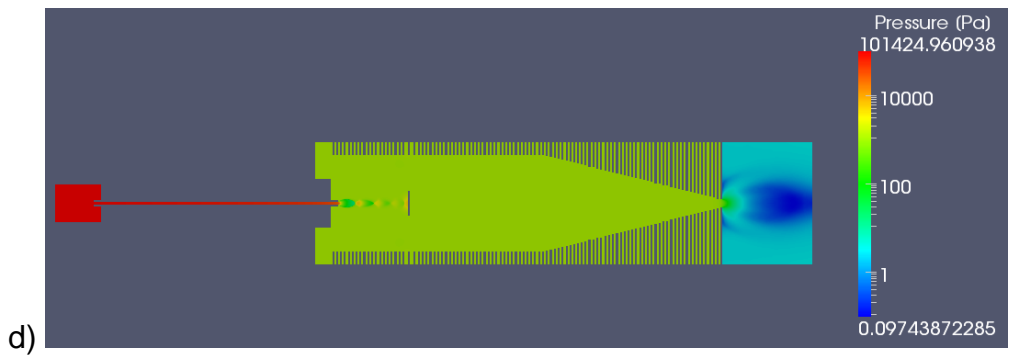
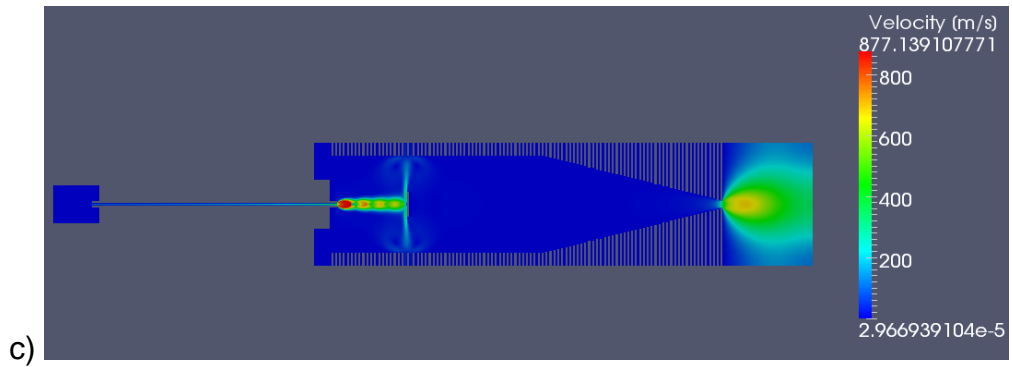


Figure 33 (Continued)

Figure 32 shows the pressure, temperature, and velocity magnitude distributions over the entire domain of the 20 and 30 mil simulations. Both simulations yield similar results; an underexpanded jet emitting from the exit of the capillary into the chamber which impinges on the jet disruptor. After impinging the jet breaks down and yields two large circular vortices which rotate opposite each other around the jet disruptor. The main notable difference between the two simulations is the resulting jet after the last lens of the funnel. Due to the different pressures of the two chambers (456 Pa and 1172 Pa for the 20 and 30 mil capillaries respectively) a more underexpanded jet will emit from the 30 mil capillary simulation.

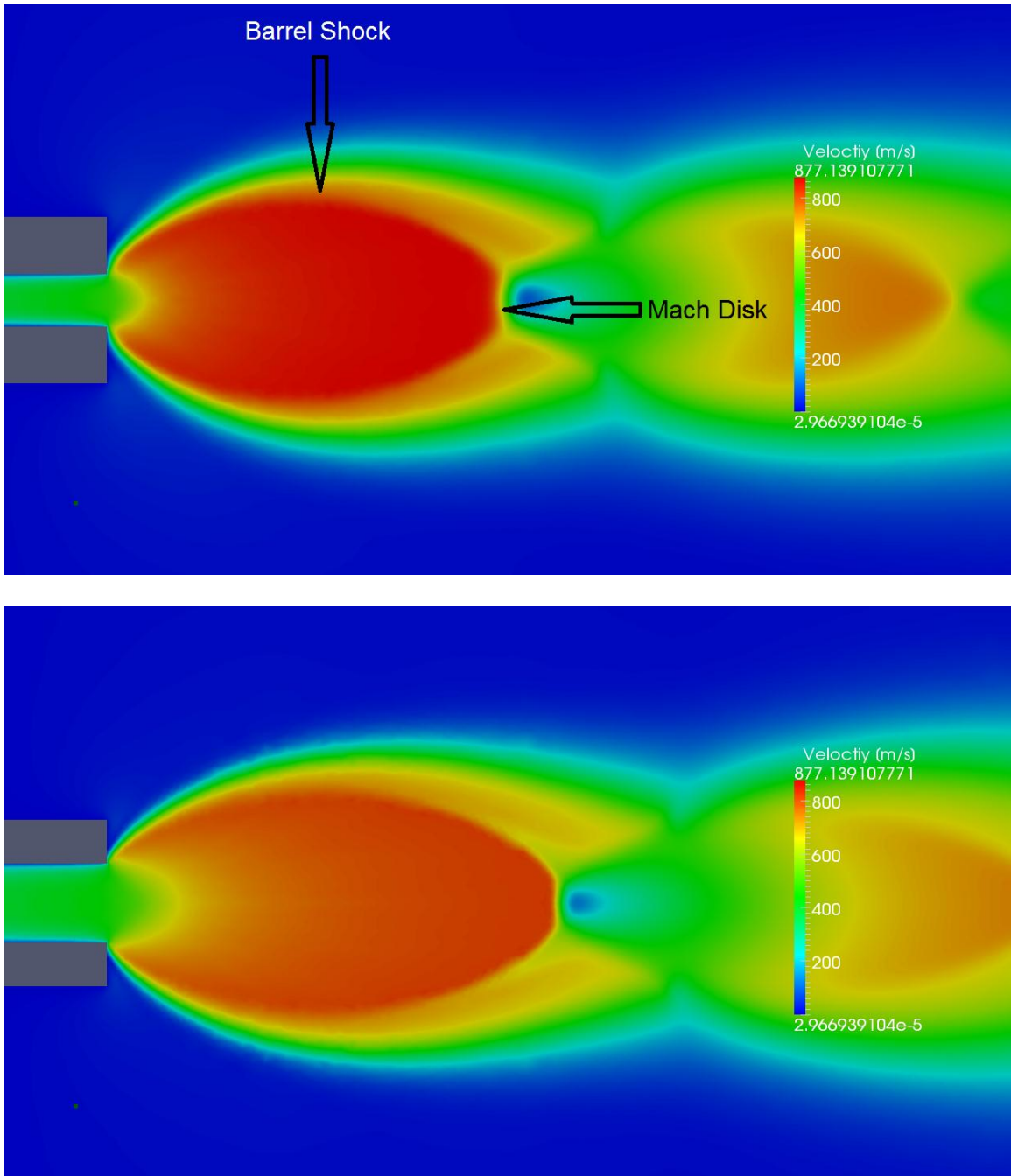


Figure 33: Closeup images of the velocity magnitude of the jet issuing from the capillary in the 20 mil (top) and 30 mil (bottom) cases.

Figure 33 shows the close up image of the jet as it emerges from the capillary. The typical features associated with an underexpanded jet can be easily observed including the barrel shock and Mach disks. The barrel shock is

formed from the reflection of the expansion waves issuing from the capillary exit with the jet boundary. The Mach disk is a strong normal shock occurring normal to the flow direction. The area enclosed by these shocks is known as the zone of silence and the flow within this area is travelling in parallel streamlines. In systems where skimmers are utilized this is the area that is generally sampled to create a high velocity ion jet.

The location of the first Mach disk is over predicted according to the equation in [16]. Although the true reason for this is unknown it is suspected that because the equation applies only to an orifice of zero length and the fact that the sonic line of the flow is not parallel with the opening of the exit of the capillary [18] the equation will not hold true. In the case of a capillary nozzle the sonic line is no parallel with the capillary outlet profile. Instead it curves inward with the fluid reaching Mach 1 approximately several diameters upstream of the outlet on the centerline. For the 20 mil capillary simulation this point is 1.45 mm upstream, for the 30 mil capillary simulation it is 3.65 mm upstream. The oscillation of the flow and resulting shocks downstream are also observed. It should also be noted that the Mach disk location of the 30 mil capillary simulation is farther downstream than that of the 20 mil capillary. This is due to the proportionality of the Mach disk location to the outlet diameter.

