THE CATHODE PLASMA SIMULATION

by

Thada Suksila

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ASTRONAUTICAL ENGINEERING)

May 2015

Copyright 2015 Thada Suksila

UMI Number: 3704256

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI
Dissertation Publishing

UMI 3704256
Published by ProQuest LLC (2015). Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.
All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code

ProQuest

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Dedication

...to my parents; General Anuchat and Nukul Suksila for their love, support and countless sacrifices so that I could have opportunities.

Acknowledgments

I would like to thank all the people who were involved in helping me with this dissertation. Without you, it never would have come together.

First, I would like to express my deepest gratitude and sincereness to my advisor, Professor Dr. Daniel Erwin, who provided me the freedom to explore this research topic that most interested me. Also, thank you for many visits and discussions to your office. I appreciated your time and help to guide me through the completion of my research.

In addition, I would like to acknowledge my committee members: Professor Dr. Mike Gruntman, Professor Dr. Joseph Kunc, Professor Dr. Aiichiro Nakano, and Professor Dr. Joseph Wang, who attended and provided helpful guidance for my research during the qualifying and defense exams.

I would especially like to thank Dr. Keith David Goodfellow, who provided insightful comments at different stages of my research, specially on the plasma sheath model in magnetoplasmadynamics (MPD) thrusters. His comments on this model helped me improve my knowledge in the area.

I appreciated the writing instructors and lecturers who helped me edit my dissertation draft many times: Dr. Fife Elizabeth, Mary Ann Murphy, Amanda Bloom and Merve Aktar. Also, many thanks to Dr. Jungmiwha Bullock, who gave

many suggestions of guidelines and strategies to complete my dissertation within timeframe.

Another thank you to our department staff who handled all of my documents efficiently and guided me through graduate school bureaucracy: Dell, Ana, Marritta, and Norma. Also, thank you to my fellow graduate student colleagues and friends, who accompanied me on this long endeavor: John, Ning, Doug, Sam, Pedro, Daoru, Ouliang, Dayung, Kevin, Will, Joshua, Janeane, Joel, Marcus, Suradej, Manachai, Jittraporn, MorTa, Akarapan, Surapong, and Kritsapong.

Most importantly, none of this would have been possible without the love and patience of my family. Therefore, this dissertation is dedicated to My father, General Anuchat Suksila; my mother, Nukul Suksila; and my sister, Raphephan Naka. They have been a constant source of love, support, concern, and strength all these years. They have always encouraged me to achieve excellence.

Furthermore, I am so grateful to my grandmother, Dr. Kanya Sitanggan, who allowed me to stay with her family near the completion of my program. In addition, I appreciated my aunts, Ubon and Benja Puangsuvan, who provided many great advices during the years.

Finally yet importantly, I would like to express my appreciation to the King Mongkut's University of Technology North Bangkok, and the Thai Ministry of Science and Technology for granting me the opportunity to come to USC as a Thai Scholarship student and pursue my master's and doctoral degrees. I would also like to thank the Astronautical Engineering department at USC for helping me to fund my research by providing me the opportunity to work as a teaching assistant after my government scholarship expired.

Table of Contents

List of Tables

List of Figures

Abbreviations and Symbols

j_b	the thermionic current density (A/cm^2)
I_{sp}	the specific impulse (s)
F_{th}, F	the thrust (N)
Δv	the velocity change (m/s)
j, J_0, J	the current density (A/m^2)
g_o	the acceleration of gravity on Earth (m/s^2)
e	the electron charge (C)
E_c, E	the electric field (V/m)
h	the Planck's constant ($J \cdot s$)
k	the Boltzmann constant (J/K)
c	the speed of light (m/s)
A_R	the Richardson coefficient ($A/m^2/K^2$)
B	the magnetic field
B_θ	the magnetic field in θ direction
ϕ	the potential or voltage (V)
ϕ_{eff}	the effective potential (V)
φ	the work function (eV)
φ_{eff}	the effective work function (eV)
φ_0	the reference work function (eV)

m	mass (kg)
T	temperature (K)
T_i	the temperature at grid i(K)
$\bar{T}_i$	the average temperature at grid i (K)
T_e	the electron temperature (eV)
T_c	the cathode temperature (K)
n_e	the electron density $(\#/cm^3)$
M_P	the spacecraft propellant mass (kg)
M_B	the burnout spacecraft mass (kg)
M_0	the total spacecraft mass (kg)
u_e	the exit velocity (m/s)
ε_0, ε	the permittivity of free space $(C^2/N/m^2)$
L_i	the primary cell length of cell i(m)
L_{s_i}	the secondary cell length of cell i (m)
K, K_{th}	the thermal conductivity $(W/m/K)$
σ, σ_e	the electrical conductivity (S/m)
σ_s	the StefanBoltzmann constant $(W/m^2/K^4)$
c_p	the heat capacity $(J/m^3/K)$
V_F	the voltage drop (V)
r_a	the anode radius (m)
r_c	the cathode radius (m)
$L_{cathode}$, L_c	the total length of the cathode (m)
L_{anode}, L_a	the total length of the anode (m)
L_{plasma}, L_p	the total length of the plasma region (m)
S_i	the surface area of i (m^2)
V_{anode}	the anode voltage (V)

$V_{cathode}$	the cathode voltage (V)
V_{int}	the initial voltage (V), the voltage different (V)
h_c	the heat transfer coefficient ($W/m^2/K$)
T_i^{n+1}	the temperature at grid i at time step n+1 (K)
T_i^n	the temperature at grid i at time step n (K)
JE	the Ohmic heating
Φ	the diameter (m)
q, $\dot{Q}$	the heat flux (W/m^2)
P	the pressure (Pa)
t_{ss}	the time to reach steady state (second)
T_{base}	the cathode base temperature (K)
T_{int}	the initial temperature (K)
Δt	the time increment (second)
$\dot{m}$	the propellant mass flow rate (kg/s)
T_{inf}	the environment temperature (K)
j_z	the current density in z direction (A/m^2)
j_r	the current density in r direction (A/m^2)
D	the diffusion coefficient
v_c	the sheath voltage (V)
I, I_i	the total current, the total current at grid i (A)
w_i	the coupling coefficient [1]
a_{i+1}	the triangular area (m^2)

Abstract

Since its invention at the University of Stuttgart, Germany in the mid-1960, scientists have been trying to understand and explain the mechanism of the plasma interaction inside the magnetoplasmadynamics (MPD) thruster. Because this thruster creates a larger level of efficiency than combustion thrusters, this MPD thruster is the primary cadidate thruster for a long duration (planetary) spacecraft. However, the complexity of this thruster make it difficult to fully understand the plasma interaction in an MPD thruster while operating the device. That is, there is a great deal of physics involved: the fluid dynamics, the electromagnetics, the plasma dynamics, and the thermodynamics. All of these physics must be included when an MPD thruster operates.

In recent years, a computer simulation helped scientists to simulate the experiments by programing the physics theories and comparing the simulation results with the experimental data. Many MPD thruster simulations have been conducted: E. Niewood et al.[5], C. K. J. Hulston et al.[6], K. D. Goodfellow[3], J Rossignol et al.[7]. All of these MPD computer simulations helped the scientists to see how quickly the system responds to the new design parameters.

For this work, a 1D MPD thruster simulation was developed to find the voltage drop between the cathode and the plasma regions. Also, the properties such as thermal conductivity, electrical conductivity and heat capacity are temperature

and pressure dependent. These two conductivity and heat capacity are usually definded as constant values in many other models. However, this 1D and 2D cylindrical symmetry MPD thruster simulations include both temperature and pressure effects to the electrical, thermal conductivities and heat capacity values interpolated from W. F. Ahtye [4]. Eventhough, the pressure effect is also significant; however, in this study the pressure at 66 Pa was set as a baseline.

The 1D MPD thruster simulation includes the sheath region, which is the interface between the plasma and the cathode regions. This sheath model [3] has been fully combined in the 1D simulation. That is, the sheath model calculates the heat flux and the sheath voltage by giving the temperature and the current density. This sheath model must be included in the simulation, as the sheath region is treated differently from the main plasma region.

For our 2D cylindrical symmetry simulation, the dimensions of the cathode, the anode, the total current, the pressure, the type of gases, the work function can be changed in the input process as needed for particular interested. Also, the sheath model is still included and fully integrated in this 2D cylindrical symmetry simulation at the cathode surface grids. In addition, the focus of the 2D cylindrical symmetry simulation is to connect the properties on the plasma and the cathode regions on the cathode surface until the MPD thruster reach steady state and estimate the plasma arc attachement edge, electroarc edge, on the cathode surface. Finally, we can understand more about the behavior of an MPD thruster under many different conditions of 2D cylindrical symmetry MPD thruster simulations.

Preface

President Obama approved NASA's budget for fiscal year 2015 and he reaffirmed the goal to send humans to Mars in the 2030s and to explore Jupiter's moon, Europa [8]. With more than 100 billion dollars invested in America's space program over the past six years including 17.5 billion dollars this year, the United States will remain the world's pioneer in space exploration for the time being. These ambitious explorations are the stepping stone approach for NASA space programs, as it is required to develop new crews, spacecraft, and new space vehicles carrying scienctific instruments for this long journey. As a result, a new advanced propulsion technology will be necessary to achieve these challenged missions. Such propulsion technology would need to have high efficiency and reduce propellant consumption.

Chapter 1

Introduction

Early in 2011, the Obama administration redefined the role of NASA and how it would achieve its objectives. With this new direction, NASA is developing two new rockets, one of which is Orion-a new crew module. By the 2030s, NASA hopes to send a manned spacecraft to orbit and land upon Mars-the Red Planet. To achieve this ultimate goal, NASA must develop advanced propulsion technology. One of the propulsion thrusters that has been identified as a prime candidate for in-space missions is the electrical propulsion (EP) thruster.

Chemical vs. Electric Propulsion Thrusters

Currently, chemical propulsion systems operate by lifting off (e.g., the Moon, the Earth, or an asteroid). Because they create a large amount of thrust force, such force can push against the gravitational force of planetary bodies. However, the chemical propulsion system provides lower exhaust velocity as well as lower specific impulse values by an order of magnitude when compared to Electic Propulsion (EP) thrusters. In a chemical propulsion system, the energy is contained in the molecular structural bonds. This energy is released through the combustion process and, as a result, is converted to kinetic energy in the converging-diverging nozzle, producing thrust. Thus, the amount of energy available is limited by the chemical bonds in the molecular structures of the propellant.

In an EP system, however, energy is produced by an external power source (e.g., solar arrays, batteries, or nuclear reactor), then transferred to the propellant, and converted into the converging-diverging nozzle to directed kinetic energy, creating thrust. This concept allows more energy to be introduced into the propellant and, as a result, EP have higher propellant-efficiency. For a higher efficiency system, this means less propellant is required to complete the same delta-V. In turn, a spacecraft can carry more cargo (e.g., people or scientific instruments) [9],[10].

EP thruster systems can be categorized as ion thrusters, arcjet thrusters, and magnetoplasmadynamics thrusters. The differences between these systems lie in the methods by which the propellant accelerates and how the energy transfers into the propellant. These two factors measure the propellant efficiency. Typically, these efficiency measures are determined by the specific impulse, I_{sp} (measured in seconds):

$$I_{sp} = \frac{F}{\dot{m}g_0} = \frac{\dot{m}u_e}{\dot{m}g_0} = \frac{u_e}{g_0} \tag{1.1}$$

where I_{sp} is the specific impulse (s), F is the thrust force (N), $\dot{m}$ is the propellant mass flow rate (kg/s), u_e is the exit velocity (m/s), and g_0 is 9.8066 (m/s^2). For each specific mission, there is a certain amount of velocity change (Δv) or, to put it another way, a certain amount of propellant mass calculated from the rocket equation. This amount of propellant can be calculated to achieve a certain delta-V from the rocket equation by:

$$M_p = M_o(1 - \exp^{\frac{(-\Delta v)}{I_{sp}g_0}}) \tag{1.2}$$

where M_p is the propellant mass, M_o is the initial mass, and Δv is the total velocity change to accomplish certain missions. As can be seen, for a certain delta-V, the higher I_{sp} means that a lower propellant mass is required. Recently, the cost of launching a payload to a lower earth orbit (LEO) has been estimated at approximately $10,000 per kilogram. To decrease the cost of a mission, thrusters need to perform more efficiently.

However, there is another problem. EPs require a longer amount of time to accelerate the spacecraft because they provide a relatively low thrust. That is, EP system component parts must last for thousands of hours, especially magnetoplasmadynamic (MPD) thrusters. This problem is due to the cathode erosion which occurs during charged particle interactions, which erodes its surface. This limits the lifetime of MPD thrusters.

The cathode erosion in MPD is due to its high operating temperature, which is in the range of 2,000-3,000K. The cathodes usually are made from tungsten or tantalum, which are refractory (high melting point) metals. There are two shapes of MPD cathodes, hollow and solid rod. The latter has been tested and developed for decades, so experimental data is more available. Mechanisms that create thrust are similar for both hollow and solid rod cathodes, the solid rod cathode has a much simpler geometry than a hollow cathode. As a result, this research will focus on developing a 1D and 2D cylindrical symmetry simulation to predict and characterize the attachment area of a plasma region and solid rod cathode region in a steady state of MPD thrusters. Information about an MPD thruster study can be read in references [6], [7], [10], [11], [12], [13], [14], [15], [16], [17].

Chapter 2

MPD Thruster

2.1 The Classification of an MPD Thruster

The classification of MPD can be considered in several categories as followed: the operating conditions, the shape of the cathode (hollow and solid rod cathode), the pulse or direct current mode and hot or cold cathode operation. The hot cathode operation is usually in a steady state operation while the cold cathode operation is usually in a start-up phase or pulse mode operation.

For this study, the MPD thruster uses direct current and it is assumed to have achieved the steady state condition. That is, there are only small changed in values of all the thermodynamic properties: the current density, magnetic field, and flow velocity of the gas values. For the shape of the cathode, this MPD thruster uses the solid shape cathode, which has a simpler design. In any case, this shape can be used to establish the fundamental physics for an MPD thruster. As a results, other more complex shapes such as multiple hollow cathode can also use this solid cathode as principal knowledge and adjust for needed applications.

2.2 The Solid Cathode MPD Thruster in Steady State

In this system, there are mainly two types of material-an anode (+) and a cathode (-), as can be seen in Fig. 2.1. A cylindrical solid cathode is placed in the center and surrounded by a cylindrical anode. The cathode is usually constructed from high melting point material such as tungsten or tantalum, as they have a low work function, which effectively provides the thermionic emission. The primary material for the anode is copper or tungsten, which can withstand high temperatures. However, at the anode, the heat load is not as high as at the cathode and the plasma attachment temperature is much higher than at the cathode. As a result, this study uses copper as an anode. Then, both anode and cathode are connected to high voltage and a high current power supply and the surrounding pressure or background pressure established by a vacuum pump can be as low as 6.75 Torr during operation at the highest flow rates of 640 sccm [2]. For the MPD thruster propellant in this study, argon is preferred as it is an inexpensive gas and a noble gas. That is, argon electron shells is fully filled so it almost never react

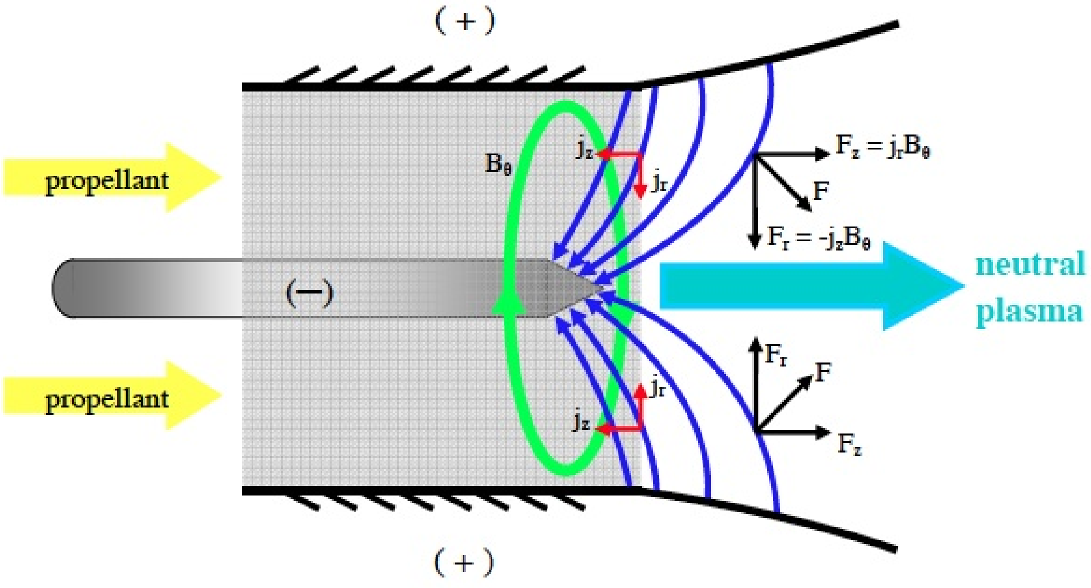

Figure 2.1: A Model of an MPD Thruster [2]

with other gases. That means it is extremely nonreactive and inert, even at very high temperatures.

Initially, the neutral argon propellant is symmetrically introduced from the cathode base and travels through the gap between electrodes. Then, the propellant is ionized by the arc discharge while traveling downstream. That is, one of the electrons is separated from the neutral propellant and the neutral propellant becomes a positively charged propellant. This flow of the electron and charged propellant is called plasma. The arc discharge is a form of current density path, which moves through plasma between electrodes. The electrons, the ions, and the neutral particles have different roles in the cathode surface interaction. That is, this interaction in steady state occurs once the cathode temperature is high and the variation of the temperature is minimal. The process where the thermal energy is given to the flow of charge carriers on the cathode surface and the charge carriers overcome the binding potential is called the thermionic emission. The thermionic emission depends on temperature and the electrical field and can be expressed in the Richardson-Dushman equation as shown below:

$$j_b = A_R T_c^2 \exp\left(\frac{-\varphi_{eff}}{kT_c}\right) \tag{2.1}$$

where j_b is thermionic emission current density, A_R is the Richardson coefficient 60 $A/cm^2/K^2$ for the tungsten cathode, φ is the material work function depending on temperature, k is Boltzmann's constant, T is the cathode temperature and φ_{eff} is the Schottky effect. This can be written as:

$$\varphi_{eff} = \varphi_0 - \sqrt{\frac{eE_c}{4\pi\varepsilon_0}} \tag{2.2}$$

where φ_0 is the reference work function (eV), e is the electron charge, E_c is the electric field surrounding the cathode, and ε_0 is the permitivity of free space. The variation of the electrical field changes the Scotty effect value and in turn the thermionic emission current density changes. For the current density, when it interacts with the magnetic field, a Lorentz force is created.