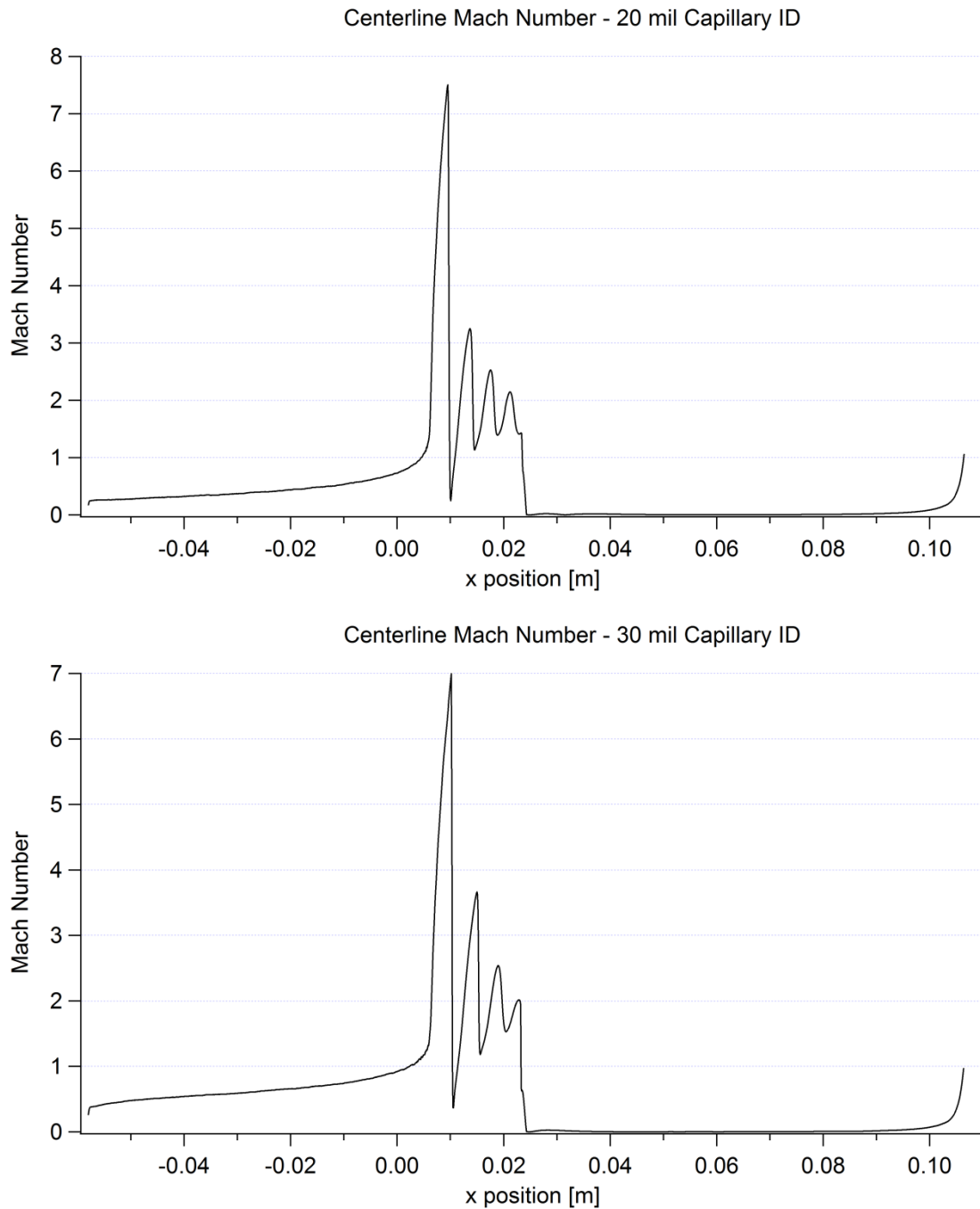


Figure 34: Centerline Mach number values for the 20 mil (top) and 30 mil (bottom) capillary simulations. The capillary exit is at 0.006 m. Note that the flow is at Mach 1 just before the capillary exit and at the last lens where the areas of high pressure difference exist. Also note the oscillating nature of the Mach number as the flow compresses and expands through the jet.

Figure 34 indicates the Mach number along the centerline of the domain from the inlet of the capillary to the outer edge of the last lens. The capillary

begins at -58 mm and ends at 6 mm (the unusual positioning of the origin is due to the constraints for the SIMION simulation mesh). As predicted the flow becomes choked just before exiting the capillary and slightly exceeds Mach 1 at the exit. The oscillating nature of the flow can be easily observed with this figure noting that the increases in velocity are associated with the expansion of the jet and rapid decreases are associated with Mach disk and recompression. The dissipation of these features is also observed. Lastly, at the location of the last lens, which acts as a thin nozzle it can be seen that the flow into the next chamber also becomes choked, reaching Mach 1 at this point as well.

### **6.3.2 Temporal Convergence**

Because the rhoCentralFoam solver is transient there is not a clear way to determine solution convergence as is possible when looking at the residuals of a steady state solver. Because of this some other means of determining convergence is necessary. A point in the flow can be probed and by measuring the amount of relative error of some flow property from one time step to the next conclusions can be drawn as to whether a significant change in the solution is occurring.

Figure 35 show how the maximum x-axis velocity of the jet changes over time. This information was used as convergence criteria for the flow. After the relative error the flow reached a plateau, was less than 1% and no other noticeable changes within the flow were observed the simulation was deemed converged. The 20 mil capillary simulation converged after 0.0018 seconds and the 30 mil capillary converged after 0.000455 seconds. The difference in

convergence times can be attributed to an oscillation of the solution of the 20 mil capillary. Two different jet structures were observed with the final steady solution only emerging after the time mentioned.

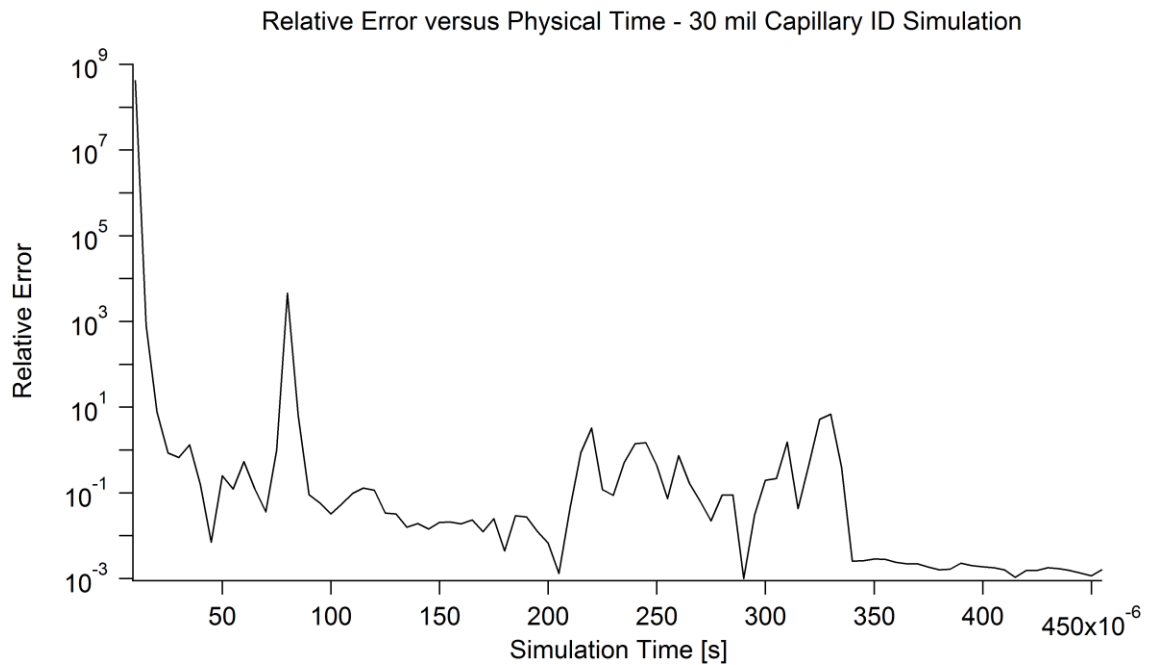
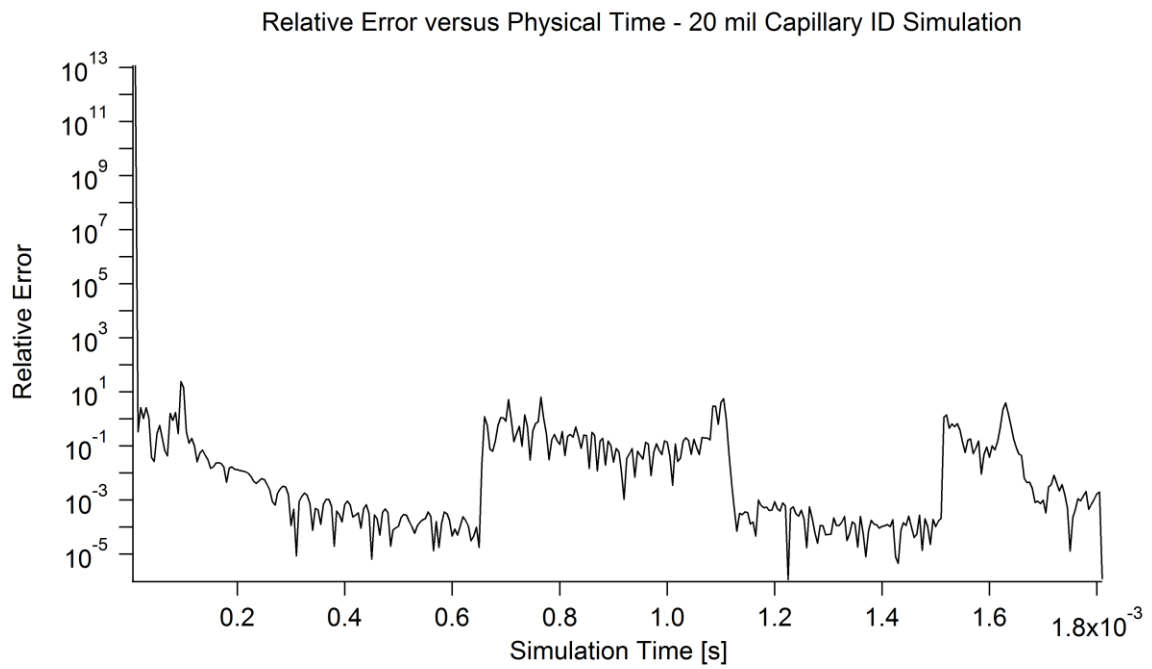


Figure 35: Temporal convergence graphs for 20 mil (top) and 30 mil (bottom) capillary simulations. Note that the 20 mil simulation appears to converge early on. A solution change occurred around 0.00115 s that eventually stabilized out at 0.0018 s.



### 6.3.3 Mesh Convergence

A mesh convergence study was also performed with both simulations. Mesh convergence studies are performed to determine whether changing the number of elements of a mesh has a significant effect on the final solution. Table 1 shows the relative error between the maximum velocities of the jet of three separate simulations. Three mesh sizes were used: one with approximately 90% of the elements, one with 100% of the elements and one with approximately 110% of the elements of the original simulation. A significant difference was found only with the lower resolution 30 mil capillary simulation. In all other simulations the error was less than 1%.

Table 1: Mesh convergence of 20 mil capillary simulation (top) and 30 mil capillary simulation (bottom).

Number of elements	143079 (0.9x)	157737 (1x)	175570 (1.1x)
% difference from 1x	0.775	0	0.9622

Number of elements	155634 (0.9x)	173333 (1x)	193102 (1.1x)
% difference from 1x	8.55	0	0.09

## 6.4 SIMION Simulation Results

### 6.4.1 Standard Operating Conditions

SIMION was used to combine the results of the fluid flow simulations with the electrodynamic effect of the funnel. The fluid flow properties used in the coupled simulations included pressure, temperature and velocity fields. After the creation of the geometry in SIMION and the application of the voltage boundaries, the properties of the ions to be simulated must also be inserted.

These properties include the mass of the particle, its charge state the location and direction of insertion and its kinetic energy. The mass of Cytochrome C is 12,233 amu and its charge is +15 in the solution used. The distribution of the particles was given as a circle with a diameter equal to the ID of the capillary used in the simulation. The kinetic energy was estimated from the velocity at the capillary exit and the mass of Cytochrome C and was calculated to be 12.756 eV. One hundred ions were simulated to provide an appropriate sample amount to observe.

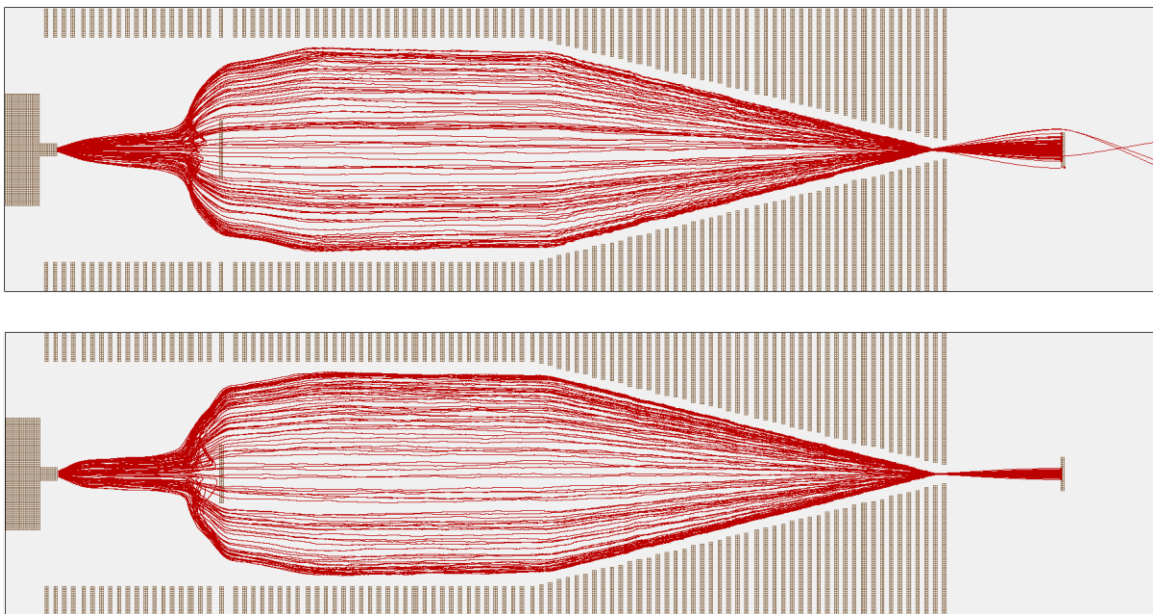


Figure 36: Combined electrodynamic and CFD simulation results for 20 mil (top) and 30 mil (bottom) capillary. The simulation ideally predicts 100% transmission when in reality the momentum of droplets causes losses in various locations.

Figure 36 shows the combined fluid flow and electrodynamic simulations. Qualitative results for both simulations show that the ions enter the chamber and initially follow the stream lines of the flow. As they get farther downstream the

electric field lines begin to have a dominant effect. The subtle differences between the 20 and 30 mil capillary simulations are mainly the product of the pressure differences (and resulting mean free path differences). The ions follow the streamlines of the fluid flow less in the 20 mil simulation than those of the 30 mil simulation. The barrel shock is visible in the path of the ions in the larger capillary instance. The ions are also pushed much closer toward the jet disruptor in the 30 mil case as is observed from the bowing features of the ion paths where the jet impinges. Because the mean free path is shorter in the higher pressure simulations, the air flow will have a greater effect, giving the ions greater kinetic energy in the flow direction. When computing the paths of the ions SIMION would run much faster in the lower pressure case as it had to compute fewer ion-gas collisions.

A notable feature of these simulations is that all ions pass through the funnel, indicating a transmission of 100%. This is not observed in the experimental results from the funnel. It is speculated that many of the ions entering the chamber are contained within liquid droplets. As discussed earlier these droplets will have more mass and therefore more momentum to carry them in their initial direction. It is theorized that these heavier droplets will travel in the axial direction (i.e. toward the jet disruptor) if emitted from the center of the capillary or obliquely from the capillary if emitted from the edge (following the initial ejection through the expanding jet). Because the entire system is modeled as single gas phase ions forces that are influenced by momentum will have less of an effect.

#### 6.4.2 No Electric Field Applied

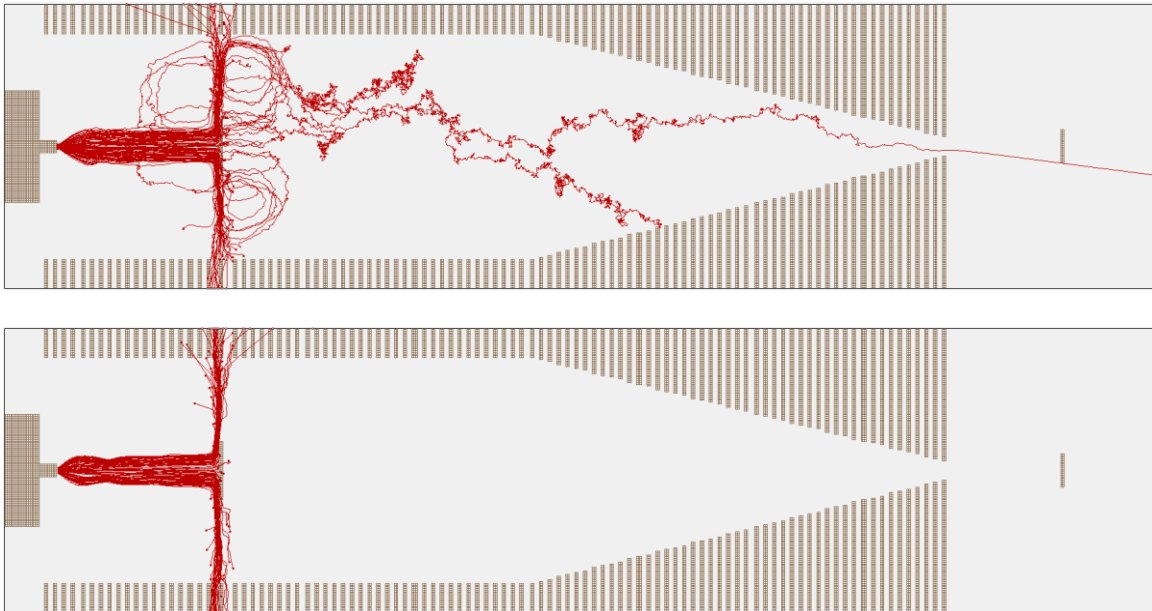


Figure 37: Coupled SIMION and CFD simulations of 20 mil (top) and 30 mil (bottom) capillaries with all electric fields disabled. This figure shows how the ions will follow the streamlines of the flow. Because of the lower pressures in the 20 mil capillary simulation the ions tend to follow a more chaotic path because of the longer mean free path.