The Lorentz force $(J \times B)$ is a result of the right hand rule relationship from the interaction between the current density and the azimutal magnetic field around the cathode. The response of the current density in the axial or outward direction with a magnetic field is called the blowing force. The other force is called the pumping force, which is interaction with the magnetic field that is in the radial direction. The combination of these two forces generates a thrust in an MPD thruster.

2.2.1 Thermionic Emission

At the surface of the metal electrodes that operate in high temperature, the electrons are emitted to the plasma and this process is called thermionic emission. This emission determines whether the system reaches a self-sustaining point.

The thermionic emission can be related to the work function of the cathode surface. That means, in order to acheieve the thermionic emission rapidly, it would require a lower work function on the surface of the electrodes. The process that can lower the work function could be reached by covering the electrodes with monatomic film such as thorium and caesium. As a result, the film of thorium will be diffused to the surface of the tungsten at a high temperature. Particularly at the start-up phase, the electrons are emitted from the cold cathode surface and this emitting generates a micro spot, which is called the "cathode spot." The temperature of the cathode spot can reach as high as 3,000 K and it is the most destructive phase of cathode operation [17].

2.2.2 Thermal Excitation and Ionization

Ionization is the process whereby one or more electrons are removed from an atom. The energy exchange between particles is assumed to come from the thermal energy of the particles. This thermal energy or kinetic-energy is assumed to follow the Maxell-Boltzmann distribution, which is the most propable temperature in an ensemble of particles. The general Boltzmann distribution appears in exponential form as:

$$\exp\left(-\frac{\text{kinetic energy}}{kT}\right) \tag{2.3}$$

where k is Boltzmann's constant, T is temperature in Kelvin, and kinetic energy can be replaced by other forms of energy such as potential energy and chemical energy.

2.2.3 Fluxes and Transportation Properties

Uniform plasma is usually assume for many theoretical studies in plasma physics; however, this assumption is very difficult to produce. The actual characteristics of plasma will reveal gradients such as particle number densities, applied electrical potentials, temperature, and velocity components. These gradients can be considered driving forces that give use to fluxes and can be expressed by

$$\begin{aligned} \text{Fick's law } \bar{\Gamma} &= -D\nabla n \\ \text{Ohm's law } \bar{j} &= -\sigma_e \nabla V \\ \text{Fourier's law } \bar{q} &= -k\nabla T \\ \text{Frictional force } \bar{f}_x &= -\mu\nabla v_x \end{aligned} \tag{2.4}$$

where $\bar{\Gamma}, \bar{j}, \bar{q}$, and $\bar{f}_x$ refer to fluxes due to diffusion, electrical conduction, thermal conduction and the frictional force in x direction, respectively. The relation between driving force and fluxes is called transport coefficients D, σ, K_{th}, and μ, which are known as the diffusion coefficien, the electrical conductivity, the thermal conductivity, and the viscosity, respectively.

The particles collisions transfer the energy and momentum between particles. Thus, sufficient detail of the collision processes must known to determine the transport coeffcient, which depends on the collision cross section between particles. However, the exact electronic structure in a molecule or atom is difficult to determine experimentally, a highly simplified model such as a classic sphere has been developed for determining collision cross sections [18].

2.2.4 Local Thermodynamic Equilibrium (LTE)

In laboratories, LTE plasma is usually optically thin plasma, which does not require a radiation field that responds to the blackbody radiation. However, it does require a collision process that governs transitions and reactions in the plasma. In addition, LTE plasma has micro reversibility properties among the collision processes or a detailed equilibrium between each reverse process and collision process is necessary. In steady state, the solution of LTE and complete thermal equilibrium will yield the same energy distribution. Furthermore, the diffusion time should be on the same order of magnitude or larger than the equilibrate time and the gradient of LTE plasma properties (e.g., temperature and heat conductivity) should not be too large to diffuse from one location to another. This LTE plasma assumption is necessary to simplyfy the problem [18].

The following sections provide a literature review that explains studies of the surrounding plasma and cathode characteristics in an MPD thruster.

Chapter 3

Literature Review

The prior studies of MPD thrusters were performed at the University of Stuttgart, Germany. Now, this experiment and numerical simulations are being conducted in many universities and countries around the world.

3.1 Electrical, Thermal Conductivities and Heat Capacity Coefficients

Warren F. Ahtye. Ames Research Center, NASA, USA 1965 [4]
In early 1965, W. F. Ahtye calculated the basic transport coefficients of partially ionized argon by using the rigorous second-order Chapman-Enskog formulation. In addition, this method determined the electrical conductivity, translational thermal conductivity and heat capacity for fully ionized argon. As the results from second-order values compared with the simultaneous collision approach of Spitzer, the electrical and translational thermal conductivities are lower by a factor of three, as they include the ion-ion interactions. Furthermore, a comparison of the electrical conductivity values calculated by second-order Chapman-Enskog with experimental values shows that this second-order approach is accurate for calculating electrical conductivity and this approach is also valid for calculating thermal conductivity. Moreover, W. F. Ahtye studied the other thermodynamics properties (e.g., heat capacity of argon). His study provided a better understanding of relationship between pressure and temperature for each thermodynamic property.

Comments:

W. F. Ahtye's work calculates many properties: electrical and translational conductivities, and the heat capacity of argon. These values's plot can be used for our MPD thruster simulations as their values are in many different pressures. Our MPD thruster simulations require pressure as low as 0.5 torr (66 Pa). For such low pressure, it is difficult to determine acceptable values of electrical and translational conductivities, and the heat capacity. However, Ahtye's method calculates those values with many different pressures. As a result, these properties of argon, the electrical, thermal conductivities and the heat capacity, will be used in our 1D and 2D cylindrical symmetry MPD thruster simulations by interpolated method [19].

Daniel A. Erwin and Joseph A. Kunc 1985-1986 [20, 21]

In mid-1985, D. A. Erwin and J. A. Kunc determined some of the transport coefficients such as electrical conductivity, which depended strongly on the electron temperature (300 to 30,000 K) and degree of ionization (10^{-8} to 1) for monatomic gases (H, O, He, and Ne) in a weakly ionized or fully ionized plasma within the steady state condition. The electron distribution function, which is described by using the Boltzmann and the Maxwellian assumptions, is also outlined in this work. The elastic collisions, which can be electron-electron, electron-ion, or electron-neutral (e-e, e-i, or e-n), were considered in the plasma transport coefficient calculations, while the electron velocity distribution function was evaluated using the 4×4 matrix formulation method from Shkarofsky et al. In a low degree of ionization, there is a great discrepancy between different gases, as the electron-neutral collision tends to depend on gas types. They also mentioned the deviation of temperature between T_e and T_n from local thermal equlibrium (LTE), when high external electric field or density gradients existed; however, the deviation is suppressed by a low electrical fields and electron temperature at around 12,000 K.

The computer program was coded and calculated for the electrical conductivity of a magnetic- field free and partially ionized plasma where neither of the electron-neutral and Coulomb collisions were important. Although this method was available to calculate the electron temperature up to 38,000 K, the high dissociation of molecular gases could not be accurately determined. The computer codes had been separated into two main parts, which were physical routines and mathermatical routines. The electron temperature, gas pressure and the electron density were built into the computer program that calculated the electrical conductivity (mho/m).

Comments:

This paper provides a method for calculating accurately the result of the electrical conductivity of plasma and it can be used as a future work that links with [1] to determine the electrical conductivity of argon in an MPD thruster.

S. Paik et al. 1990 [22]

In the mid-1990, S. Paik and E. Pfender presented the argon plasma transport properties at low pressures (0.01atm). At this low pressure, the electron temperature deviated from the heavy particle temperature so the two-temperature particles were adapted to calculate the transport properties, such as electrical and thermal conductivities. These values with two-temperature effect and without two-temperature effect were compared. There is a large discrepancy at the low electron temperature ; however, these discrepancies become smaller when the electron temperature is around 10,000 K and higher.

In addition, the composition of particles was calculated from the Saha equation, which included the lowering of the ionization energy. From there, this calculation would be compared with experimental data: the momentum transfer cross sections. This method assumes the partially ionized collision-dominated gas to have the Maxwell-Boltzmann distribution. Their method all species (i.e., ion, atom, electron) interacted to create a cross-section collision. When the electron temperature becomes much higher than the heavy particles temperature, the two particles temperature effect needs to be included. Furthermore, the calculation can be calculated for the heat capacity composed of three parts (i.e., electron, reaction, and heavy particles). These three particles interacted with each other; however, the main contribution of the total value of the specific heat is the interreaction between electrons and heavy particles.

Comments:

The electrical and thermal conductivity values of argon at a pressure of 0.01 atm in S. Paik et al.'s work could not be used in this proposal because the pressure inside MPD thruster is much less than 0.01 atm. Moreover, we know the two-temperature effect starts at approximately 5,000 to 10,000 K; however, above 10,000 K the two-temperature effect becomes less important.

G. J. Dunn and T.W.Eagar 1990 [23]

G.J. Dunn and T.W. Eagar presented their own theory to calculate the electrical and the thermal conductivities of metallurgical plasmas at the ambient pressure. Their work was concerned with the metal vapor that changed the electrical and the thermal conductivity values in the plasma. At 1 atm, the assumption of local thermodynamic equilibrium (LTE), quasi-neutrality ($T_e = T_a = T_i$), and ideal gas was given for this calculation. Also, the second ionization became appreciable at $T > 17,000K$. These theories of this electrical conductivity depended strongly on the accuracy of the cross section collision theory. They compared the electrical and thermal conductivity values to other scientists such as Devoto and Cambel, and the comparison values were well within agreement.

Comments:

At ambient pressure, this paper provided the electrical and the thermal conductivity values of argon plasma. These electrical and thermal conductivity values can be used to compare with our argon plasma model at 1 atm. Also, we can apply it to our MPD thruster simulations; however, their work does not provide the electrical and thermal conductivity values as low as 0.5 torr (66 Pa). Therefore, it would not be possible to include these conductivity values for our MPD simulations.

3.2 Numerical Simulation of an MPD Thruster

E. Niewood et al. Massachusetts Institute of Technology, USA 1992 [5]

In mid-to-late 1992 at Massachusetts Institute of Technology (MIT), E. Niewood and his group developed a quasi-one-dimensional model of an MPD thruster that included a two-fluid flow. That is, the two fluid flow consisted of heavy particles and electron particles. Their model used the energy conservation equation and the momentum equation. These two equations colculated the ion and the neutral particles separately. In addition, the governing equations included global continuity, global momentum and electron density, all of which were dominated by Coulomb collisions. In the energy equation, the main source terms were from Joule heating (Ohmic heating), elastic transfer between electron and heavy particles, energy lost from ionization, and the axial heat conduction.

From this work, the temperature of the heavy species was a magnitude lower than the electron temperature in which the two fluid temperatures were necessary to have accurate electrical or thermal conductivities or viscosity coefficients, and also to determine the effect of plasma instability. In addition, the other phenomena, such as the area variation of the channel and velocity slip between ions and neutrals, were considered and then compared to the experimental data.

Comment:

To compare the numerical result and experimental value at one point, it is necessary to increase the voltage by 45 V to match the experimental data. This result confirms that the cathode sheath, fall voltage, and the real value of the conductivity are important in order to understand more about an MPD thruster.

C. K. J. Hulston et al. College of Engineering, India 1994 [6]

In mid-to-late 1996, Hulston et al. conducted a one-dimensional numerical simulation of cathode thermal erosion. He used the energy equation: conduction, convection, radiation and ohmic heating. Then, the equations are discretized in terms of deforming finite elements treated as ablation at the surface. Following the Galerkin method, the heat conduction was obtained in the form of a semidiscretized differential equation. The time step was increased until the system reached steady state and the time step for each element was within the allowable stability.

Then, the temperature profiles were obtained along the cathode by comparing the numerical results to verify his model. The erosion rate or material loss of the cathode was calculated by the convective matrix after the temperature was more than ablation temperature.

Comments:

Hulston, et al.'s work assumed constant thermo-physical properties such as electrical, thermal conductivities and heat capacity of the material. In addition, if this work had included the temperature dependent of the heat flux at the cathode root, this work would have improved the cathode thermal erosion simulation. For the fundamental relationship of an MPD thruster, it was found to be at a higher temperature in the conical portion of the cathode and the Ohmic heating produced a higher amount of heat generation in the conical region of the cathode than in the cylinder region of the cathode.

K. D. Goodfellow University of Southern California, USA 1996 [3]

In mid-1996, at the University of Southern California, Goodfellow conducted an experiment and created a quasi-one-dimensional numerical model. This model required a thermal model and near-cathode plasma model in a steady state for an electrical thruster to predict the lifetime of the cathode. The thermal model determined the characteristics of the temperature profile in the cathode and the near-cathode plasma model calculated the characteristics for heat flux and current density. The near-cathode plasma was represented as a transition region between the plasma region and the cathode surface. This near-cathode plasma included the sheath region, and boundary layers, and was coupled with the thermal model. A series of operating condition experiments was conducted to verify the numerical model values. Then, the temperature profile of the cathode was observed under different conditions.

Comments:

Goodfellow's work included the sheath region, which was the most important model to determine the voltage drop. This fundamental knowledge of the particle interaction between electron and atom particles provided more understanding about the MPD thruster. The arc discharge area on the cathode had been measured from the experiment and put into the numerical model. However, if the arc attachment area model had been included, his numerical model could have fullfilled the understand of the MPD thruster theory on the cathode surface. Further development of the arc attachment area for the MPD thruster model is required and this topic is in this dissertation.

J Rossignol et al. University Blaise Pascal, France 2003 [7]

J Rossignal et al. introduced a one-dimension model of the cathode sheath in an electrical arc. Heavy particles interacting above the sheath edge was introduced as a friction zone derived from an ion-atom collision. This model determined the heat flux, the electrons, and density of atoms in a steady state. Overall, his work is divided into the physical process and the modeling. The physical process describes the cathode's rise in temperature derived from the plasma bombardment (i.e., ions, electrons, atoms, radiation) on the surface and the Ohmic heating in the cathode, and the energy dissipation consisted of thermal conduction droplet ejection and surface vaporization from the cathode. The sheath region was introduced as the trasition zone between the plasma and the cathode. The emitted electron could be accelerated outward by the voltage difference. Some of the newly created ions then fall into the voltage drop region and bombard the surface of the cathode.

In the surface and sheath model, there are many interactions, consisting of Bohm criterion velocity, the ionic energy bombardment to the cathode surface, radiation in the plasma and the cathode surface, and the bombardment of the electrons from plasma and the returning electron to the cathode surface. In addition, the electric field, atomic emission, bombardment and ionic friction were included in this model. Next, he uses a coupling of the cathode heating model and sheath model to solve for the energy balance between the energy flux of vaporization of the cathode and the energy flux due to the thermal conductivity.

Comments:

This work does not explain the properties of the plasma region connected to the friction zone. However, the basic detail of each particle interaction is explained well and the results of the numerical simulation model obtain a realistic estimation of characteristics for heat flux, current density and particle density.

H. Kawaguchi Hokkaido et al. University, Japan 1995 [24]

In early 1995, H. Kawaguchi presented the numerical study of the thrust mechanism in a two-dimensional self-field MPD thruster. Two types of thrusters, which were flared and converge-diverge (C-D), had been studied and compared with experimental data. This work assumed the quasi-steady state, a fully ionized one-fluid argon plasma, and a constant of electrical conductivity, and neglected the voltage drop across the plasma sheath. This numerical simulation used the total variation diminishing (TVD), successive over relaxation (SOR) method and iterated until the electromagnetic and fluid equation reached self-consistency. For convenience, the curvilinear coordinates were used instead of the Cartesian coordinates.

The numerical simulation results show that a shock wave occurred near the tip of the cathode and the fluid became supersonic near the inlet in the flared type. In the C-D type, the fluid became supersonic at the throat. The shock wave and braking force were the main suppression in the flared type and C-D type respectively for the aerodynamic force in MPD.

Comments:

Kawaguchi's comparison between numerical results and the experimental data provides fundamental knowledge about the relationship between the thrust and discharge current. In addition, microscopic phenomena can be observed in this numerical simulation. However, if the electrical conductivity temperature dependent in argon plasma had included in the plasma region, his study might have delivered the relationship between the voltage drop and temperature. Finally, the fact that voltage sheath drops across the cathode, which is the main power loss in the MPD thruster, has not been included in this study.

T. Miyasaka et al. Nagoya University, Japan 1998 [25]

In early 1998, T. Miyasaka et al. studied the Numerical Analyses of 2-Dimension Axisymmetric MHD Flow Satisfying Sonic Conditions in an MPD Thruster. This work explained the two-temperature effect, ionization, recombination, and onset phenomenon to the straight-configuration electrodes in a self-induced MPD thruster. Argon was the propellant gas, which assumed a sonic speed at the inlet (argon plasma is a perfect gas having constant specific heats).

The method used was a second-order accurate explicit total variation diminishing (TVD) upwind scheme to solve the electromagnetic and compressible Euler equation in steady state. The pressure and current contours showed that the strong Lorentz force, which came from the concentration of the discharge current, reduced the pressure from downstream of the inlet. The force was slowly increased with the current (i.e., the value of thrust was approximately 8 N at 10 kA).

Comments:

This paper explains the relationship between thrust and the discharge current; however, the cathode sheath phenomena has not been included in the model. The cathode sheath is considered repondsible for the voltage drop across the electrodes in the MPD thruster.

Chapter 4

Prior Work

4.1 Introduction

In the recent years, the advancements in graphics and processors technology have allowed the computational simulation to solve the nonlinearity problem because of its accuracy and sophisticate. In addition, there are several benefits i.e. it allows scientists and researchers to see how the systems respond. Also, it can help detecting wrong calculations design, which can save a lot of money before developing and building experimental set up.