Figure 37 shows the effect of the airflow on the ions in the absence of an electric field. The ions are observed following the streamlines of the air while at the same time experiencing collisions with the background gas molecules. One notable aspect of these simulations is that the higher pressure (30 mil) simulation has a dominating effect. When simulating, none of the ions collided with the jet disruptor for this case, seven percent collided in the 20 mil case. It is assumed that the lower pressure resulted in fewer ion-gas collisions and therefore a reduced fluid flow impact. This allows the ions to travel for a longer distance in the direction defined by the last collision. Consequently, the ions of the 20 mil capillary simulation have a greater scattered distribution around the streamlines that define their flow.

### 6.4.3 Variation of Jet Disruptor Voltage

Simulations were performed to correlate the data from the experiment with theoretical predictions. Figure 38 shows the results of these simulations for the 20 and 30 mil capillary cases. Jet disruptor voltages of 170 V and 210 V were simulated. In the 170 V case a strong attractive electric field developed between the capillary and the jet disruptor. This resulted in the majority of the ions being drawn towards the jet disruptor. In the 210 V case a repulsive field formed where the competition between fluid flow and coulomb forces prevented the ions from impacting on the jet disruptor. The ions in the 210 V case became contained to a region where the energy of the particles was not enough to draw them down the funnel axis due to the field produced by the jet disruptor nor was it high enough to allow motion up the voltage gradient present on the lenses.

This is an ideal case and is a result of the fluid flow turbulence model used in the simulation. Because the model is based on the RANS equations no time dependence is present in the fluctuating velocities associated with the air flow. Because of this, a steady flow field is simulated when in reality variations in all flow parameters would exist with respect to time. This would cause the ions to be displaced from this stagnant region and either contact the lenses or be pushed farther down the funnel.

Although the transmission with these simulations is nearly zero in contrast with the experimental results, it demonstrates that there is an optimum jet disruptor voltage that lies in the range observed in these experiments. It is theorized that errors within the system result from the lack of the momentum of

larger droplets that enter the system in addition to the gas phase ions. All simulations were performed without the addition of the ion-gas collisions. The results of these simulations can be found in Appendix B – Additional SIMION Simulation Results.

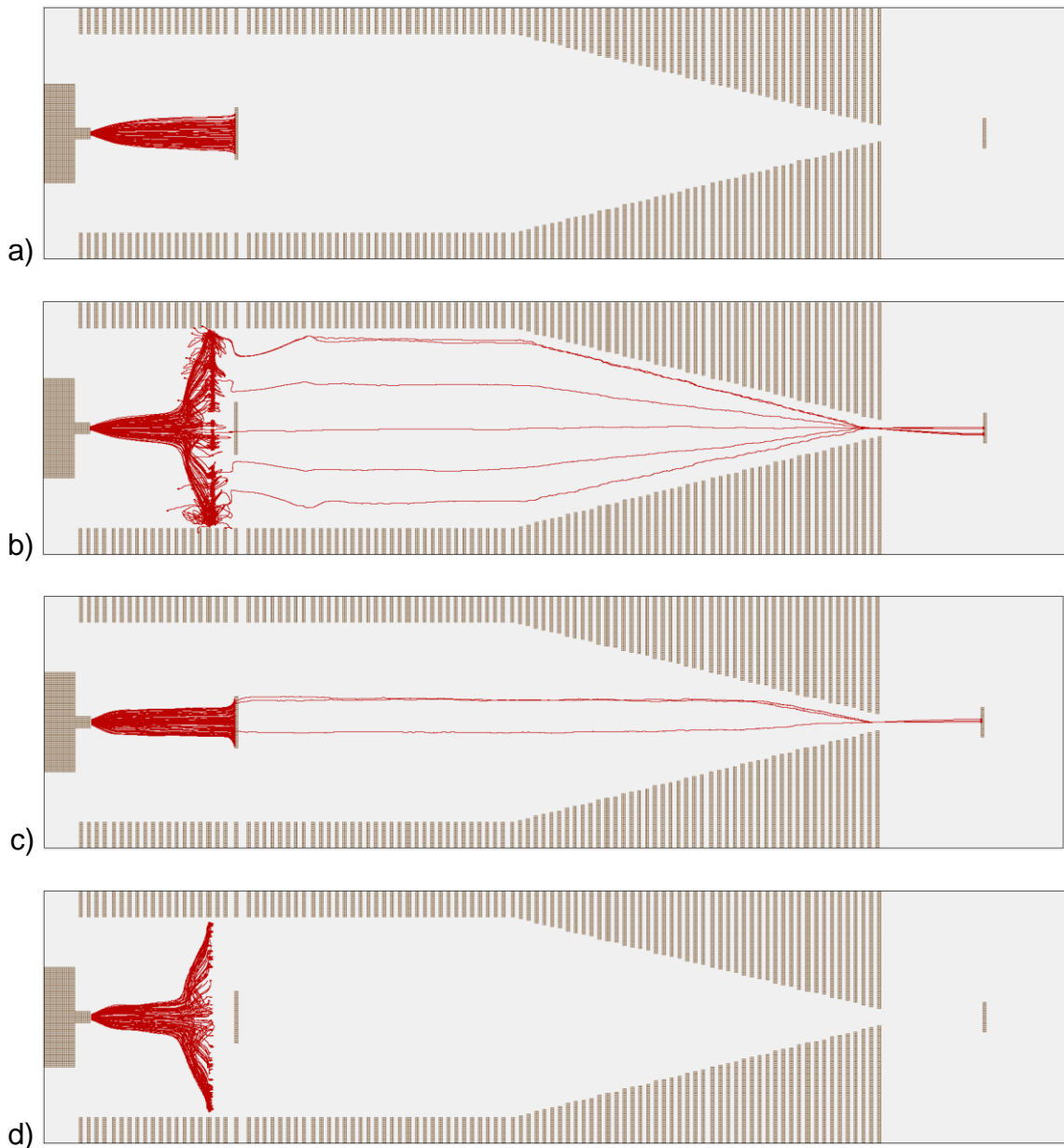


Figure 38: Coupled SIMION and CFD simulations with an applied jet disruptor voltage of 170 V and 210 V for the 20 mil capillary (a, b) and the 30 mil capillary (c, d). With the low voltage a strong attractive force is generated between the jet disruptor and capillary. When the higher voltage is applied a repulsive force is generated. The result of this is little to no transmission. This trend is in agreement with the experimental results.

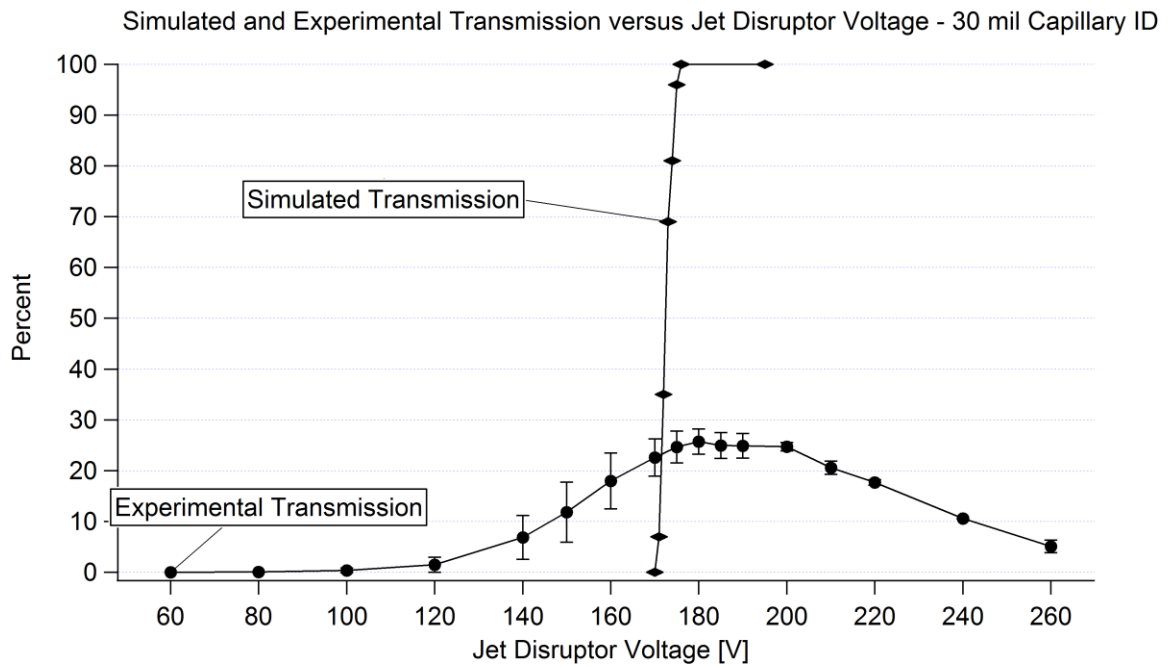
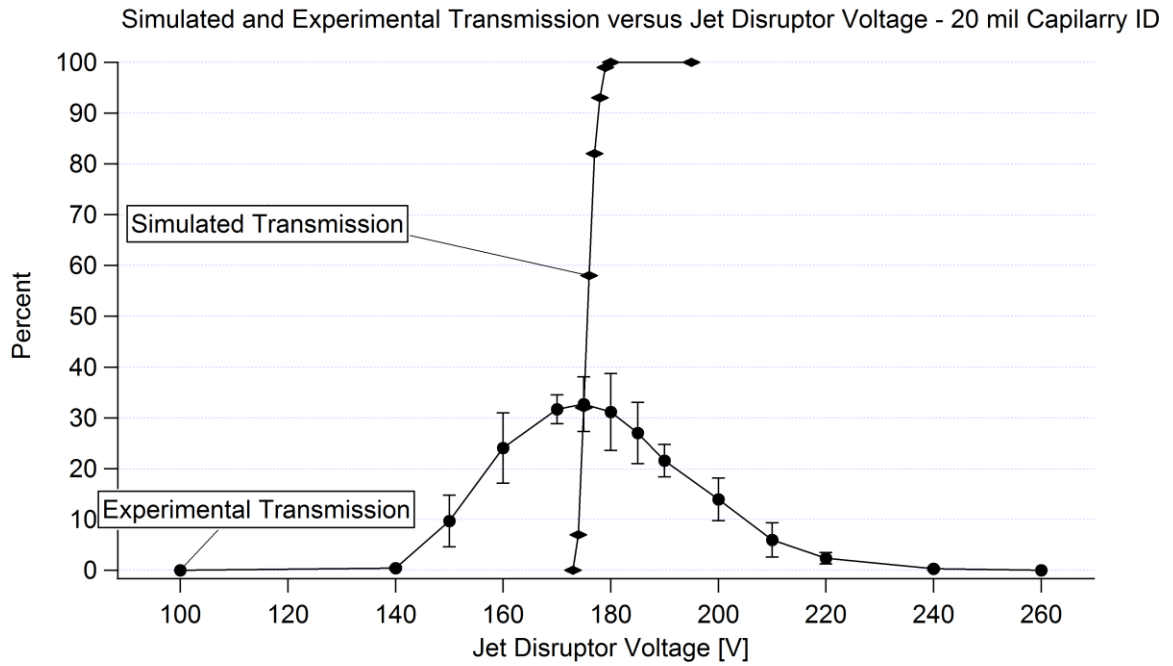


Figure 39: Experimental and simulated values of transmission when varying the jet disruptor voltage for the 20 mil (top) and 30 mil (bottom) cases. Note that the simulated results tend to peak at the maximum transmission values of the experimental results.

Figure 39 shows the experimental and simulated values of the funnel transmission when varying the jet disruptor voltage. The simulated results are much more sensitive to the variation of the voltage and show a peak simulated transmission at the voltage of measured maximum transmission. For the simulated results a variation in jet disruptor voltage of a single volt produced a significant difference in transmission.

Only the first half of the simulated curve is shown as limitations with the software prevent accurate counting of the number of transmitted ions once they become trapped in the stagnation region. However, preliminary results indicate that there is a sharp decline in the transmission with increasing voltage. This decline in transmission begins after the jet disruptor voltage is increased above 195 V for both simulations.

The most evident differences between the simulated and experimental curves are the maximum transmission values, the onset of transmission and width of the peaks. The difference in maximum transmission values can be explained due to lack of droplet modeling explained earlier which results in losses within the system due to the momentum of the droplets directing the ions toward the jet disruptor or lenses. The difference in onset voltages and width of the peaks can be explained by the use of the turbulence model in the CFD simulations. Because the model is based on the RANS equations the turbulent nature of the flow is time averaged and so an inherently unsteady process is modeled as a steady one. Because of this fluctuations present in the physical system are not accurately represented in the model. The fluctuations drive the



ions out of the time averaged streamlines of the flow and around the jet disruptor (in the low jet disruptor voltage case) as well as out of the stagnation region (in the high jet disruptor voltage case).

## 7 Conclusions

Development of the LabVIEW interface allowed for easy manipulation of the parameters of the system that previously required manual changes on individual power supplies. With the introduction of the controller developed by Dr. Rudy Schlaf, and the development of the GUI, a systematic approach to the analysis of the RF ion funnel was made fast and simple. Notable features of the interface include individual focusing element current measurement points and time averaged results.

The experimental data gathered in this study produced information on the effectiveness of droplet removal techniques and transmission based on the variable parameters of the system. When increasing the distance between the needle and capillary the overall current as well as the usable transmitted current decreased. This is an indication that as the distance increases, larger droplets will preferentially enter the funnel due to their higher momentum and resulting in a smaller portion of the spray being sampled. The consequence is a higher ratio of unusable to usable material. A decreasing jet disruptor current to funnel current ratio with increasing temperature was observed. This increased the transmission of the usable current of the system and was occurred as a result of a higher rate of desolvation of the droplets through the capillary/heater assembly. Increasing the flow rate of the system produced larger droplets, again causing a

higher ratio of unusable to usable material. A systematic variation of the transmission versus jet disruptor voltage was also performed. This resulted in an optimum jet disruptor voltage.

Fluid flow simulations were performed which were eventually coupled to SIMION electrodynamic simulations. These simulations were performed for the two capillary sizes using the compressible OpenFOAM solver rhoCentralFoam with the Menter SST turbulence model. The simulations were axisymmetric and were meshed using the open source mesher GMSH. The solution resulting from these simulations produced an underexpanded jet emitting from the capillary exit that would impinge on the surface of the jet disruptor. The results matched the theory indicating that in areas of very large pressure differences the flow would become choked. Some of the limitations of these simulations include assuming an ideal, calorically perfect gas. The symmetry of the system also likely produced inaccuracies compared with the physical system.

The electrodynamic simulations performed with SIMION validated the results of the experimentation. Most notably was the simulation of the optimum jet disruptor voltage which was in agreement with the experimental results. A sharp increase in transmission was observed in the simulation at the point of maximum transmission in the experiments. Inaccuracies were produced mainly due to the fact that all ions were modeled in the gas phase only. In reality many of these ions exist dissolved within small submicron droplets that will have much more mass (and therefore momentum) than their gaseous counterparts.

Some sources of error within the system include:

- No accounting for larger mass droplets
- Idealized simulation conditions including axisymmetric modeling, square feature edges, simplification of geometry
- Pressure outlet boundary condition for OpenFOAM simulations as the system is attached to a vacuum pump and the pressure is most likely not homogeneous at the domain boundary.
- Inherent variability of electrospray process

Further improvements to this study could involve the following:

- Use of smaller OD and ID electrospray emitters. This would result in smaller droplets and a better quality spray.
- Implementation of Lagrangian particle tracking for OpenFOAM simulations to observe the effect of larger mass particles. OpenFOAM's easy to implement solver syntax could be used to include the electrodynamic forces in such a simulation.
- Variation of the other parameters of the system including solution concentration and conductivity, chamber pressure and systematic variation of the other DC voltages on the funnel.
- Repetition of all experiments due to the large variability found with the system to obtain more significant statistical data.

## References

1. Morozov, V., *Electrospray deposition as a method to fabricate functional active protein films*. Analytical Chemistry, 1999. 71: p. 1415-1420.
2. Kebarle, P., *A brief overview of the present status of the mechanisms involved in electrospray mass spectrometry*. Journal of Mass Spectrometry, 2000. 35: p. 804-817.
3. Dole, M., *Molecular beams of macroions*. Journal of Chemical Physics, 1968. 49: p. 2240-2249.
4. Fenn, J., *Electrospray ionization for mass spectrometry of large biomolecules*. Science, 1989. 246: p. 64-71.
5. Schlaf, R. *Schlaf Group USF Engineering*. [cited 2012 May 21]; Available from: <http://rsl.eng.usf.edu/index.htm>.
6. Cloupeau, M., *Electrostatic spraying of liquids in cone-jet mode*. Journal of Electrostatics, 1989. 22: p. 135-159.
7. Loeb, L., *Pulses in negative point-to-plane corona*. Physical Review, 1941. 60: p. 714-722.
8. Cole, R., *Some tenets pertaining to electrospray ionization mass spectrometry*. Journal of Mass Spectrometry, 2000. 35: p. 763-772.
9. Kebarle, P., *From ions in solution to ions in the gas phase - the mechanism of electrospray mass spectrometry*. Analytical Chemistry, 1993. 65: p. 972-986.
10. Fernández De La Mora, J., *The current emitted by highly conducting Taylor cones*. Journal of Fluid Mechanics, 1994. 260: p. 155-184.
11. O'Hanlon, J., *A user's guide to vacuum technology* 2003, Hoboken, New Jersey: John Wiley & Sons Inc.
12. Launder, B., *The numerical computation of turbulent flows*. Computer Methods in Applied Mechanics and Engineering, 1974. 3: p. 269-289.