Furthermore, the physical intuition can be tested much more quickly and inexpensively than an experiment. Moreover, the computational simulation can be repeated to investigate the different conditions, once the correct computational simulation model has been verified.

The MPD thruster has several different characteristics and these differences are required to be included in the computational simulation for the high-current solid rods cathode and the surrounding plasma in MPD thruster.

4.2 The Role of MPD Numerical Simulation

The MPD thruster model consists of the following parts: computational grid, solid cathode model, plasma model, cathode sheath.

The computational grid is a system of cells covering the model geometry, the geometry will be axisymmetric, and the cells will be triangular. This triangular grid system is based on Winslows method [1], which provides many advantages i.e. the triangular grid can generate the curvature at the interface between two different material or at the corner much more uniformly than rectangular grid. Solid cathode model is a set of equations describing flow of heat and electric current in the cathode. The plasma model is a set of equations describing flow of partially ionized argon through the model, and the flow of heat and electric current. The cathode sheath is a model describing the interface between the cathode and the adjacent plasma using [3].

The goal of this dissertation are to develop 1D and 2D cylindrical symmetry simulations of an MPD thruster. These simulations consist with 3 mains regions which are the plasma region, the cathode region, and the plasma sheath region (on the cathode surface [3]). Then, the electrical conductvity, the thermal conductivity and the heat capacity are applied into the plasma and the cathode regions until 1D and 2D cylindrical symmetry simulations reach steady state to describe the temperature, potential, electric field and current density in the simulations.

That is, 1D MPD thruster simulation will be developed first to obtain the fundamental physics of an MPD thruster and to calculate to obtain the temperature, the voltage in the cathode and the plasma regions. Also, the voltage drop (sheath voltage) between the cathode tip and the plasma can be evaluated. Then, the 2D cylindrical symmetry MPD thruster simulation calculates temperature, potential, the current density, the electric field to fully estimate the electroarc edge or plasma arc attachement edge on the cathode surface. Further development to include the particles flow of electrons and ions could improve the cathode erosion simulations as describe in [3] as shown in Fig.4.1. Some of the references for these simulations are listed as [5], [14], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42].

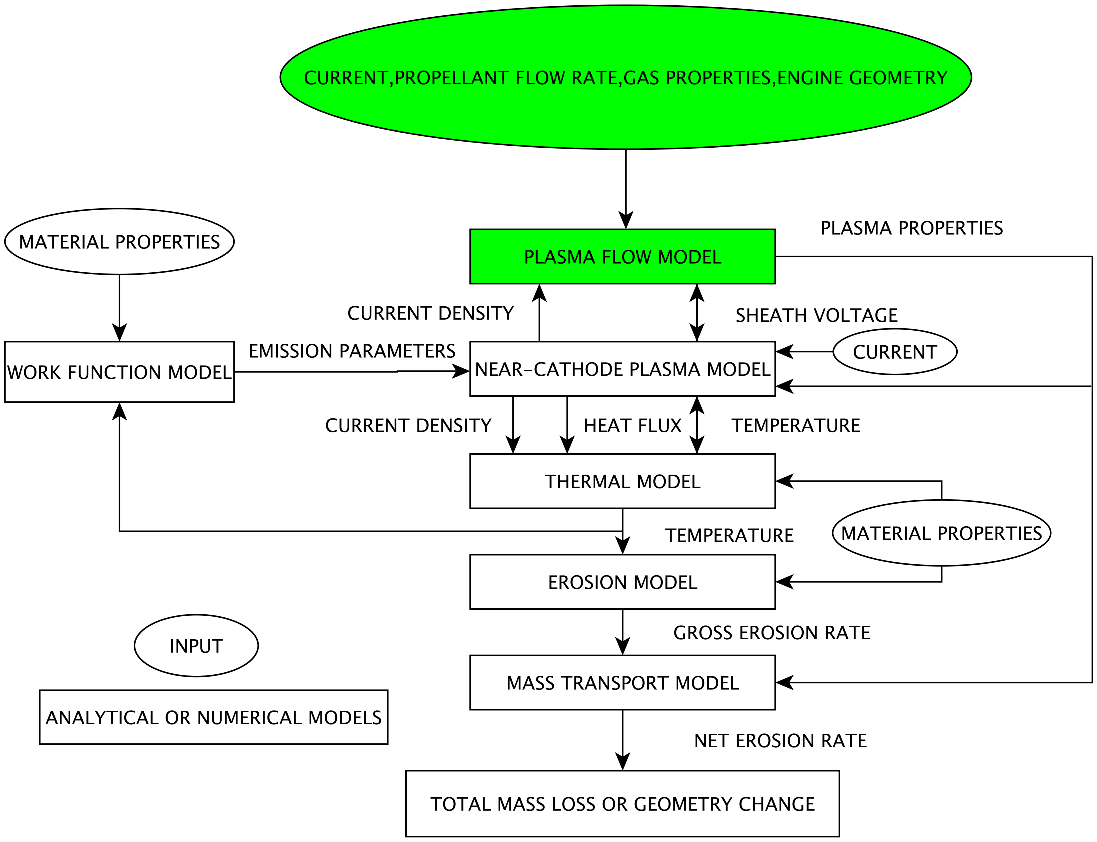

Figure 4.1: Diagram of the cathode erosion model [3].

4.3 1D MPD Thruster Simulation

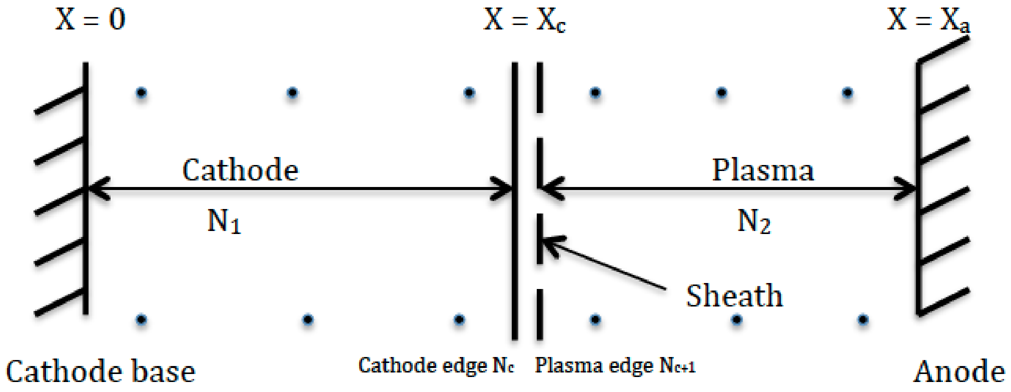

Figure 4.2: The one-dimensional cathode-plasma system with magnetic field outward direction from the page.

4.3.1 Introduction

This section describes a one-dimensional cathode-plasma model. The model is designed to find the steady-state behavior of a system consisting of a cathode, a plasma, and an enclosing electric circuit that forces a specified current density into the system see Fig. 4.2.

Following are the effects modeled: There are an electric and thermal conduction within the cathode and the plasma regions. Also, the joule heating (Ohmic heating) are in both cathode and plasma regions. Then, the voltage drop (cathode fall) can be calculated using the model and apply at the anode toward the cathode as the anode voltage is given as 0 V and assume no magnetic field interaction as it is 1D simulation.

4.3.2 Grid definitions

This section describes the computational grid. There are two regions, the cathode and the plasma regions, as can be viewed in Fig.4.3. The N_1 and N_2 define as the number of grid in each region and the L_c and L_p define as the total length in the cathode and the plasma regions, repectively. There is one spatial variable x, which ranges from 0 at the cathode base to x_A at the anode. The grid is composed of N + 1 points numbered 0 to N. Point 0 is the cathode base; point N_C is at the cathode edge; point N_{C+1} is at the plasma edge; and point N is at the anode. The sheath is located between points N_C and N_{C+1}. These two points are very close together and may in fact be collocated, since the sheath model assumes the sheath to be of infinitesimal thickness. Grid point i is located at x = x_i . Thus, $x_0 =$ 0 and $x_N = x_A$. Two kinds of cells are defined. Primary cells span the intervals between grid points. There are N primary cells numbered from 1 to N. Primary cell 1 covers the interval between point 0 and point 1; more generally, primary cell i covers the interval between point i-1 and point i. The width of primary cell i is

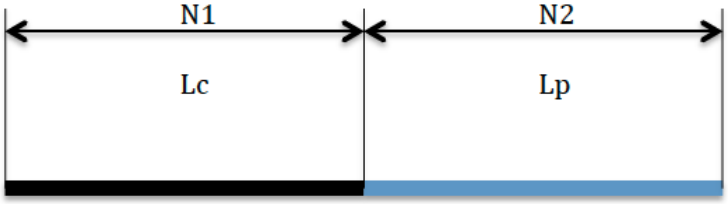

Figure 4.3: 1D cathode N1 and N2 are the cathode and plasma regions.

$$L_i = x_i - x_{i-1} \tag{4.1}$$

Secondary cells span the intervals between the midpoints of primary cells. There are N+1 secondary cells numbered 0 to N. The secondary cell numbered i surrounds grid point i . Note that if the primary cells adjacent to grid point i are of different widths, then grid point i is not the midpoint of secondary cell i . The width of secondary cell i is

$$L_{S,i} = (L_i + L_{i+1})/2 \tag{4.2}$$

The two secondary cells at the grid edges are bounded by the grid endpoints. Thus, secondary cell 0 ranges from x = 0 to x = $L_1/2 = x_1/2$, and secondary cell N ranges from $x = (x_{N-1} + x_N)/2$ to $x_N = x_A$. As you can see in Fig. 4.4, it is the example of the primary cells, secondary cells, primary grid points and secondary grid points.

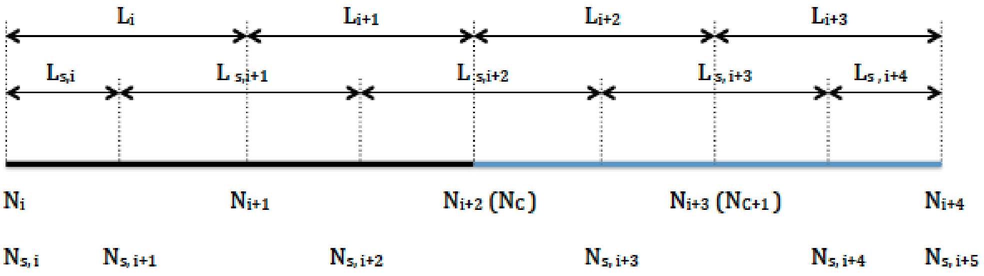

Figure 4.4: 1D cathode numerical grid cells.

4.3.3 Equations to be solved

4.3.3.1 Electrical Potential

The external circuit pumps current density J_0 through the system, flowing from anode to cathode. Since the system is one-dimensional, the current density is constant throughout the system. The electrical potential ϕ varies from 0 at the anode to $\phi_C < 0$ at the cathode base. The electric field is

$$E = -\frac{d\phi}{dx} \tag{4.3}$$

The current density is

$$J = \sigma E \tag{4.4}$$

where σ is the electrical conductivity. (Note that both the electric field and the current density are negative, since the field points from right to left.) The electric field is found from requiring that the current density found from eq. 4.4 is the specified value:

$$\sigma E = -J_0 \tag{4.5}$$

where the specified current density J_0 is taken to be a positive number. Assumed no magnetic field interaction in this study for 1D and 2D cylindrical symmetry simulations; however, the magnetic field can be included in the future work for 2D cylindrical symmetry simulation.

4.3.3.2 Temperature

We solve the one-dimensional thermal diffusion equation,

$$c_p \frac{dT}{dt} = JE + \frac{d}{dx}\left(K \frac{dT}{dx}\right) \tag{4.6}$$

where c_p is the heat capacity per unit volume, JE is the ohmic or Joule heating, and K is the thermal conductivity.

4.3.3.3 Cathode sheath

The characteristic of near-cathode model from Goodfellow's model is the characteristic between plasma and cathode in steady state and considers the sheath to be a discontinuity between cathode and plasma. The model predicts the heat flux (q_{tot}), current density (j_{tot}), electron number density, and electron temperature (eV) as a function of temperature and pressure at the near-cathode surface.

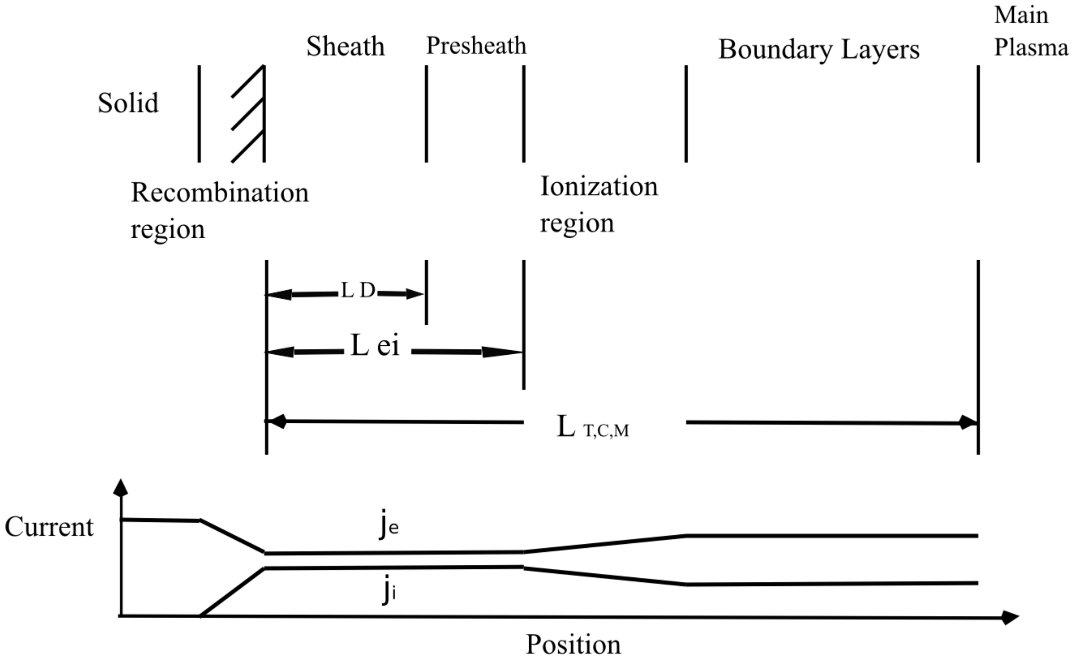

Figure 4.5: Near Cathode Plasma Region [3].

The model of near-cathode model consists many properties regions such as solid, recombination, sheath, presheath, ionization, boundary layers and main plasma region. An illustration of near cathode plasma region is shown in Fig. 4.5. The detail of the characteristic of near-cathode model can be read in reference [3] where L_D, L_{ei}, and $L_{T,C,M}$ are the Debye length, mean free path, and thermal, concentration and momentum boundary layer thickness, respectively. (j_e, j_i) are the relative magnitude of electron and ion current density with the region or position.

The mole fraction of argon varies with the electron temperature and the second ionization starts around electron temperature at 1.1 eV as can be seen in Fig.4.6. The heat flux and the current density of each species at the interface between the cathode and the plasma regions with work function of 4.5 eV and pressure of 66 Pa can be calculated as in Fig.4.7 and 4.8.

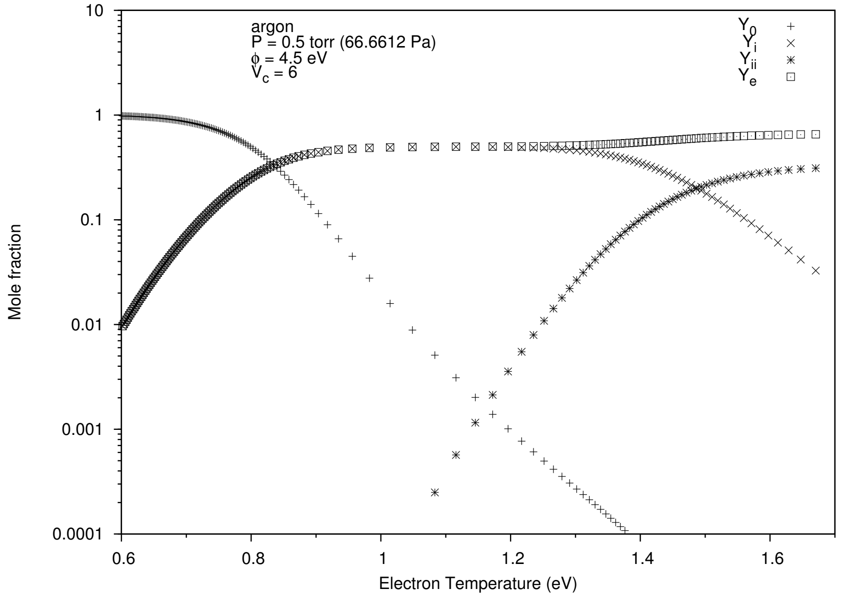

Figure 4.6: Mole fraction species as functions of electron temperature.

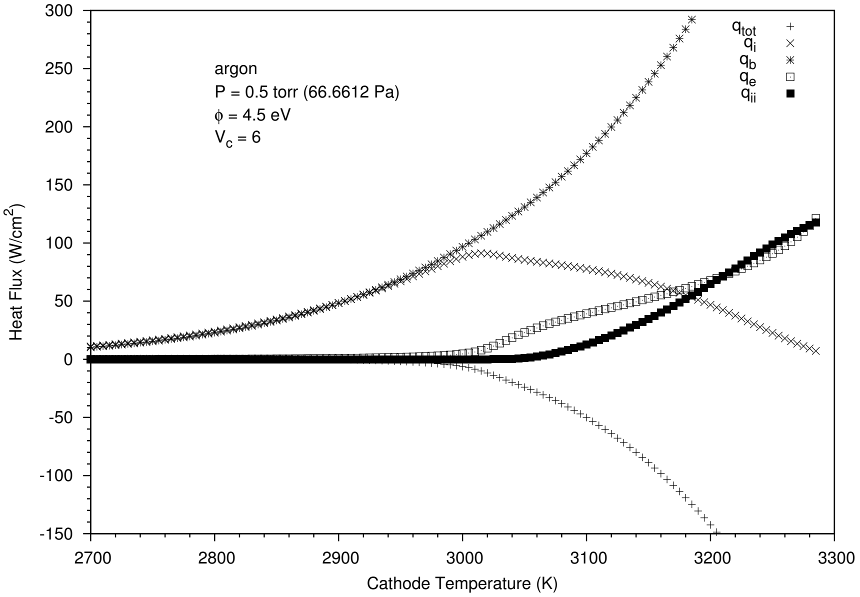

Figure 4.7: Magnitude of heat flux components as functions of cathode surface temperature.