13. Wilcox, D., *Reassessment of the scale-determining equation for advanced turbulence models*. AIAA Journal, 1988. 26: p. 1299-1310.
14. Menter, R., *Ten years of industrial experience with the SST turbulence model*. Turbulence, Heat and Mass Transfer, 2004. 4.
15. Bartosieqicz, Y., *Numerical and experimental investigations on supersonic ejectors*. International Journal of Heat and Fluid Flow, 2005. 26: p. 56-70.
16. Fenn, J., *Mass spectrometric implications of high-pressure ion sources*. International Journal of Mass Spectrometry, 2000. 200: p. 459-478.
17. Powell, A., *The sound-producing oscillations of round underexpanded jets impinging on normal plates*. Journal of the Acoustical Society of America, 1988. 83: p. 515-533.
18. Murphy, H., *Effects of nozzle geometry of kinetics in free-jet expansions*. Journal of Physical Chemistry, 1984. 88: p. 4474-4478.
19. *OpenFOAM user guide*. 2012 [cited 2012 May 15]; Available from: <http://www.openfoam.org/docs/user/>.
20. Versteeg, H., *An introduction to computational fluid dynamics - the finite volume method* 1995, Essex, England: Pearson Education Limited.
21. Kurganov, A., *New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations*. Journal of Computational Physics, 2000. 160: p. 241-282.
22. Kurganov, A., *Central-upwind schemes on triangular grids for hyperbolic systems of conservation laws*. Numerical Methods for Partial Differential Equations, 2005. 21: p. 536-552.
23. Greenshields, C., *Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed viscous flows*. International Journal for Numerical Methods in Fluids, 2009.
24. van Leer, B., *Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme*. Journal of Computational Physics, 1974. 14: p. 631-370.
25. Ladenburg, R., *Interferometric studies of faster than sound phenomena. Part I. The gas flow around various objects in a free, homogeneous, supersonic air stream*. Physical Review, 1948. 73: p. 1359-1377.

26. *SIMION: Ion and Electron Optics Simulator*. [cited 2012 May 25]; Available from: <http://www.simion.com>.
27. Shaffer, S., *A novel ion funnel for focusing ions at elevated pressure using electrospray ionization mass spectrometry*. *Rapid Communications in Mass Spectrometry*, 1997. 11: p. 1813-1817.
28. Geuzaine, C., *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. *International Journal for Numerical Methods in Engineering*, 2009. 0: p. 1-24.
29. Schöberl, J., *An advancing front 2D/3D-mesh generator based on abstract rules*. *Computing and Visualization in Science*, 1997. 1: p. 41-52.
30. Page, J., *Ionization and transmission efficiency in an electrospray ionization-mass spectrometry interface*. *Journal of American Society for Mass Spectrometry*, 2007. 18: p. 1582-1590.

## Appendices



## Appendix A – OpenFOAM Simulation Configuration Files

The following pages include the boundary condition files and other important configuration files for the rhoCentralFoam solver in OpenFOAM.

Generally each file has a header that looks as follows:

```
/*-----*-- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.0.1 |
| \\ / A n d | Web: www.OpenFOAM.com |
| \\ / M a n i p u l a t i o n |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       alphas;
}
// ***** //
```

as an example. This header has been omitted from these files. The boundary files are nearly identical for the 20 and 30 mil capillary ID simulations aside from the pressure files. Only one copy of each of the files will be listed in the appendix aside from the pressure boundary fields

## Appendix A (Continued)

### $\alpha_t$ Boundary File

```
dimensions      [1 -1 -1 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    inlet
    {
        type      calculated;
        value     uniform 0;
    }

    outlet
    {
        type      calculated;
        value     uniform 0;
    }

    freestream
    {
        type      calculated;
        value     uniform 0;
    }

    chamberWalls
    {
        type      alphasatWallFunction;
        Prt       0.85;
        value     uniform 0;
    }

    capWalls
    {
        type      alphasatWallFunction;
        Prt       0.85;
        value     uniform 0;
    }

    lenses
    {
        type      alphasatWallFunction;
        Prt       0.85;
        value     uniform 0;
    }

    wedge1 {type wedge;}
    wedge2 {type wedge;}
}

// ***** //
```

## Appendix A (Continued)

### k Boundary File

```
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 6.933e-4;
boundaryField
{
    inlet
    {
        type          inletOutlet;
        inletValue     $internalField;
        value          $internalField;
    }

    outlet
    {
        type          zeroGradient;
    }

    freestream
    {
        type          zeroGradient;
    }

    chamberWalls
    {
        type          fixedValue;
        value         uniform 1e-12;
    }

    capWalls
    {
        type          fixedValue;
        value         uniform 1e-12;
    }

    lenses
    {
        type          fixedValue;
        value         uniform 1e-12;
    }

    wedge1 {type wedge;}
    wedge2 {type wedge;}
}

// ***** //
```

## Appendix A (Continued)

### $\mu_t$ Boundary File

```
dimensions      [1 -1 -1 0 0 0 0];
internalField   uniform 1e-12;

boundaryField
{
    inlet
    {
        type      calculated;
        value     $internalField;
    }

    outlet
    {
        type      calculated;
        value     $internalField;
    }

    freestream
    {
        type      calculated;
        value     $internalField;
    }

    chamberWalls
    {
        type      fixedValue;
        value     $internalField; /*mutUWallFunction;
        Cmu       0.09;
        kappa     0.41;
        E         9.8;
        value     uniform 0;*/
    }

    capWalls
    {
        type      fixedValue;
        value     $internalField; /*mutUWallFunction;
        Cmu       0.09;
        kappa     0.41;
        E         9.8;
        value     uniform 0;*/
    }

    lenses
    {
        type      fixedValue;
        value     $internalField; /*mutUWallFunction;
        Cmu       0.09;
        kappa     0.41;
        E         9.8;
        value     uniform 0;*/
    }

    wedge1 {type wedge;}
    wedge2 {type wedge;}
}

// ***** //
```

## Appendix A (Continued)

### $\omega$ Boundary File

```
dimensions      [0 0 -1 0 0 0 0];
internalField   uniform 2.633;

boundaryField
{
    inlet
    {
        type          inletOutlet;
        inletValue     $internalField;
        value          $internalField;
    }

    outlet
    {
        type          zeroGradient;
    }

    freestream
    {
        type          zeroGradient;
    }

    chamberWalls
    {
        type          compressible::omegaWallFunction;
        value         $internalField;
    }

    capWalls
    {
        type          compressible::omegaWallFunction;
        value         $internalField;
    }

    lenses
    {
        type          compressible::omegaWallFunction;
        value         $internalField;
    }

    wedge1 {type wedge;}
    wedge2 {type wedge;}
}

// ***** //
```

## Appendix A (Continued)

### T Boundary File

```
dimensions      [0 0 0 1 0 0 0];
internalField   uniform 293.0;
boundaryField
{
    inlet
    {
        type      fixedValue;
        value     uniform 293;
    }

    outlet
    {
        type      zeroGradient;
    }

    freestream
    {
        type      zeroGradient;
    }

    chamberWalls
    {
        type      zeroGradient;
    }

    capWalls
    {
        type      fixedValue;
        value     uniform 423;
    }

    lenses
    {
        type      zeroGradient;
    }

    wedge1 {type wedge;}
    wedge2 {type wedge;}
}

// ***** //
```

## Appendix A (Continued)

### U Boundary File

```
dimensions      [0 1 -1 0 0 0];
internalField   uniform (0 0 0);
boundaryField
{
    wedge2
    {
        type      wedge;
    }
    outlet
    {
        type      inletOutlet;
        inletValue uniform (0 0 0);
        value     uniform (0 0 0);
    }
    lenses
    {
        type      fixedValue;
        value     uniform (0 0 0);
    }
    inlet
    {
        type      pressureInletOutletVelocity;
        value     uniform (0 0 0);
    }
    wedge1
    {
        type      wedge;
    }
    chamberWalls
    {
        type      fixedValue;
        value     uniform (0 0 0);
    }
    freestream
    {
        type      inletOutlet;
        inletValue uniform (0 0 0);
        value     uniform (0 0 0);
    }
    capWalls
    {
        type      fixedValue;
        value     uniform (0 0 0);
    }
}

// ***** //
```

## Appendix A (Continued)

### p Boundary File for 20 mil Capillary Simulation

```
dimensions      [1 -1 -2 0 0 0 0];

internalField   nonuniform List<scalar>
157737
( NOTE LIST OMITED DUE TO LENGTH )
;

boundaryField
{
    wedge2
    {
        type      wedge;
    }
    outlet
    {
        type      waveTransmissive;
        gamma     1.4;
        fieldInf  3;
        lInf      0.01;
        value     uniform 3;
    }
    lenses
    {
        type      zeroGradient;
    }
    inlet
    {
        type      fixedValue;
        value     uniform 101325;
    }
    wedge1
    {
        type      wedge;
    }
    chamberWalls
    {
        type      zeroGradient;
    }
    freestream
    {
        type      fixedValue;
        value     uniform 456;
    }
    capWalls
    {
        type      zeroGradient;
    }
}

// ***** //
```



## Appendix A (Continued)

### p Boundary File for 30 mil Capillary Simulation

```
dimensions      [1 -1 -2 0 0 0 0];

internalField   nonuniform List<scalar>
173333
( NOTE LIST OMITED DUE TO LENGTH )
;

boundaryField
{
    wedge2
    {
        type      wedge;
    }
    outlet
    {
        type      waveTransmissive;
        gamma     1.4;
        fieldInf  3;
        lInf      0.01;
        value     uniform 3;
    }
    lenses
    {
        type      zeroGradient;
    }
    inlet
    {
        type      fixedValue;
        value     uniform 101325;
    }
    wedge1
    {
        type      wedge;
    }
    chamberWalls
    {
        type      zeroGradient;
    }
    freestream
    {
        type      fixedValue;
        value     uniform 1172;
    }
    capWalls
    {
        type      zeroGradient;
    }
}

// ***** //
```

## Appendix A (Continued)

### controlDict File

```
application      rhoCentralFoam;
startFrom        latestTime;
startTime        0;
stopAt           endTime;
endTime          0.010;
deltaT           1e-10;
writeControl     adjustableRunTime;
writeInterval    5e-06;
cycleWrite       0;
writeFormat      ascii;
writePrecision   7;
writeCompression off;
timeFormat       general;
timePrecision    7;
adjustTimeStep   yes;
maxCo            0.5;
maxDeltaT        1;
```

```
// ***** //
```

## Appendix A (Continued)

### decomposeParDict File

```
numberOfSubdomains 64;

method          simple;

simpleCoeffs
{
    n            ( 64 1 1 );
    delta        0.001;
}

hierarchicalCoeffs
{
    n            ( 1 1 1 );
    delta        0.001;
    order        xyz;
}

manualCoeffs
{
    dataFile     "";
}

distributed     no;

roots           ( );

// ***** //
```

## Appendix A (Continued)

### fvSchemes File

```
fluxScheme      Kurganov;

ddtSchemes
{
    default      Euler;
}

gradSchemes
{
    default      Gauss linear;
}

divSchemes
{
    default      none;
    div(tauMC)   Gauss linear;
    div(phi,epsilon) Gauss limitedLinear 1;
    div(phi,k)   Gauss upwind;
    div(phi,omega) Gauss upwind;
}

laplacianSchemes
{
    default      Gauss linear corrected;
}

interpolationSchemes
{
    default      linear;
    reconstruct(rho) vanLeer;
    reconstruct(U) vanLeerV;
    reconstruct(T) vanLeer;
}

snGradSchemes
{
    default      corrected;
}

// ***** //
```

## Appendix A (Continued)

### fvSolution File

```
solvers
{
  "(rho|rhoU|rhoE)"
  {
    solver          diagonal;
  }

  U
  {
    solver          smoothSolver;
    smoother        GaussSeidel;
    nSweeps         2;
    tolerance        1e-10;
    relTol          0;
  }

  "(e|epsilon|omega|k)"
  {
    $U;
    tolerance        1e-10;
    relTol          0;
  }
}

// ***** //
```

## Appendix A (Continued)

### RAS Properties File

```
RASModel      kOmegaSST;
```

```
turbulence    on;
```

```
printCoeffs   on;
```

```
// ***** //
```

## Appendix A (Continued)

### thermophysicalProperties File

```
thermoType
ePsiThermo<pureMixture<sutherlandTransport<specieThermo<hConstThermo<perfectGas>>>>>>;

mixture
{
    specie
    {
        nMoles      1;
        molWeight   28.96;
    }
    thermodynamics
    {
        Cp          1004.5;
        Hf          0;
    }
    transport
    {
        As          1.458e-06;
        Ts          110.4;
    }
}

// ***** //
```

## **Appendix B – Additional SIMION Simulation Results**

This section contains the SIMION simulation results with the absence of the ion-gas collisions. This serves as evidence that the addition of the ion-gas collisions is necessary for the simulation to give physically realistic results.



## Appendix B (Continued)

### 20 mil Capillary Simulation Results

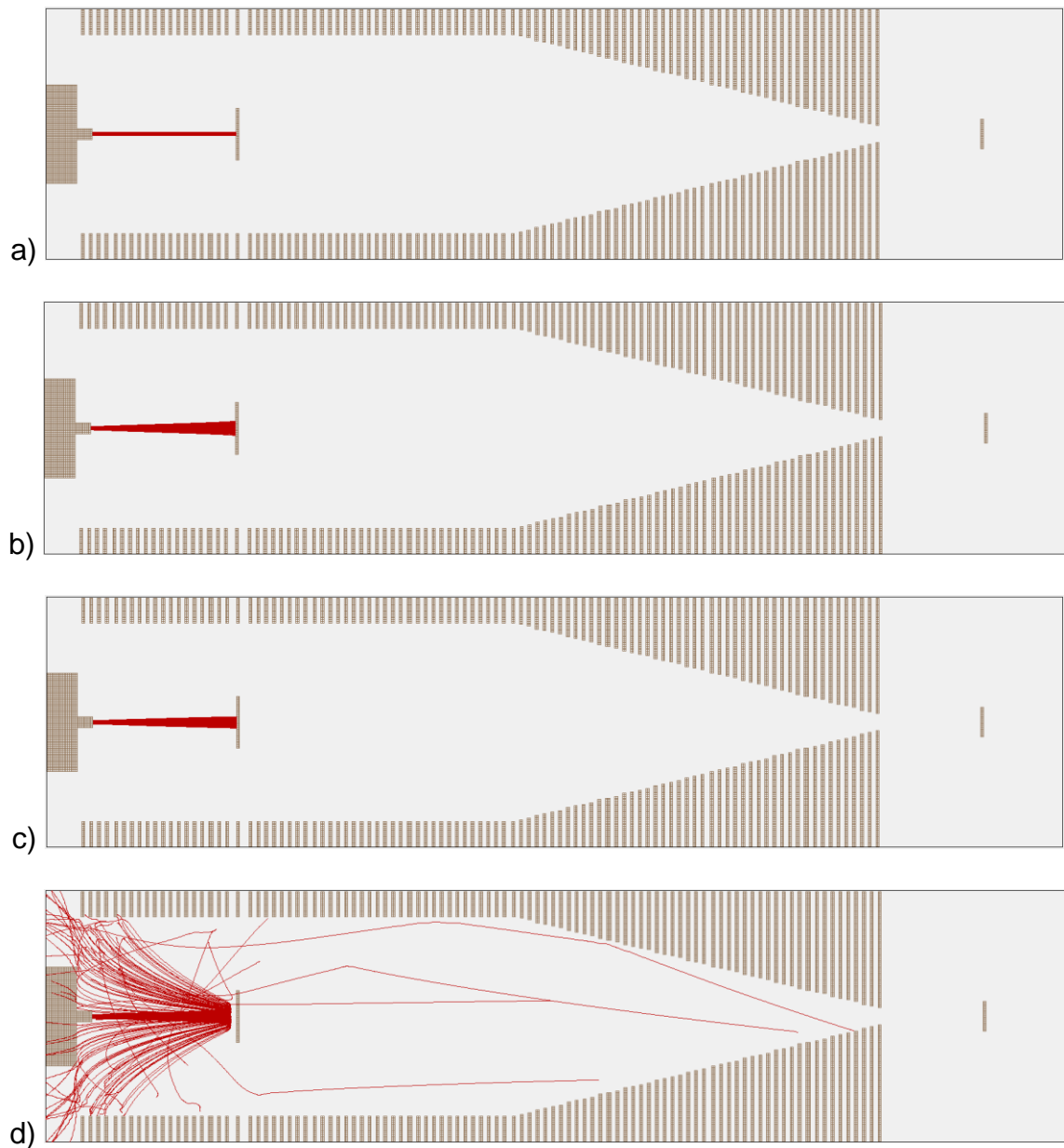


Figure 40: SIMION simulation results of 20 mil capillary with no fluid flow coupling. a) no electric field; b) standard operating conditions; c) standard operating conditions with 170 V jet disruptor; d) standard operating conditions with 210 V jet disruptor.