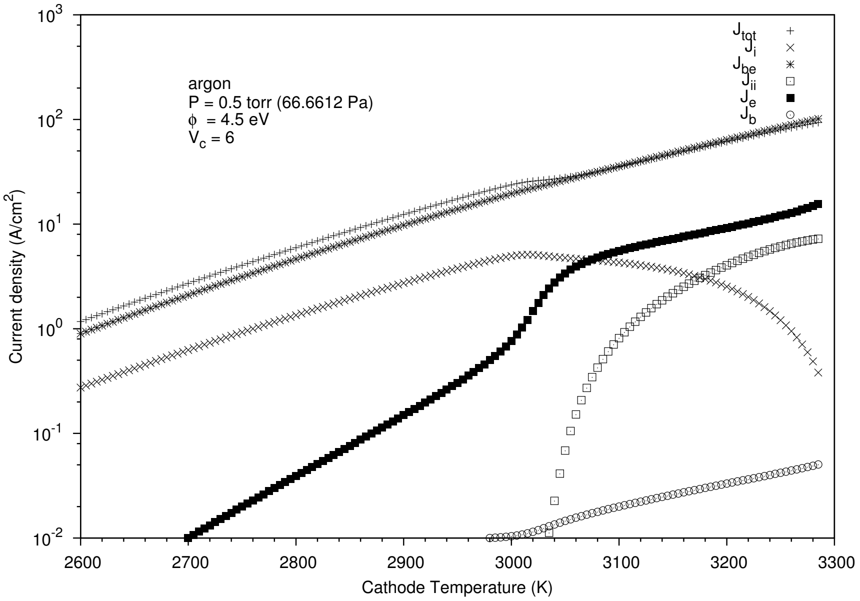

Figure 4.8: Magnitude of current density components as functions of cathode surface temperature.

As can be seen in the Fig.4.7, the total heat flux consists of many particles as shown in the equation below:

$$q_{tot} = q_{i,1} + q_{ii,1} - q_{n,1} + q_{i,2} + q_{ii,2} - q_{n,2} - q_b + q_e \tag{4.7}$$

$$q_{tot} = \sum_{s=1}^{2} \left(\frac{j_{i,s}}{e}(eV_e + eV_B + \epsilon_{i,s} - \phi_{eff}) + \frac{j_{ii,s}}{2e}(2eV_c + eV_B + \epsilon_{ii,s} - 2\phi_{eff} \right)$$

$$- \sum_{s=1}^{2} [F_{n,c,s} 2kTc] - \frac{j_b}{e}(\phi_{eff} + 2kT_c) + \frac{j_e}{e}(\phi_{eff} + 2kT_c)$$

where $q_{i,1}, q_{i,2}$ represent the energy gain by singly-charged ion, $q_{ii,1}, q_{ii,2}$ are the energy gain by doubly-charged ion, $q_{n,1}, q_{n,2}, q_b$ are the thermal energy removed by the neutrals, and thermionic electron, q_e is the energy gain from plasma electrons, and k is Boltzmann's constant expressed in units of eV/K.

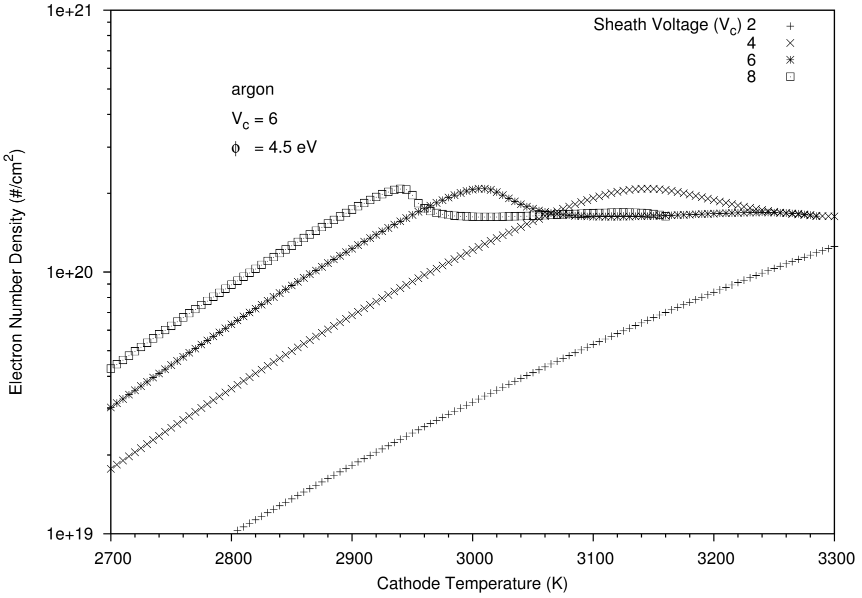

Figure 4.9: Electron number density as a function of cathode surface temperature with sheath voltage as a parameter.

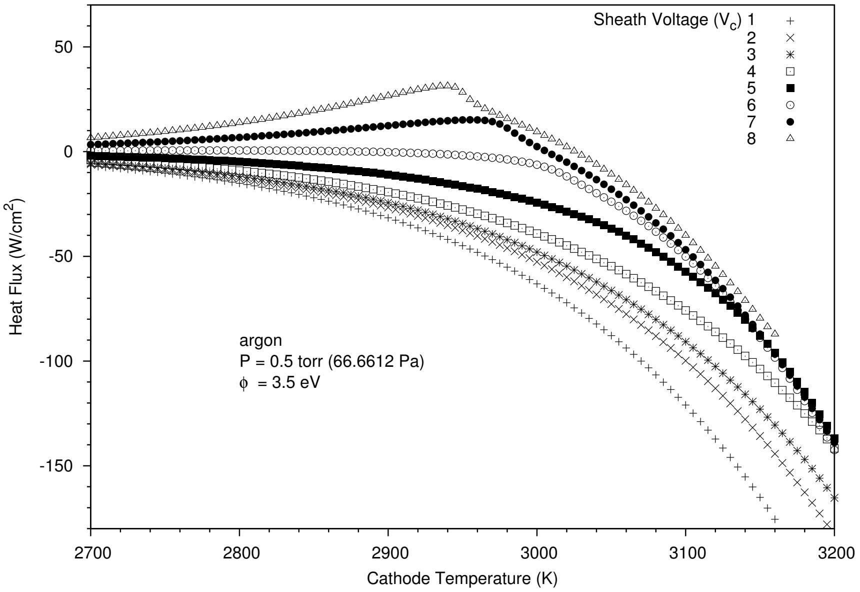

Figure 4.10: Total heat flux as a function of cathode surface temperature with sheath voltage as a parameter.

The net current on the cathode surface consisted many species as in Fig.4.8 and the expression can be given as

$$j_{tot} = j_{i,1} + j_{ii,1} + j_{i,2} + j_{ii,2} + j_b - j_e \qquad (4.8)$$

where $j_{i,1}, j_{i,2}, j_{ii,1}, j_{ii,2}, j_b$ are the current density gained by singly-charged, boubly-charged ions, and thermionic current density. j_e is the plasma electron current density lost.

The electron temperature, the heat flux, the current density and the electron temperature of work function of 4.5 eV and pressure of 66 Pa will be considered as a standard case. The effect of different sheath voltage can be seen in Fig.4.9 - 4.12. The effect of different work function can be seen in Fig.4.13 - 4.16. Then, the pressure effect can be viewed in Fig.4.17 - 4.20. The discussion of each effect can be studied in [3].

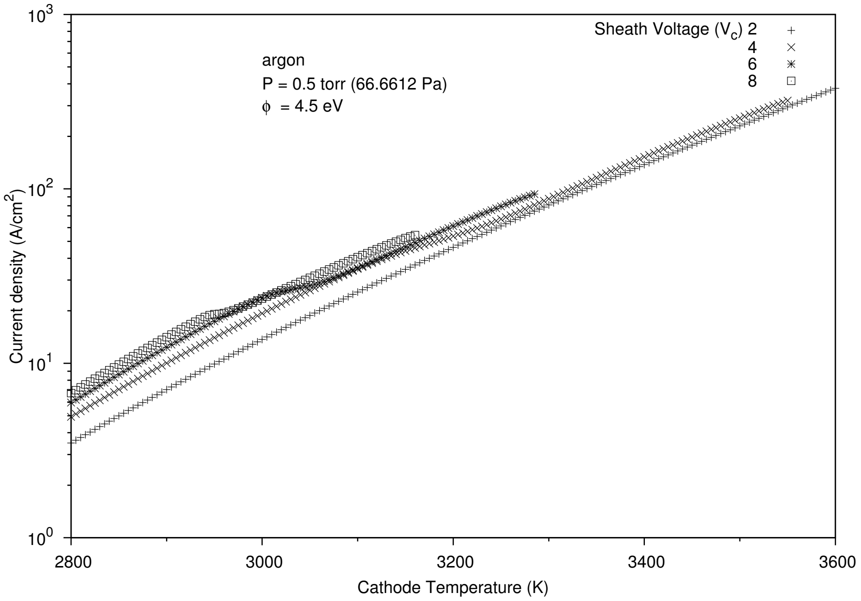

Figure 4.11: Total current as a function of cathode surface temperature with sheath volatge as a parameter.

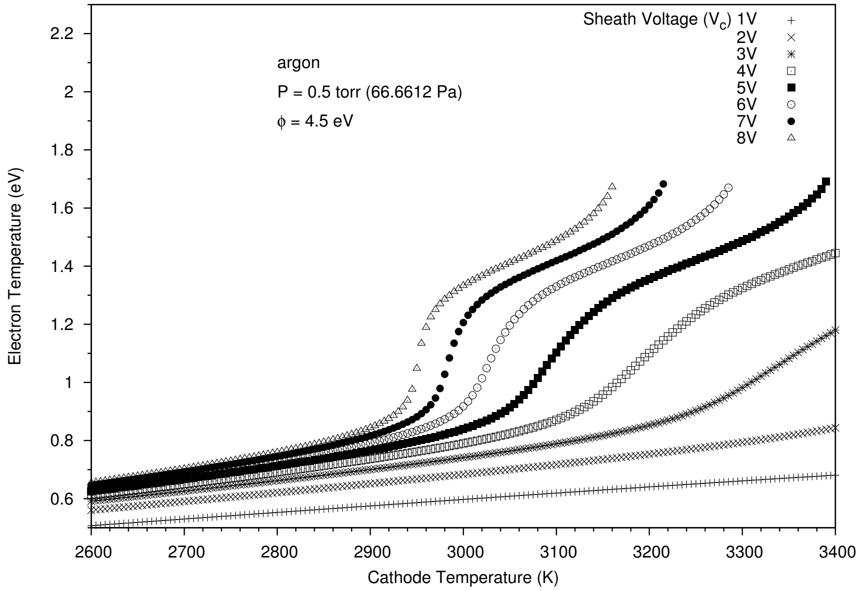

Figure 4.12: Electron temperature as a function of cathode surface temperature with sheath voltage as a parameter.

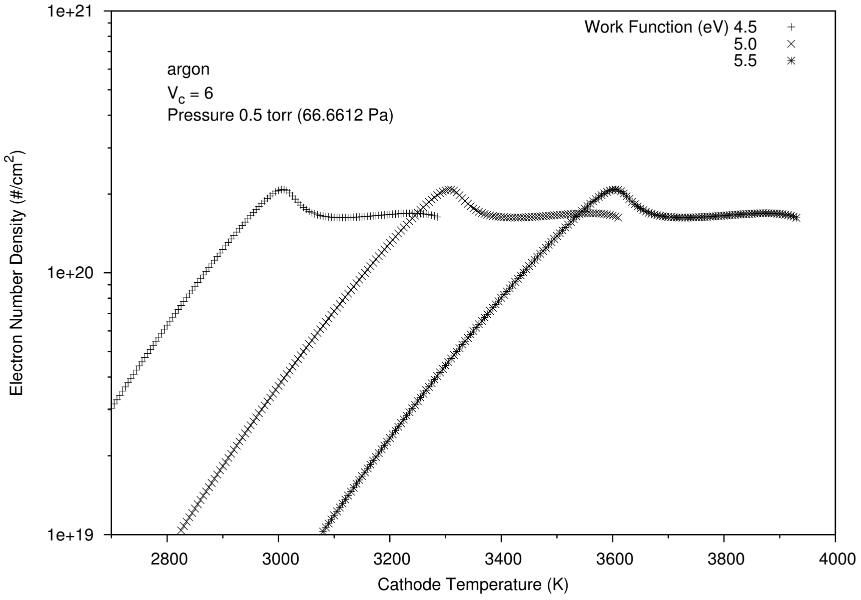

Figure 4.13: Electron number density as a function of cathode surface temperature with work function as a paramter.

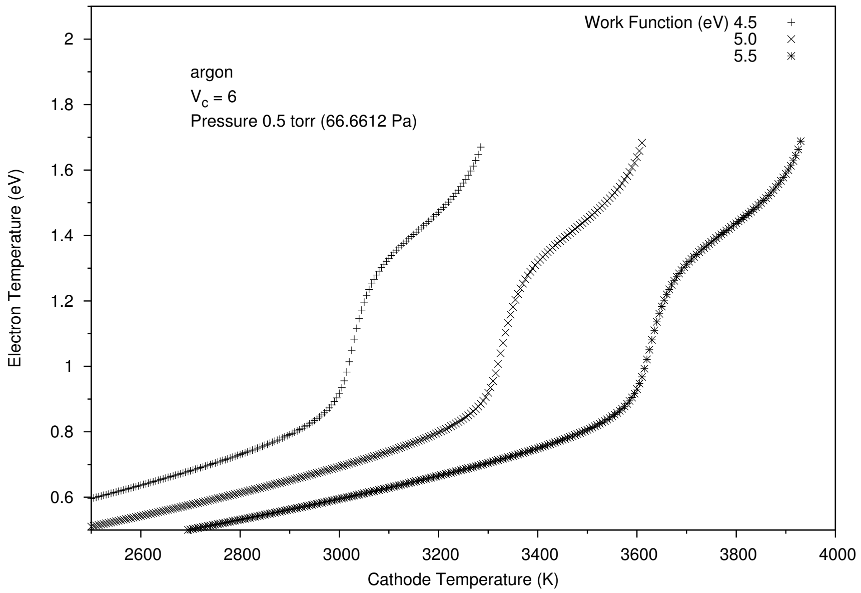

Figure 4.14: Electron temperature as a function of cathode surface temperature with work function as a parameter.

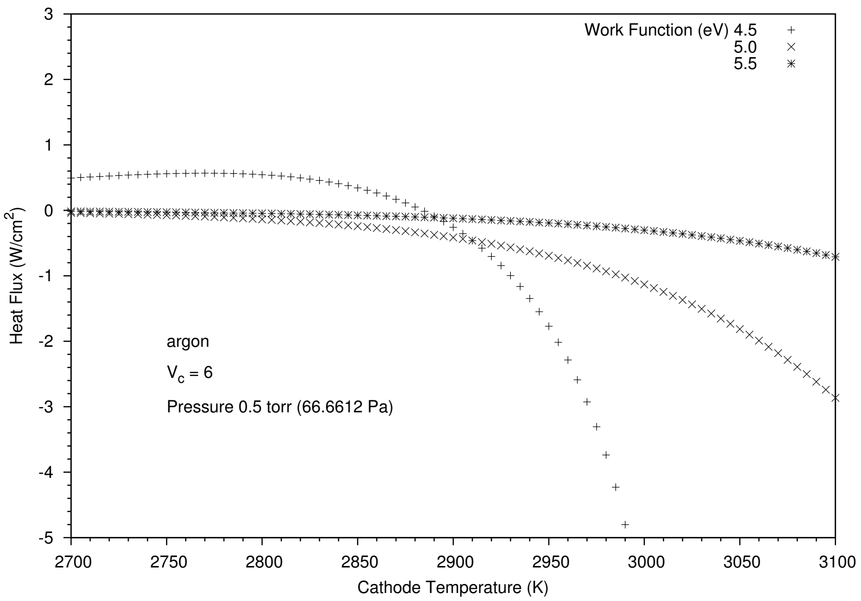

Figure 4.15: Total heat flux as a function of cathode surface temperature with work function as a parameter.

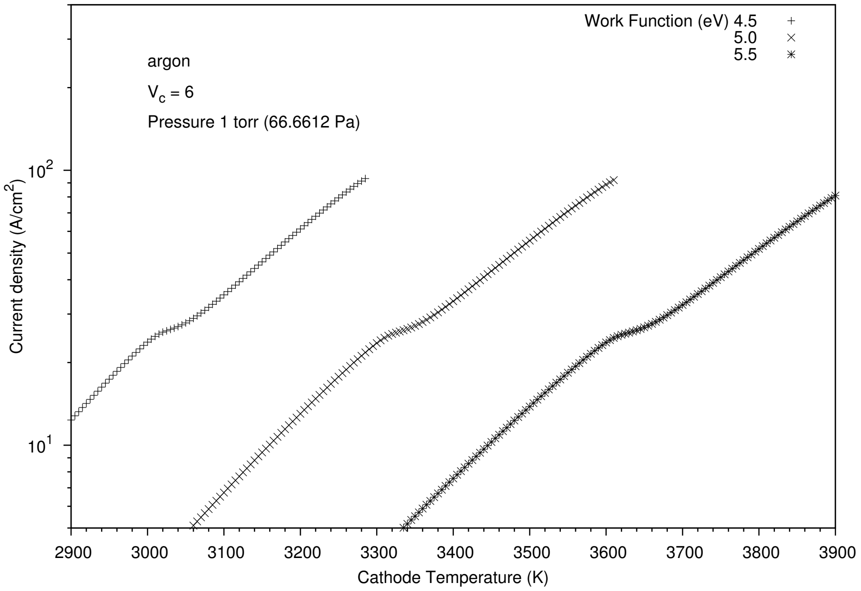

Figure 4.16: Total current density as a function of cathode surface temperature with work function as a parameter.

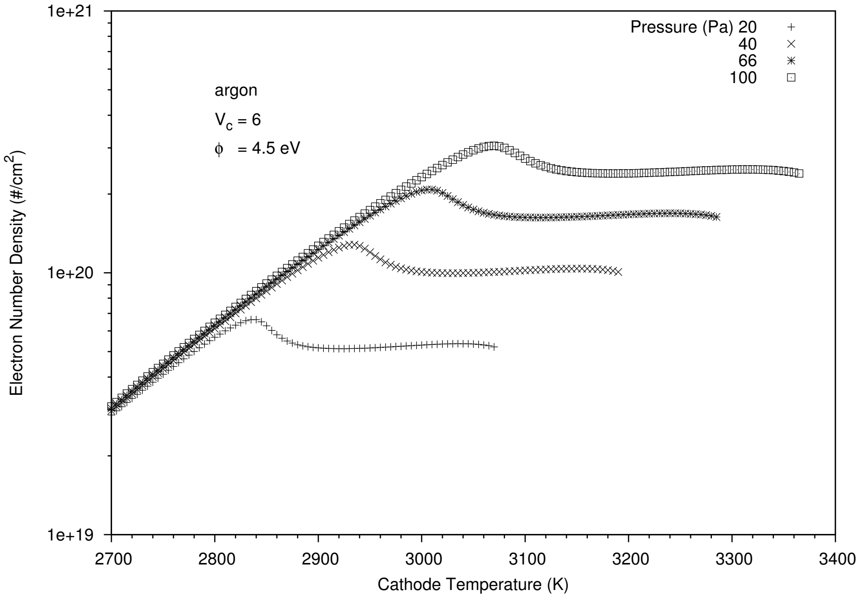

Figure 4.17: Number Density as a function of cathode surface temperature with pressure as a parameter.

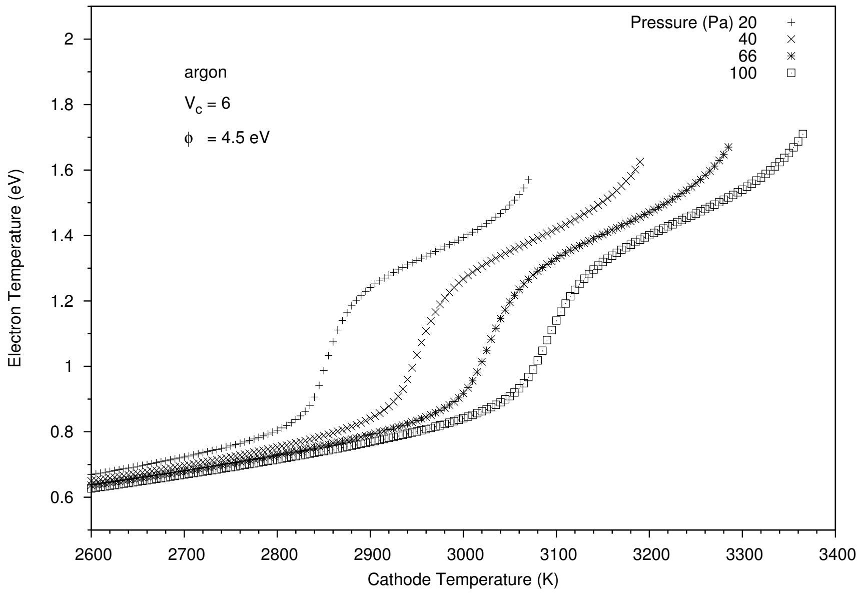

Figure 4.18: Electron Temperature as a function of cathode surface temperature with pressure as a parameter.

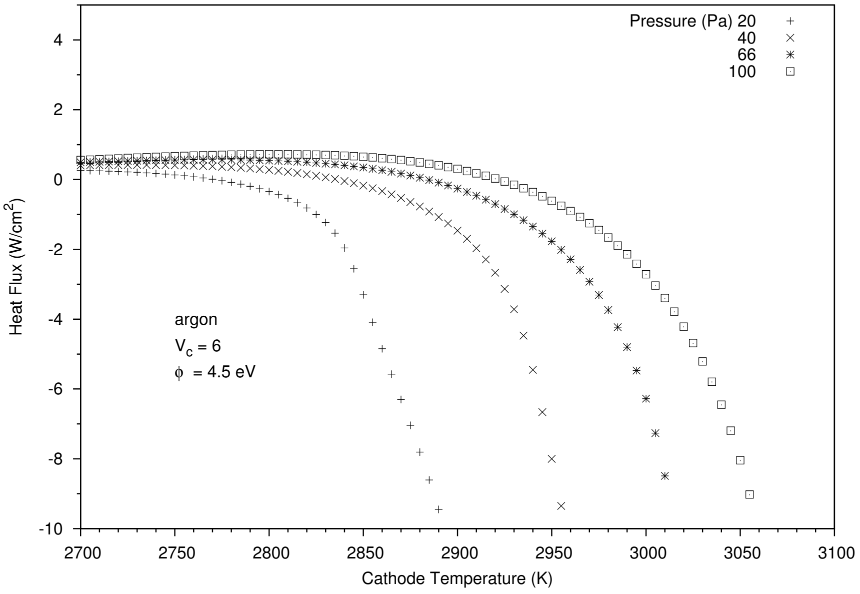

Figure 4.19: Total heat flux as a function of cathode surface temperature with pressure as a parameter.

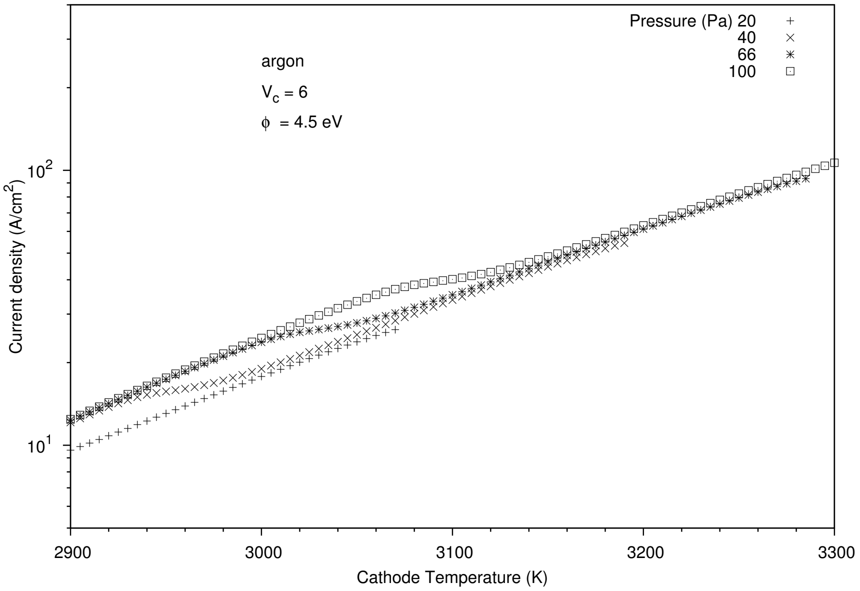

Figure 4.20: Total current density as a function of cathode surface temperature with pressure as a parameter.

4.3.3.4 Boundary conditions

At the base of the cathode, the temperature is fixed at T_{base} or T_0, with a typical value of 500 K or 1500 K. (An alternative boundary condition can be specified as needed to compare with the experimental data.) At the anode, the simplest condition is to set the plasma temperature to a fixed value $T_{p,A}$, for which a typical value would be chosen by the condition that the degree of ionization obtained by the Saha equation be 10^{-2} to 10^{-1}. (Again, an alternative boundary condition can be obtained to analyze with the experimental data.) The plasma pressure is considered to be constant at a value P, for which a typical value is 0.5 Torr (66 Pa).

4.3.3.5 Transport Coefficients

For 1D and 2D cylindrical symmetry simulations, the electrical conductivity, the thermal conductivity and the heat capacity which are temperature and pressure dependent, are required. However, in this work, pressure is at 66 Pa only but can be adjusted as needed. The electrical conductivity of argon plasma changes significantly as the temperatue changes. This temperature effect must be included to fully obtain the realistic MPD thruster simulations.

Again in order to calculate the transport coefficients, the electrical and the thermal conductivity values, it is require to understand the concept of carriers in transport coefficients. There are three types of carriers in plasma neutrals, ions, and electrons. The particles collision among carriers can be seen in Table 4.1. At high temperature, there are relatively more charged particles and thus the electron particles play an important role in the collision to neutral and charged particles. The table below shows various particle collision for different types of transport phenomena [43]. There are many valuable references about transport coefficients [18], [22], [23], [44], [45], [46], [47], [48], [49], [50], [51].

Table 4.1: Various types of transport processes

Transport process	Controlling modes of collisions
Electrical conduction	Electron-neutral, Electron-ion
Thermal conduction	Electron-electron, Electron-neutral,
Thermal conduction (con't)	Ion-Neutral, Electron-ion
Viscosity	Neutral-neutral, Ion-ion, Ion-neutral
Diffusion	Electron-ion, Ion-neutral

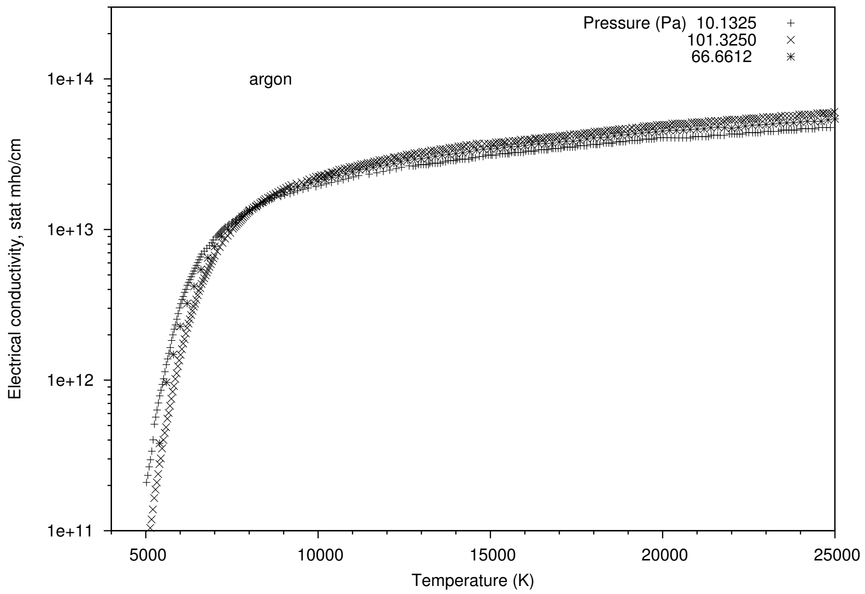

Figure 4.21: Translational electrical conductivity of argon as a function of temperature with various pressure [4] using interpolation method.

Most of the fully or high ionization plasma are presented in the universe but hardly found in laboratory. However, the partially ionization, which is in between fully and low ionization region, are the most useful in engineering applications and researches. For the argon plasma, only the translational electrical conductivity portion is important because argon's electron shells are filled so it has no vibrational portion. As it can be seen in Fig.4.21-4.22, the translational electrical conductivity and translational thermal conductvity from [4] are a function of temperature values with various pressure as a parameter.

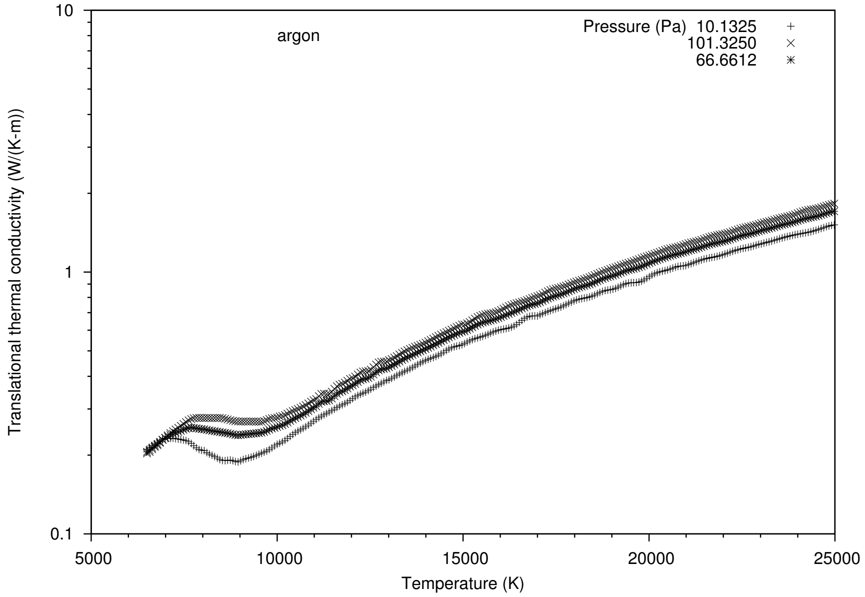

Figure 4.22: Translational thermal conductivity of argon as a function of temperature with various pressure [4] using interpolation method.

As can be seen, the electrical conductivity and thermal conductivity depend on temperature and pressure. However, the electrical conductivity increase exponentially and slowly incresed as temperature rises for various pressure. While the thermal conductivity at low pressure has a kink at temperature below 12,000 K and then steadily increased.

4.3.3.6 Heat Capacity

The thermodynamic properties can be purely calculated from the statistical mechanic. The statistical mechanics describe the thermodynamics properties from the atomic and spectrum [52].

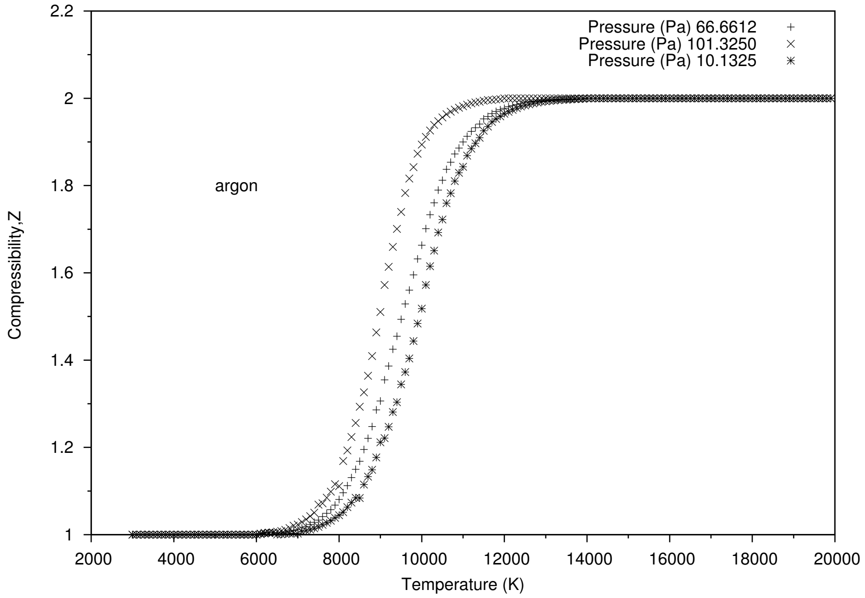

Figure 4.23: Compressibility of argon as a function of temperature [4].

For this study, the heat capacity will be interpolated from [4] as can be seen in Fig.4.23 and 4.24. The compressibility of argon at specific pressure must be calculated with the heat capacity. The heat capacity of argon for 1D and 2D cylindrical symmetry simulations can be seen in Fig.4.25 or Fig.4.27 as shown in different units. The density of argon assumes as an ideal gas law in Fig.4.26. For the cathode region, the property of tungsten such as electrical, thermal conductivities and heat capacity, are shown in Fig.4.28 - 4.30. The detail about statistical mechanics and statistical thermodynamics can be read in detail in references [50], [42], [41], [53], [51], [54], [55].

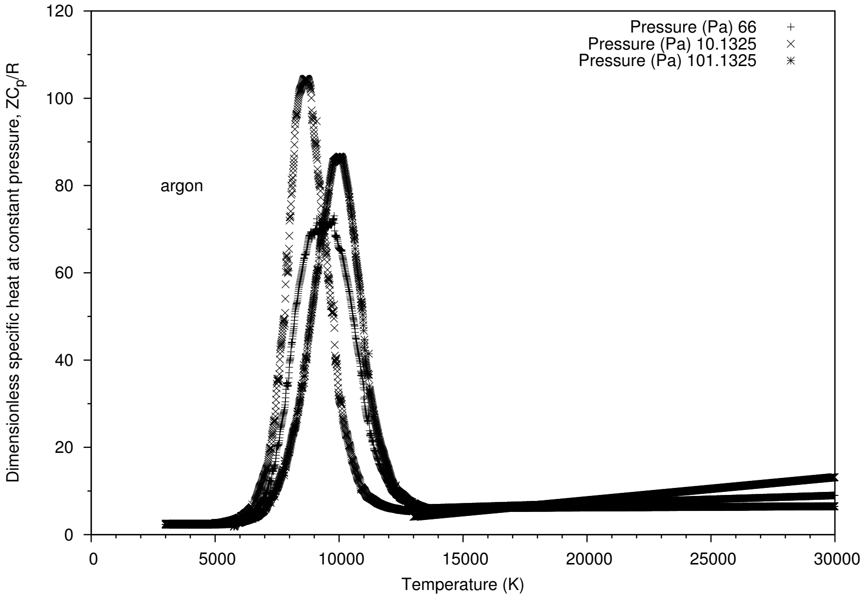

Figure 4.24: Specific heat of argon at constant density as a function of temperature [4].