## Appendix B (Continued)

### 30 mil Capillary Simulation Results

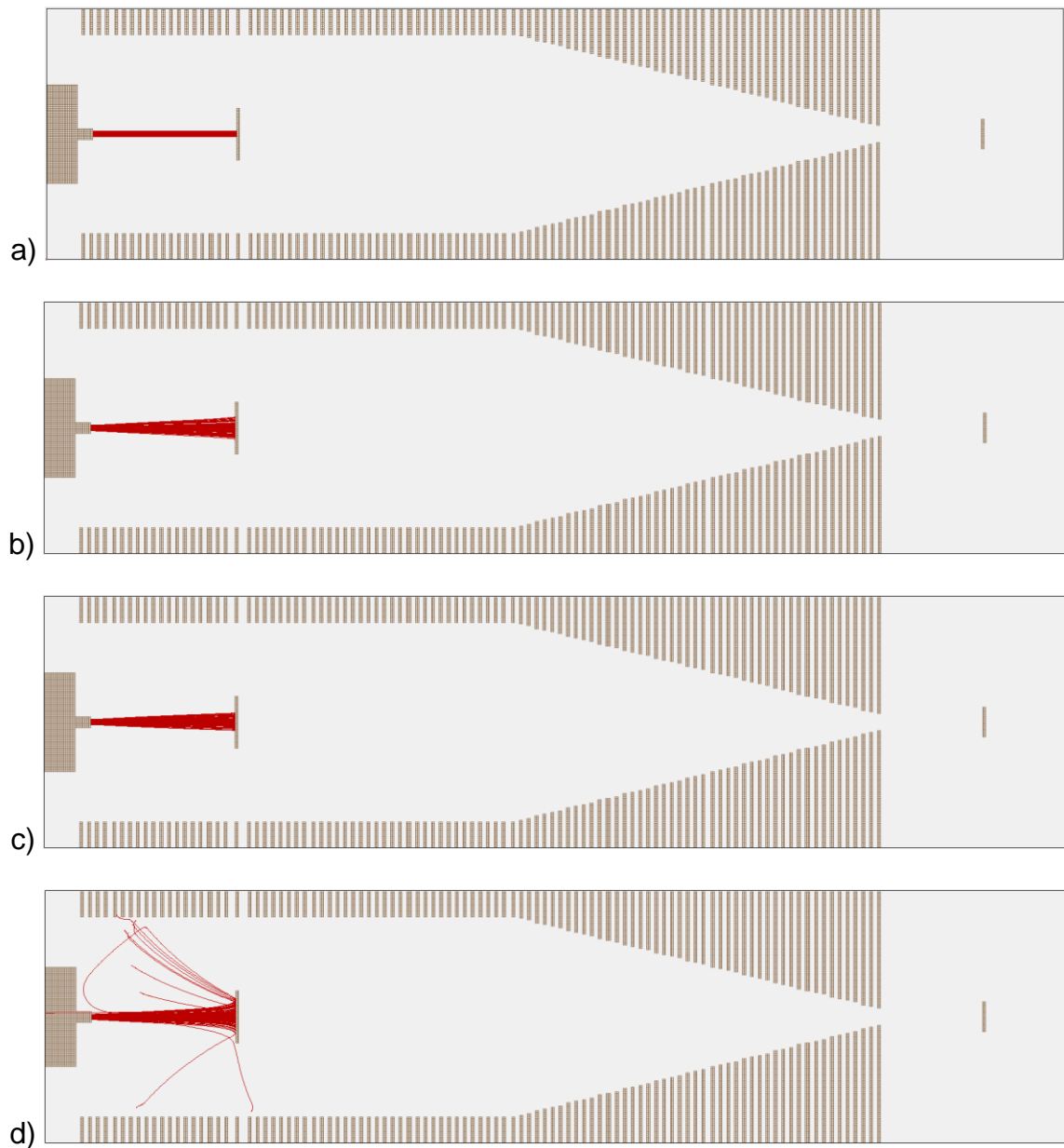



Figure 41: SIMION simulation results of 30 mil capillary with no fluid flow coupling. a) no electric field; b) standard operating conditions; c) standard operating conditions with 170 V jet disruptor; d) standard operating conditions with 210 V jet disruptor.

## Appendix C – Third Party Material Permissions

### Written permission for Figure 6



Welcome, Eric  
Not you?

Log out | Cart (1) | Manage Account | Feedback | Help | Live Help

GET PERMISSION | LICENSE YOUR CONTENT | PRODUCTS AND SOLUTIONS | PARTNERS | EDUCATION | ABOUT US

Get Permission / Find Title  
   
Advanced Search Options

**The journal of physical chemistry**

**ISSN:** 0022-3654  
**Publication year(s):** 1952 - 1999  
**Author/Editor:** AMERICAN CHEMICAL SOCIETY ; FARADAY SOCIETY ; CHEMICAL SOCIETY (GREAT BRITAIN)  
**Publication type:** Journal  
**Publisher:** AMERICAN CHEMICAL SOCIETY [ETC.]  
**Rightsholder:** AMERICAN CHEMICAL SOCIETY

**Language:** English  
**Country of publication:** United States of America

**Permission type selected:** Republish or display content  
**Type of use selected:** reuse in a Thesis/Dissertation  
[Select different permission](#)

**Article title:** Effects of nozzle geometry on kinetics in free-jet expansions  
**Author(s):** Murphy, Hylton R. ; Miller, David R.  
**DOI:** 10.1021/J150664A005  
**Date:** Sep 1, 1984  
**Volume:** 88  
**Issue:** 20  
[Select different article](#)

**Permission for this particular request is granted for print and electronic formats, and translations, at no charge. Figures and tables may be modified. Appropriate credit should be given. Please print this page for your records and provide a copy to your publisher. Requests for up to 4 figures require only this record. Five or more figures will generate a printout of additional terms and conditions. Appropriate credit should read: "Reprinted with permission from {COMPLETE REFERENCE CITATION}. Copyright {YEAR} American Chemical Society." Insert appropriate information in place of the capitalized words.**

**If credit is given to another source for the material you requested, permission must be obtained from that source.**

**Enter the following details to determine a price:**

**Requestor Type**

**Portion**

**Number of Table/Figure/Micrographs**

**Format**

## Appendix C (Continued)

### Written Permission for Figure 2

#### Order Details

##### Journal of electrostatics

Billing Status:  
N/A

**Order detail ID:** 62488354  
**Article Title:** Electrostatic spraying of liquids in cone-jet mode  
**Author(s):** Cloupeau, M. ; Prunet-Foch, B.  
**DOI:** 10.1016/0304-3886(89)90081-8  
**Date:** Jan 01, 1989  
**ISSN:** 0304-3886  
**Publication Type:** Journal  
**Volume:** 22  
**Issue:** 2  
**Start page:** 135  
**Publisher:** ELSEVIER BV

**Permission Status:**  **Granted**

**Permission type:** Republish or display content  
**Type of use:** reuse in a thesis/dissertation  
**Order License Id:** 2917191471276

[View details](#)

**Note:** This item was invoiced separately through our [RightsLink service](#). [More info](#)


\$ 0.00

### Written permission for Figure 3

##### Journal of mass spectrometry

Billing Status:  
N/A

**Order detail ID:** 62488357  
**Article Title:** Some tenets pertaining to electrospray ionization mass spectrometry  
**Author(s):** Cole, Richard B.  
**DOI:** 10.1002/1096-9888(200007)35:7<76  
JMS16>3.3.CO;2-R [View details](#)  
**Date:** Jan 01, 2000  
**ISSN:** 1096-9888  
**Publication Type:** e-Journal  
**Volume:** 35  
**Issue:** 7  
**Start page:** 763  
**Publisher:** JOHN WILEY & SONS LTD.  
**Author/Editor:** Wiley InterScience (Online service)

**Permission Status:**  **Granted**

**Permission type:** Republish or display content  
**Type of use:** reuse in a dissertation/thesis  
**Order License Id:** 2917191475901

**Note:** This item was invoiced separately through our [RightsLink service](#). [More info](#)

\$ 0.00

### Written Permission for Figure 5

##### International journal of mass spectrometry

Billing Status:  
N/A

**Order detail ID:** 62488355  
**Article Title:** Mass spectrometric implications of high-pressure ion sources  
**Author(s):** Fenn, John B.  
**DOI:** 10.1016/S1387-3806(00)00328-6  
**Date:** Dec 01, 2000  
**ISSN:** 1873-2798  
**Publication Type:** e-Journal  
**Volume:** 200  
**Issue:** 1-3  
**Start page:** 459  
**Publisher:** Elsevier Science

**Permission Status:**  **Granted**

**Permission type:** Republish or display content  
**Type of use:** reuse in a thesis/dissertation  
**Order License Id:** 2917191472963

[View details](#)

**Note:** This item was invoiced separately through our [RightsLink service](#). [More info](#)

\$ 0.00


## Appendix C (Continued)

### Written Permission for Figure 7

International journal for numerical methods in fluids

Billing Status:  
N/A

**Order detail ID:** 62488356

**Permission Status:**  **Granted**

**Article Title:** Implementation of semi-discrete, non-staggered central schemes in a collocated, polyhedral, finite volume framework, for high-speed viscous flows

**Permission type:** Republish or display content reuse in a dissertation/thesis

**Type of use:**

**Order License Id:** 2917191474432

[View details](#)

**Author(s):** Greenshields, Christopher J. ; et al

**DOI:** 10.1002/FLD.2069

**Date:** Jan 01, 2009

**ISSN:** 0271-2091

**Publication Type:** Journal

**Volume:**

**Issue:**

**Start page:** n/a

**Publisher:** JOHN/WILEY & SONS LTD.

**Note:** This item was invoiced separately through our **RightsLink service**. [More info](#)

\$ 0.00