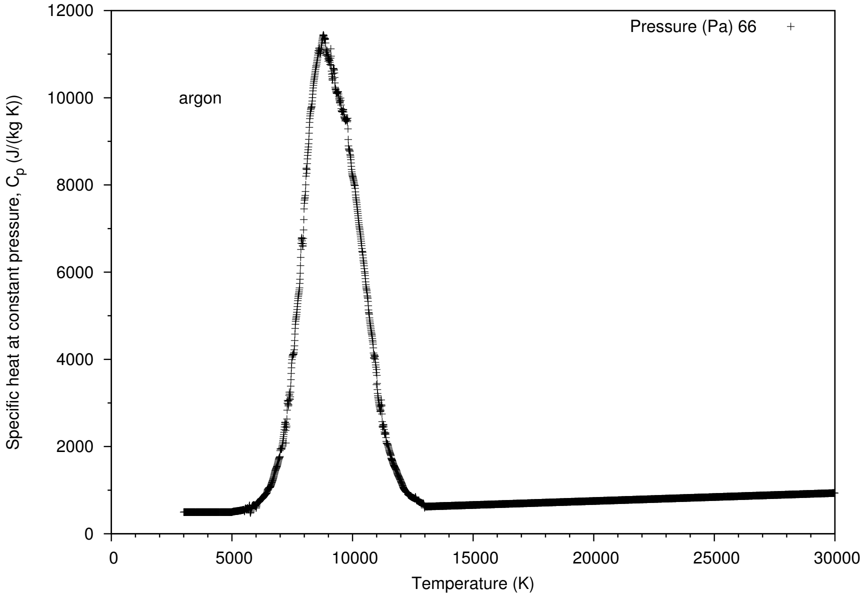

Figure 4.25: Specific heat of argon at 0.5 torr (66 Pa) as a function of temperature.

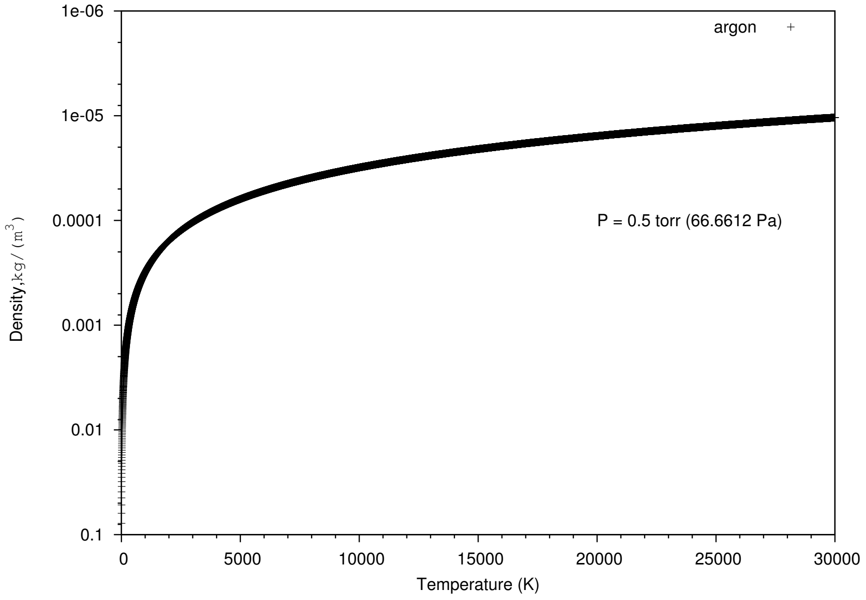

Figure 4.26: Density of argon at 0.5 torr as a function of temperature.

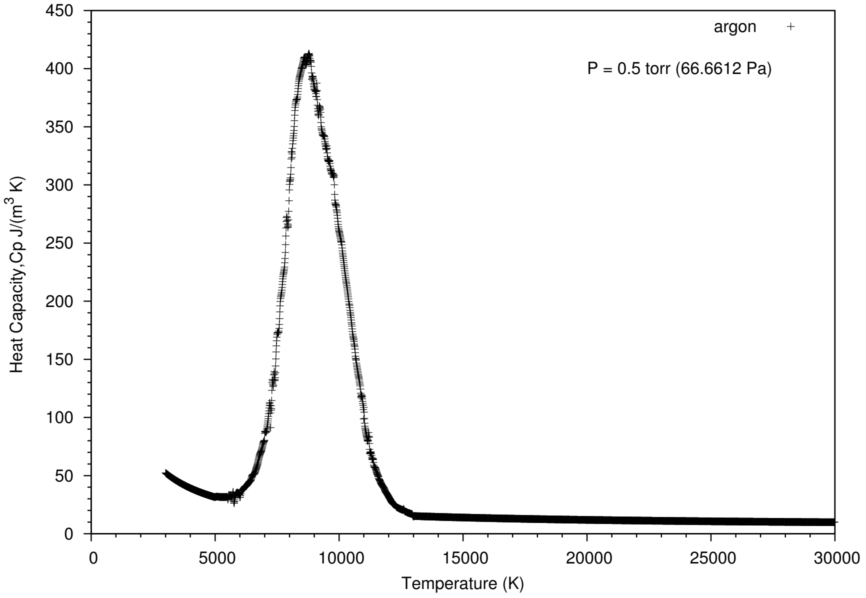

Figure 4.27: Heat capacity of argon at 0.5 torr (66 Pa) as a function of temperature.

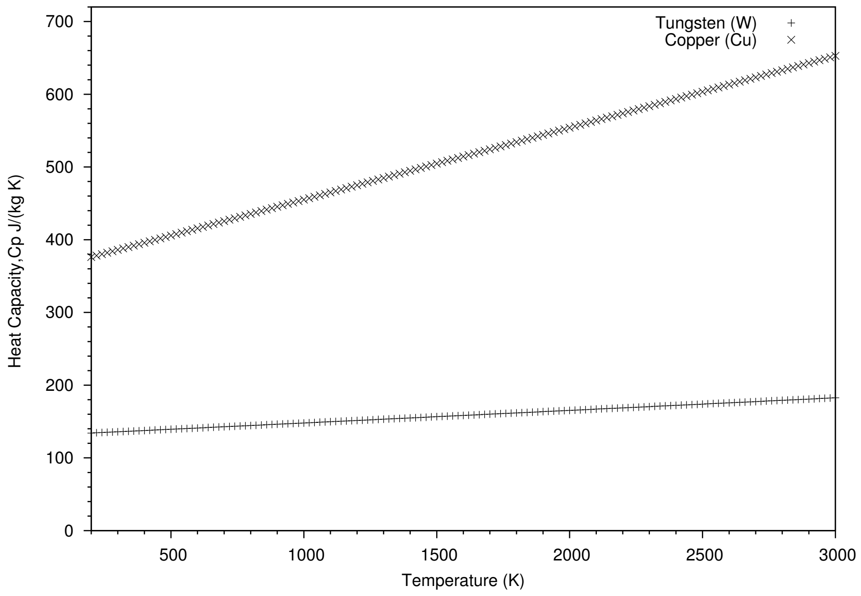

Figure 4.28: Heat capacity of electrods as a function of temperature.

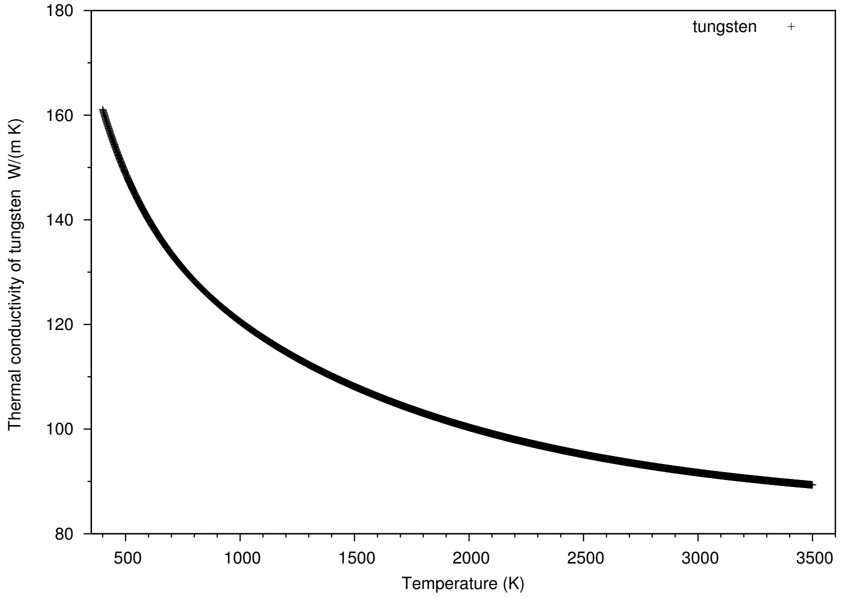

Figure 4.29: Thermal conductivity of tungsten as a function of temperature.

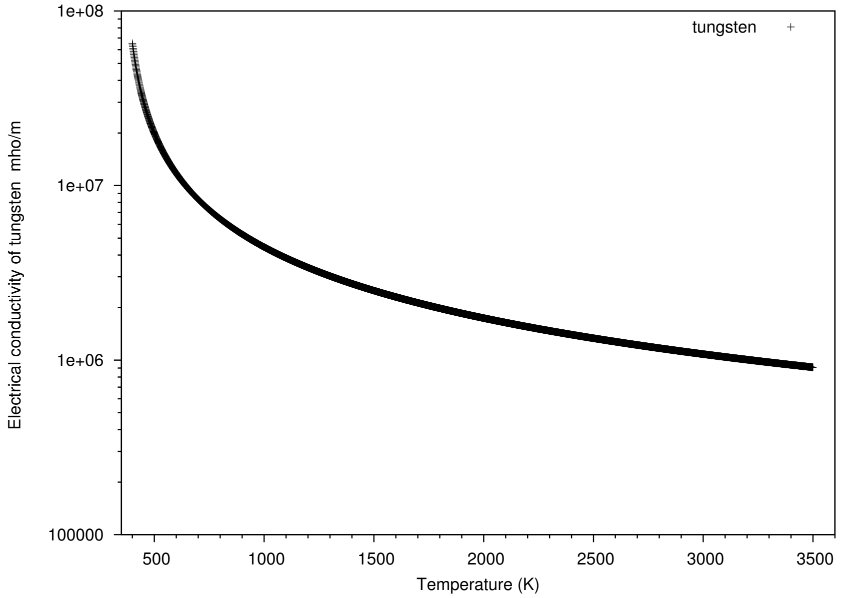

Figure 4.30: Electrical conductivity of tungsten as a function of temperature.

4.3.3.7 Discretization of the equations

The primary values to be determined by the model are the electric potential ϕ and the temperature T . These values are defined at the grid points. The electric field is defined in the primary cells by

$$E_i = \frac{\phi_{i-1} - \phi_i}{L_i} \tag{4.9}$$

Note that the electric field is negative, since it points in the direction from anode to cathode (to- wards negative x), while the potential is increasing with x. Note also that the electric field is taken to be constant within a primary cell.

Integrating the thermal diffusion equation across secondary cell i surrounding grid point i ,

$$\int_{x_i - L_i/2}^{+x_i + L_{i+1}/2} c_p \frac{dT}{dt} dx = \int_{x_i - L_i/2}^{x_i + L_{i+1}/2} \left(JE + \frac{d}{dx}\left(K \frac{dT}{dx} \right) \right) dx \tag{4.10}$$

On the left-hand side, we take the temperature within the secondary cell to be the constant value T_i , so the left-hand side becomes

$$c_p(T_i)\frac{dT_i}{dt}\frac{L_i+L_{i+1}}{2} \tag{4.11}$$

For the Ohmic heating term, the current density is the constant value $-J_0$. The electric field is constant in each primary cell surrounding grid point i , so

$$\begin{aligned} E = (\phi_{i-1}-\phi_i)/L_i, for x_i - L_{i/2} < x < x_i; \\ E = (\phi_i-\phi_{i+1})/L_{i+1}, for x_i < x < x_i + L_{i+1}/2 \end{aligned} \tag{4.12}$$

The integrated Ohmic heating is

$$\begin{aligned} \int_{x_i-L_i/2}^{+x_i+L_{i+1}/2} JEdx = J_0\left[\frac{\phi_i-\phi_{i-1}}{L_i}\frac{L_i}{2}+\frac{\phi_{i+1}-\phi_i}{L_{i+1}}\frac{L_{i+1}}{2}\right] \\ = J_0\frac{\phi_{i+1}-\phi_{i-1}}{2} \end{aligned} \tag{4.13}$$

We take the temperature derivative to be constant in primary cells

$$K\frac{dT}{dx} \approx K(\bar{T}_i)\frac{T_i-T_{i-1}}{L_i} \tag{4.14}$$

where the average temperature in primary cell i is

$$\bar{T}_i \equiv (T_{i-1}+T_i)/2 \tag{4.15}$$

The derivative of this quantity is then defined in secondary cell i as

$$\frac{d}{dx}\left(K\frac{dT}{dx}\right) = \frac{1}{L_{s,i}}\left(K(\bar{T}_{i+1})\frac{T_{i+1}-T_i}{L_{i+1}} - K(\bar{T}_i)\frac{T_i-T_{i-1}}{L_i}\right) \tag{4.16}$$

and, integrating across secondary cell i ,

$$\int_{x_i - L_i/2}^{+x_i + L_{i+1}/2} \frac{d}{dx}\left(K\frac{dT}{dx}\right) = K(\bar{T}_{i+1})\frac{T_{i+1} - T_i}{L_{i+1}} - K(\bar{T}_i)\frac{T_i - T_{i-1}}{L_i} \tag{4.17}$$

Putting together Eq.4.11, 4.13 and 4.17 into the heat equation and rearrange, we have

$$c_p(T_i)\frac{dT_i}{dt}\frac{L_i + L_{i+1}}{2} = J_0\frac{\phi_{i+1} - \phi_{i-1}}{2} + K(\bar{T}_{i+1})\frac{T_{i+1} - T_i}{L_{i+1}} - K(\bar{T}_i)\frac{T_i - T_{i-1}}{L_i} \tag{4.18}$$

or

$$\frac{dT_i}{dt} = \frac{1}{c_p(T_i)L_{s,i}}\left(J_0\frac{\phi_{i+1} - \phi_{i-1}}{2} + K(\bar{T}_{i+1})\frac{T_{i+1} - T_i}{L_{i+1}} - K(\bar{T}_i)\frac{T_i - T_{i-1}}{L_i}\right) \tag{4.19}$$

where T_{i+1} and T_{i-1} are the temperature at grid points i+1 and i-1, respectively. At the cathode edge and the plasma edge, the heat flow equation takes a special form. At grid point N_C , i.e., at the edge of the cathode, the heat flux from the secondary cell to the right is replaced by the sheath heat flux $\dot{Q}_S$ found from the Goodfellow model. Thus,

$$\frac{dT_{N_C}}{dt} = \frac{1}{c_p(T_{N_C})L_{S,N_C}}\left(J_0\frac{\phi_{N_C} - \phi_{N_{C-1}}}{2} + \dot{Q}_S - K(\bar{T}_{N_C})\frac{T_{N_C} - T_{N_{C-1}}}{L_{N_C}}\right) \tag{4.20}$$

Similarly, at grid point N_{C+1}, i.e., at the plasma edge,

$$\frac{dT_{N_{C+1}}}{dt} = \frac{1}{c_p(T_{N_{C+1}})L_{S,N_{C+1}}}\left(J_0\frac{\phi_{N_{C+2}} - \phi_{N_C+1}}{2} - \dot{Q}_S + K(\bar{T}_{N_{C+2}})\frac{T_{N_{C+2}} - T_{N_C+1}}{L_{N_{C+1}}}\right) \tag{4.21}$$

The current density is constant along the 1D MPD thruster simulation and by given cathode temperature and current density to the plasma sheath model. The sheath model will calcualte the heat flux and the sheath voltage to the cathode surface.

In later chapters, these quations can be discretized to calculate the temperature in the cathode and the plasma regions.

4.3.3.8 Solution of the difference equations

The model is initialized by making an initial guess for the temperature at each grid point. Eq.4.19 -4.21 must then be integrated until reach steady state, i.e., until the time derivatives of temperature converge to zero. At each timestep, the electric potentials must be found. This is easy to do in one dimension. Combining Eq.4.5 and Eq.4.9, then rearrange and we can calculate for potential next to the anode to cathode base.

$$\sigma(\bar{T}_i)\frac{\phi_i - \phi_{i-1}}{L_i} = J_0 \tag{4.22}$$

or

$$\phi_{i-1} = \phi_i - \frac{J_0 L_i}{\sigma(\bar{T}_i)} \tag{4.23}$$

Starting at the anode where $\phi_N = 0$, the potential at each successive grid point working toward the base of the cathode can be found by repeated Eq.4.23.
Note, however, that the potential difference across the sheath is special and is given by the Goodfellow model:

$$\phi_{N_C} = \phi_{N_C+1} - V_F \tag{4.24}$$

where V_F is the cathode fall voltage or the sheath voltage (V). As it can be seen in Eq.4.20 and Eq.4.21 that the heat flux at the cathode surface can be transferred toward the cathode for cold cathode temperature or outward the cathode for hot cathode temperature for every time step until this 1D MPD thruster reaches steady state condition.

4.4 2D Cylindrical Symmetry MPD thruster Simulation

A solid cathode model represents a set of equations descibing the flow of heat, potential, electric fiedl and current density in the cathode. Fig.4.31, and Fig.4.32 show the solid cathode assembly under experiment and the diagram of the solid cathode assembly.

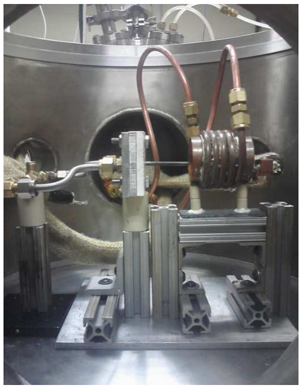

Figure 4.31: Solid cathode assembly under an experiment [2].

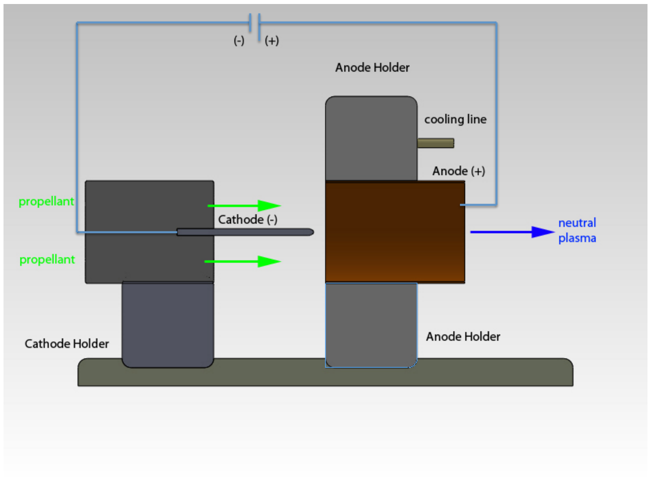

Figure 4.32: Section side view diagram of solid cathode assembly.

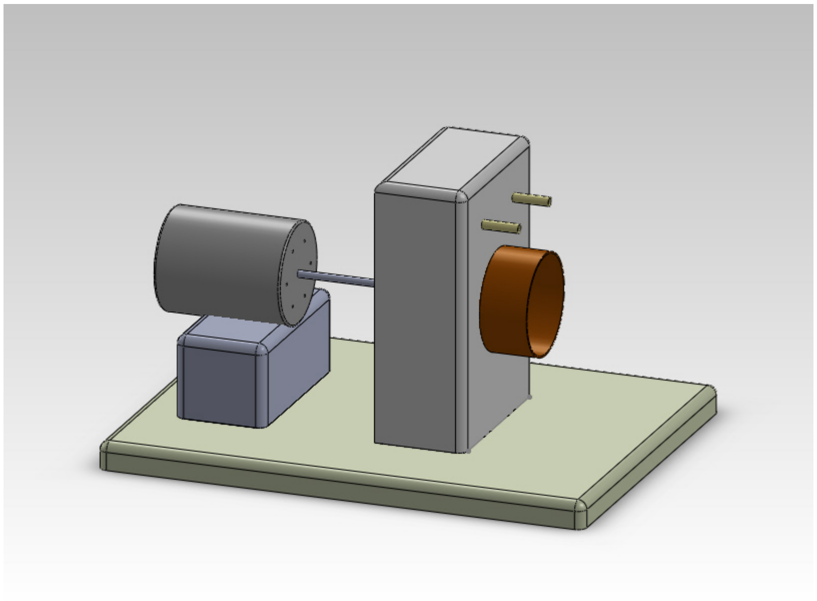

Figure 4.33: Solid cathode assembly.

Fig.4.33 - 4.35 show the diagram for the solid cathode assembly, the pressure chamber under experiment and the dimension of pressure chamber. The pressure chamber diameter and length are 520 mm and 370 mm, respectively. The operating pressure inside the chamber is 640 sccm. The detail experiment operating conditions can be read in references [2] and [3].

As can be seen in Fig.4.36, the cathode is placed 6 mm from the center of concentric anode. The cathode lenght is 75 mm with 3.96 mm in diameter. The length and the diameter of anode are 76mm and 54 mm, respectively. The argon is used as a propellant and it is injected into the base of the cathode with 1.5 mm diameter tube. The anode is surrounded by cooling line to maintain anode temperature below the melting point. The material of cooling line in this experiment is copper. The white material supports the cathode and anode is ceramic or electrical insulators. The high voltage and high current power supply are connected to cathode and anode. The solid cathode assembly in the pressure chamber in Fig.4.37 and the numerical boundary can be seen in Fig.4.38.

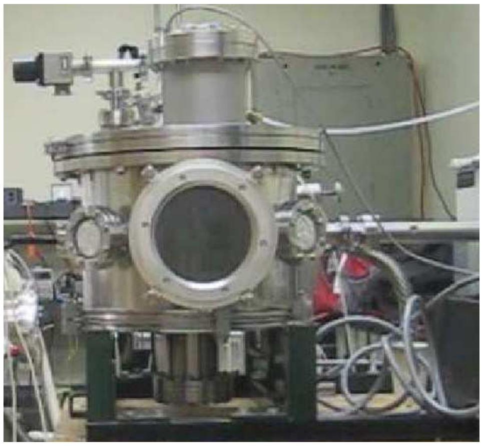

Figure 4.34: Pressure chamber under an experiment [2].

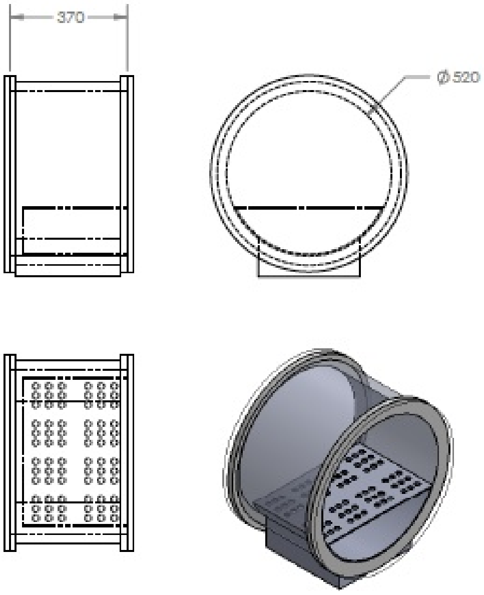

Figure 4.35: Dimension of the pressure chamber.

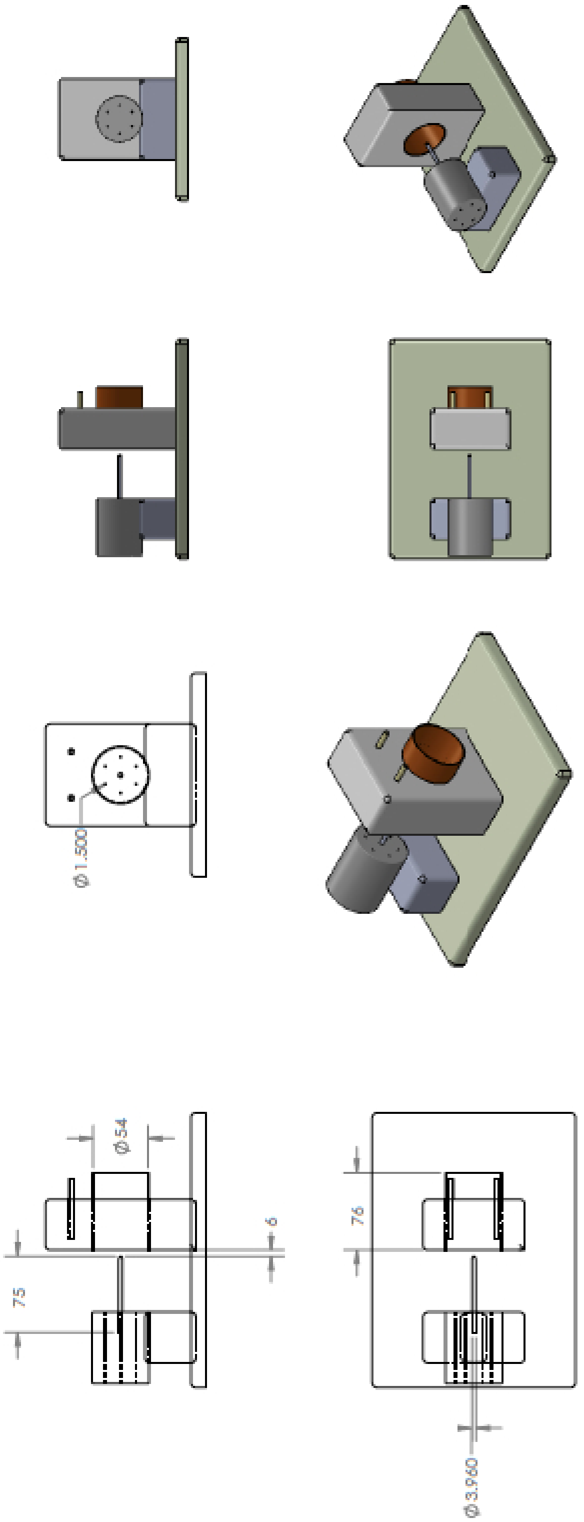

Figure 4.36: Solid cathode assembly diagram from experimental set up [8].

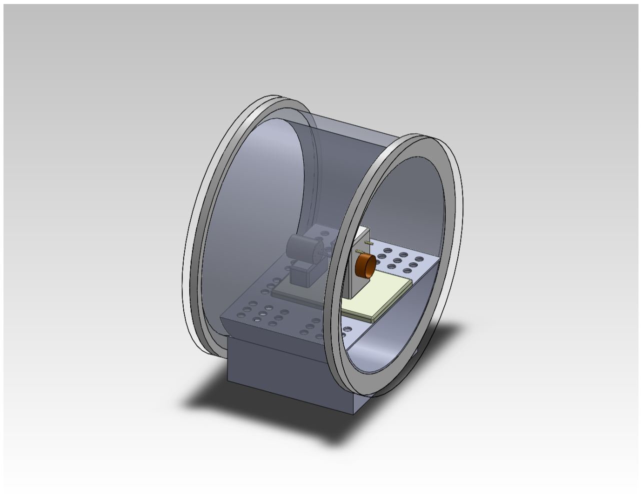

Figure 4.37: Solid cathode assembly in the pressure chamber.

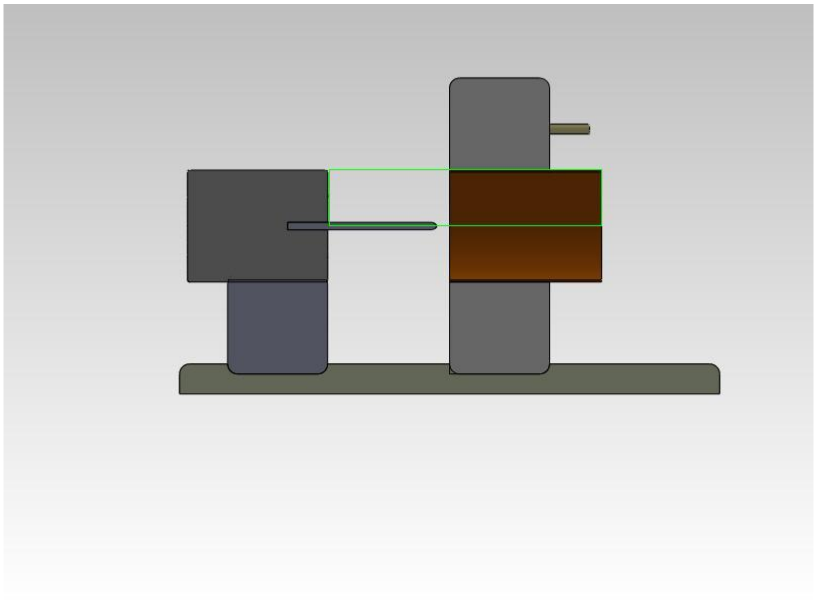

Figure 4.38: The boundary of 2D cylindrical symmetry MPD thruster simulation.

4.4.1 Computational Grid for Catesian Coordinates

4.4.1.1 Introduction

This triangular computational grid method is well developed by Winslow [1] called topologically regular. That is, it is topologically equivalent to an equilateral triangle array in which six triangles meet at every interior mesh point. The primary triangle mesh have a common vertex anda secondary mesh of 12-sided figures whose vertices are alternately the centroids of the six adjacent triangles and the midpoints of the six adjacent sides as can be viewed in Fig.4.39 and can be read in detail discussion in [1].

The main advantages of using the structured triangular grids rather than use the rectangular structural grids are that the triangular structural grid can be easily fitted into the irregular shape of the boundary domain and it can be used with adaptive methode to confine the grid in the specific region.

In next several chapters, the detail equations on how to apply for the cylindrical symmetery will be introduced.

4.4.1.2 Numerical Construction of Topologically Regular Nonumiform Triangle Meshes

The Laplace equation is in the form

$$\nabla^2\chi = 0,\ \ \nabla^2\psi = 0 \tag{4.25}$$

Solving Eqn.4.25, the intersecting "equipotentials" $\chi =$ constant and $\psi =$ constant, and together with the thrid set drawn through the intersection points, form the desired triangle mesh. A mesh constructed in this way is smooth because the well-known averaging propperty of solutions to Laplace's equation. The derivation of this method can be further study in Appendix A, B and [1] in details.

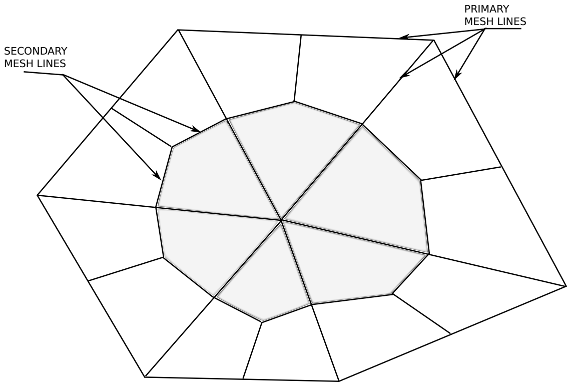

Figure 4.39: Primary and secondary mesh lines.

4.4.1.3 Description of the method and derivation of the difference equations [1]

The nonlinear diffusion equation can be written as

$$c\frac{\partial \phi}{\partial t} = \nabla \cdot (\lambda \nabla \phi) + S \tag{4.26}$$

The generaized Poisson equation for steady state can be expressed as

$$\nabla \cdot (\lambda \nabla \phi) + S = 0 \tag{4.27}$$

where S is the function of position or source term (i.e.thermionic heating), ϕ is the potential or voltage at the vertice of triangle, λ is a function of ϕ or its derivatives (a positive function or electrical or thermal conductivity; however, this study includes only electrical conductivity and assumed the thermally condition of an MPD thruster is in steady state). That is, the temperature is given and assumed constant at the cathode and the plasma regions as in steady state. As a result, the source term S or the thermionic heating will not be considered. However, this source term can be included in the future models as described in the appendix C. Fig.4.40 shows the triangle i+1/2 defined by the two side vectors $\boldsymbol{s}_i, \boldsymbol{s}_{i+1}$, with values ϕ, ϕ_i, ϕ_{i+1} at the respective vertices.

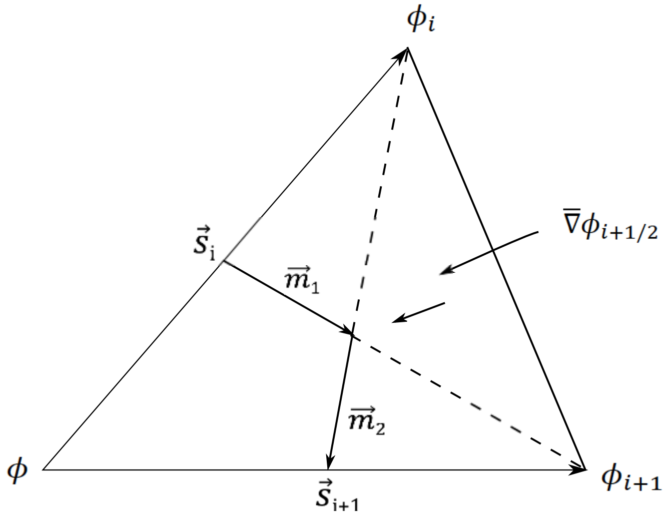

Figure 4.40: Vectors used in flux calculation [1].

The potential or voltage of each triangular grid point can be seen in Fig.4.40 and the gradient of the potential in each triangular can be calculated from Eq.4.29. This term of gradient is definded as the electric field. Then, the current density can be determined from Eqn.4.30. The derivation detail can be read in [1] The equation of this triangle can be expressed by

$$\phi_j = \phi + \boldsymbol{s}_j \cdot \boldsymbol{\nabla}\phi_{i+1/2}, \quad j = i, i+1 \tag{4.28}$$

A vector $\boldsymbol{\nabla}\phi_{i+1/2}$ is given by

$$\boldsymbol{\nabla}\phi_{i+1/2} = \frac{(\phi_i - \phi)\boldsymbol{s}_{i+1}^{\dagger} - (\phi_{i+1} - \phi)\boldsymbol{s}_i^{\dagger}}{\boldsymbol{s}_i \cdot \boldsymbol{s}_{i+1}^{\dagger}} \tag{4.29}$$

Within each triangle the flux of the diffusing quantity is given by

$$\boldsymbol{F}_{i+1/2} = -\lambda_{i+1/2}\boldsymbol{\nabla}\phi_{i+1/2} \tag{4.30}$$

Comparing Ohm's law in Eqn.(2.4),which is $\bar{j} = -\sigma_e \nabla V$, and Winslow Theory [1], which is $\lambda\nabla\phi$. The term $\sigma_e \equiv \lambda$ and $V \equiv \phi$ where j refers to current density, σ_e and λ refer to electrical conductivity, V and ϕ refer to electrical potentials or voltage.

4.4.2 Outline of the Algorithm

The details of this algorithm was developed by Erwin [56]. In this section, we will construct the 2D cylindrical symmetry with the dimensions as same as the experiment set up as in previous section. However, the dimension can be changed in our 2D simulation as needed in following chapters. In this section, the program will be described in several steps followed.

1. As it can be seen in Fig.4.41, the physical grid space are separated into three main regions, which are cathode, the plasma, and the sheath regions on the cathode surface. The number of cells in the computational grid space can be defined as seen in Fig.4.42.

2. In Fig.4.41, $R_A, R_C, Z_{LC}, Z_{SIM}, Z_{LA}$, which represent the radius of the anode, the radius of the cathode, the length of the cathode, the length of the MPD system, and the length of the anode. The values are 27 mm, 1.98 mm, 75 mm, 157 mm, and 76 mm, respectively.

3. In Fig.4.42, N_1, N_2, N_3, N_4, N_5, which represent the number of cells in the radius of anode, the radius of the cathode, the lenght of the cathode, the length of the MPD system, and the length of the anode, repectively.

4. As mentioned in the Winslow thoery [1], the program in this section only in a catesian coordinate and it will generate the computational grid space and physical grid space as in Fig.4.43 and Fig.4.44. In later chapters, we will develp the 2D cylindrical symmetry to calculate of 2D cylindrical symmetry MPD thruster.

5. Each points are saved in hexagon grid point and each point has a coordinate value in both computational and physical grid space.

6. In Fig.4.43, there are two type of triangles. The first type of triangle are 1, 3, 5, 7, 9, 11, 14, 16 etc. The latter type are 2, 4, 6, 8, 10, 12, 13, 15 etc.

7. The Winslow [1] calculated the computational grid and physical grid. For the physical grids, an additional numerical method to solve Laplace equation is required called "LU decomposition" and "backward substitutions".

8. The output can be ploted using ghost script (gview) or gnuplot as can be seen in Fig4.44 and Fig.4.45 with different of the N_1, N_2, N_3, N_4, N_5 values.

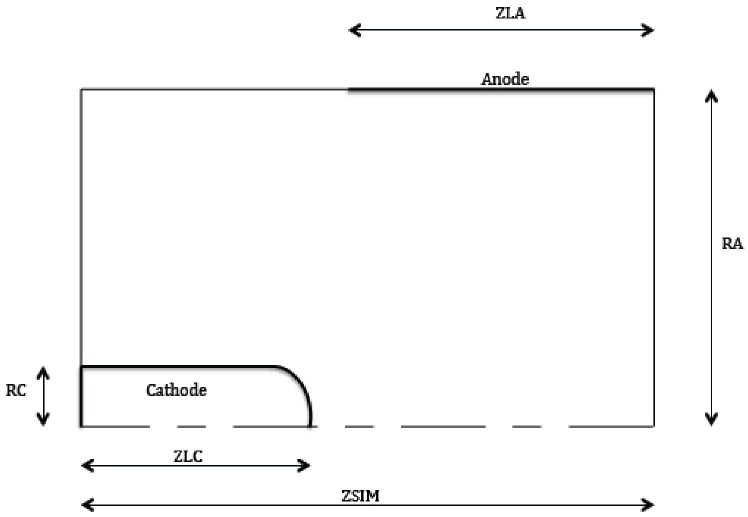

Figure 4.41: Physical grid space with the physical dimension defined as R_C, R_A, Z_{SIM}, Z_{LC} and Z_{LA}.

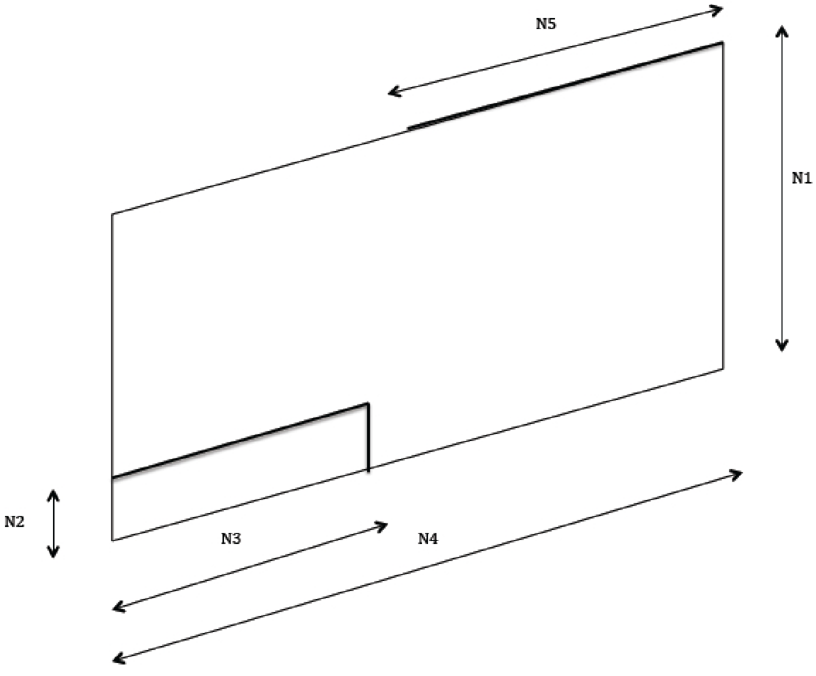

Figure 4.42: Computational grid space with the number of cells definded as N_1, N_2, N_3, N_4, and N_5.

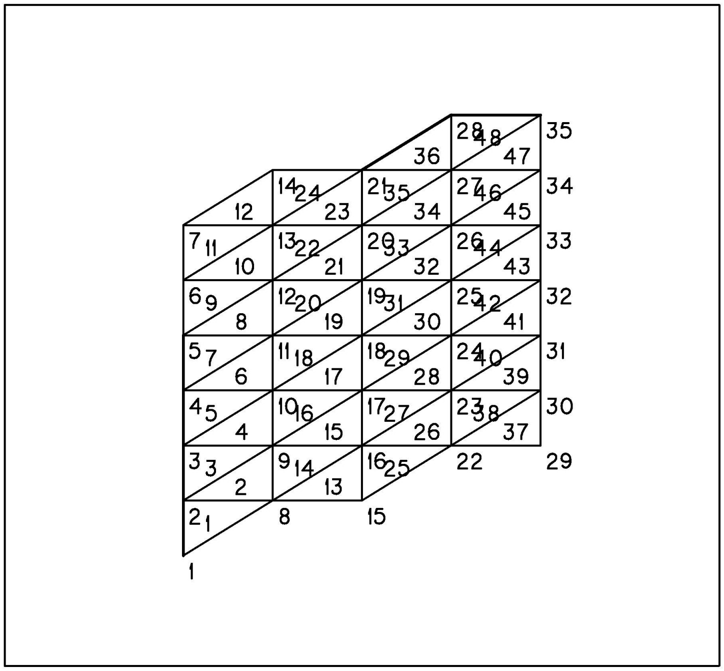

Figure 4.43: Computational grid space N1=6, N2=4, N3=2, N4=4, N5=2.

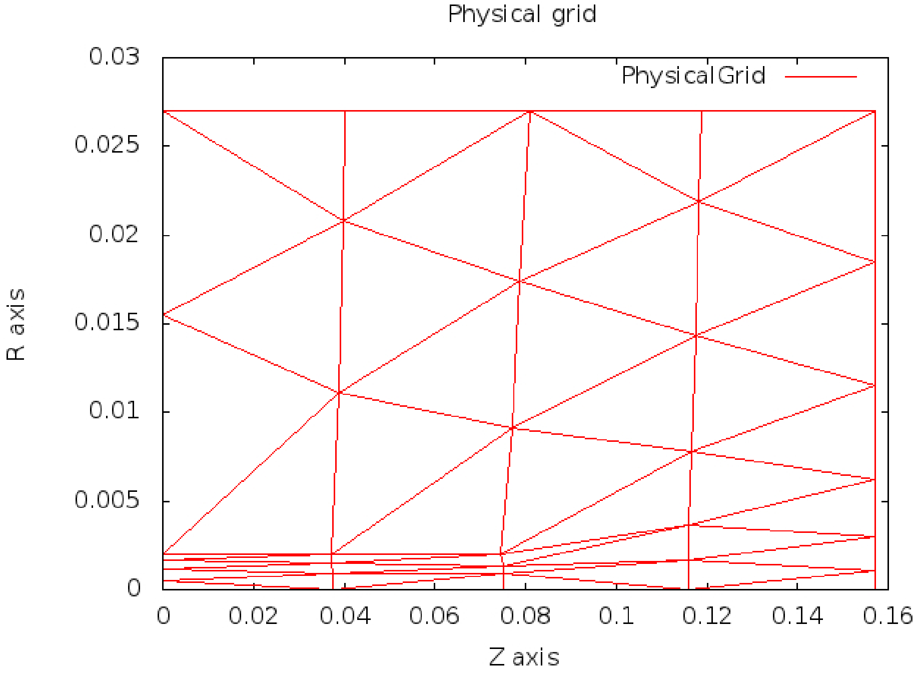

Figure 4.44: Physical grid space $N_1 = 6, N_2 = 4, N_3 = 2, N_4 = 4, N_5 = 2$.

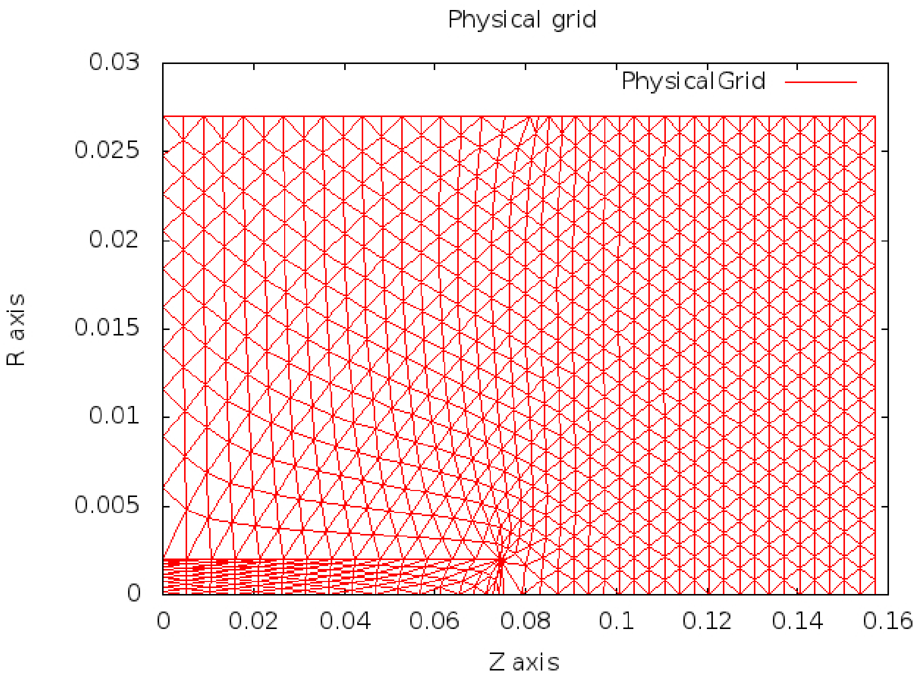

Figure 4.45: Physical grid space $N_1 = 20, N_2 = 10, N_3 = 18, N_4 = 42, N_5 = 24$.

4.4.3 The current and potential boundary conditions

As can be seen in Fig.4.43 or Fig.4.44, point 1 to 5 represente as cathode base and point 21, 28, 35 represent as anode. Using [1], the current at 1, 2, 3 ,4 can be expressed as

$$\begin{aligned}
I_1 &= w_{2\to1}(\phi_2 - \phi_1) + w_{8\to1}(\phi_8 - \phi_1) \\
I_2 &= w_{1\to2}(\phi_1 - \phi_2) + w_{8\to2}(\phi_8 - \phi_2) + w_{9\to2}(\phi_9 - \phi_2) + w_{3\to2}(\phi_3 - \phi_2) \\
I_3 &= w_{2\to3}(\phi_2 - \phi_3) + w_{9\to3}(\phi_9 - \phi_3) + w_{10\to3}(\phi_{10} - \phi_3) + w_{4\to3}(\phi_4 - \phi_3) \\
I_4 &= w_{3\to4}(\phi_3 - \phi_4) + w_{10\to4}(\phi_{10} - \phi_4) + w_{11\to4}(\phi_{11} - \phi_4) + w_{5\to4}(\phi_5 - \phi_4) \\
I_5 &= w_{4\to5}(\phi_4 - \phi_5) + w_{11\to5}(\phi_{11} - \phi_5) + w_{12\to5}(\phi_{12} - \phi_5) + w_{6\to5}(\phi_6 - \phi_5)
\end{aligned} \tag{4.31}$$

At the cathode base, we assumed that the potential values at point 1, 2, 3, 4 and 5 are equal ($\phi_1 = \phi_2 = \phi_3 = \phi_4 = \phi_5$) we got

$$\begin{aligned} &\text{At grid 1:} \qquad \phi_1 = \phi_2 \\ &\text{At grid 2:} \qquad \phi_2 = \phi_3 \\ &\text{At grid 3:} \qquad \phi_3 = \phi_4 \\ &\text{At grid 4:} \qquad \phi_4 = \phi_5 \end{aligned} \tag{4.32}$$

The expression can be reduced to

$$\begin{aligned} I_1 &= w_{8\to1}(\phi_8 - \phi_1) \\ I_2 &= w_{8\to2}(\phi_8 + \phi_2) + w_{9\to2}(\phi_9 - \phi_2) \\ I_3 &= w_{9\to3}(\phi_9 + \phi_3) + w_{10\to3}(\phi_{10} - \phi_3) \\ I_4 &= w_{10\to4}(\phi_{10} + \phi_4) + w_{11\to4}(\phi_{11} - \phi_4) + w_{5\to4}(\phi_5 - \phi_4) \\ I_5 &= w_{11\to5}(\phi_{11} - \phi_5) + w_{12\to5}(\phi_{12} - \phi_5) + w_{6\to5}(\phi_6 - \phi_5) \end{aligned} \tag{4.33}$$

The total current is 60A which can be expressed by

$$I_1 + I_2 + I_3 + I_4 + I_5 = 60 \tag{4.34}$$

substituting

$$\begin{aligned} w_{8\to1}(\phi_8 - \phi_1) + w_{8\to2}(\phi_8 + \phi_2) + w_{9\to2}(\phi_9 - \phi_2) + w_{9\to3}(\phi_9 + \phi_3) \\ +w_{10\to3}(\phi_{10} - \phi_3) + w_{10\to4}(\phi_{10} + \phi_4) + w_{11\to4}(\phi_{11} - \phi_4) \\ +w_{5\to4}(\phi_5 - \phi_4) + w_{11\to5}(\phi_{11} - \phi_5) + w_{12\to5}(\phi_{12} - \phi_5) \\ +w_{6\to5}(\phi_6 - \phi_5) = 60 \end{aligned} \tag{4.35}$$

At the anode region, we use similar approach and we get

$$\begin{aligned} I_{21} &= w_{20\to21}(\phi_{20}-\phi_{21}) + w_{27\to21}(\phi_{27}-\phi_{21}) + w_{28\to21}(\phi_{28}-\phi_{21}) \\ &\qquad + w_{14\to21}(\phi_{14}-\phi_{21}) + w_{13\to21}(\phi_{13}-\phi_{27}) \\ I_{28} &= w_{27\to28}(\phi_{27}-\phi_{28}) + w_{35\to28}(\phi_{35}-\phi_{28}) + w_{21\to28}(\phi_{21}-\phi_{28}) \\ I_{35} &= w_{34\to35}(\phi_{34}-\phi_{35}) + w_{28\to35}(\phi_{28}-\phi_{35}) + w_{27\to35}(\phi_{27}-\phi_{35}) \end{aligned} \tag{4.36}$$

We assumed that the potential values of point 21, 28, 35 are equal ($\phi_{21} = \phi_{28} = \phi_{35}$) and the expression reduce to

$$\begin{aligned} I_{21} = w_{20\to21}(\phi_{20}-\phi_{21}) + w_{27\to21}(\phi_{27}-\phi_{21}) + w_{14\to21}(\phi_{14}-\phi_{21}) \\ + w_{13\to21}(\phi_{13}-\phi_{27}) \\ I_{28} = w_{27\to28}(\phi_{27}-\phi_{28}) \\ I_{35} = w_{34\to35}(\phi_{34}-\phi_{35}) + w_{27\to35}(\phi_{27}-\phi_{35}) \end{aligned} \tag{4.37}$$

However, the sum of current at cathode point 1, 2, 3, 4, 5 equal to the sum of current at anode point 21, 28, 35. The total current is $I_1 + I_2 + I_3 + I_4 + I_5 = I_{21} + I_{28} + I_{35} = I_{total} = 60A$. The total current at the cathode base and the anode region are within 1 percent error.

Chapter 5

Completed Works

Objectives for this doctoral work are as follows:

Theory/Modeling Goals

1. Completed a 1D MPD thruster simulation to fully combined cathode and plasma regions with the sheath model and calculate for sheath voltage across the cathode and the plasma regions at steady state.

2. Completed the 1D MPD thruster simulation to achieve prediction the heat flux, and temperature until the system reaches steady state.

3. Completed to include the electrical, thermal conductivities, and heat capacity which are temperature dependent, derived by Ahtye, W. F. [4] into the 1D MPD thruster simulations at 66 Pa.

4. Completed 2D cylindrical symmetry MPD thruster simulation base on Winslow [1], [26], which correctly predicts the characteristic profile of specific plasma properties in the plasma region and the cathode region. At the cathode surface, it will include the plasma sheath model [3]. The plasma and cathode properties of specific interest are: the temperature, potential, electric field, and current density.

5. Completed to include the electrical, thermal conductivities, and heat capacity which are temperature dependent, derived by Ahtye, W. F. [4] into the 2D

cylindrical symmetry MPD thruster simulation at 66 Pa.

6. Completed to verify the newly gained quantitative description of the plasma and cathode properties such temperature, potential, electric field and current density with argon as a propellant until the system reach steady state.

7. Completed to verify the convection and pressure effect to the cathode surface for 2D cylindrical symmetry MPD thruster.

8. Completed to verify the plasma arc attachment edge called electroarc edge on the cathode surface.

Chapter 6

Research Methodology

Fundamental Assumptions

In 1D and 2D cylindrical symmetry MPD thruster simulations, the plasma region is assumed to have a local thermal equilibrium (LTE) and the bulk of plasma is represented as fluid, which obeys the ideal gas state equation. Also, the coulomb force is neclected and no other transport phenomena except the electrical conductivity, the thermal conductivity, and the heat capacity. That is, the viscousity effects are neclected.

Moreover, the plasma is in Saha equilibrium with no chemical reaction. Therefore, the simulation starts when ionization has completed at the inlet. Also, only the first ionization has been included and the electron temperature is around 0.9 - 1.0 eV. For computational grid of 1D, the thermal conductivity and the electrical conductivity are in the cells but temperature, potential and heat capacity are at the grid points. For 2D cylindrical symmetry, the grid points represent the temperature, potential; however, inside the triangular grids represent electrical, thermal conductivities, the heat capacity, the current density,and the electric field. For this study, the magnetic field in the flow will not be calculated, however, it can be further developed as needed. Fig.6.1 shows the physical grid space in our 2D cylindrical symmetry MPD thruster simulation and the surface of the cathode where the sheath model will be applied.

Furthermore, there is no current into the insulator at the upstream end and no current flow across the nozzle exit plane at the downstream end (the current flow only from anode to cathode). Also, the magnetic field on the arc jet axis assume to be zero. At the base of the cathode from a to b in Fig.6.1, the potential are the same. For the open boundaries c-d and e-d, the mass flow assumes to be constant. Also, the magnetic field on the azimuthal axis is zero.

As shown in Fig.6.1, the calculation domain consists of the cathode, plasma and sheath regions. The plasma sheath model is included in the calculation on the cathode surface. That is, the 2D cylindrical symmetry MPD thruster balances the energy between the cathode, the plasma and the sheath regions until the system reach steady state.

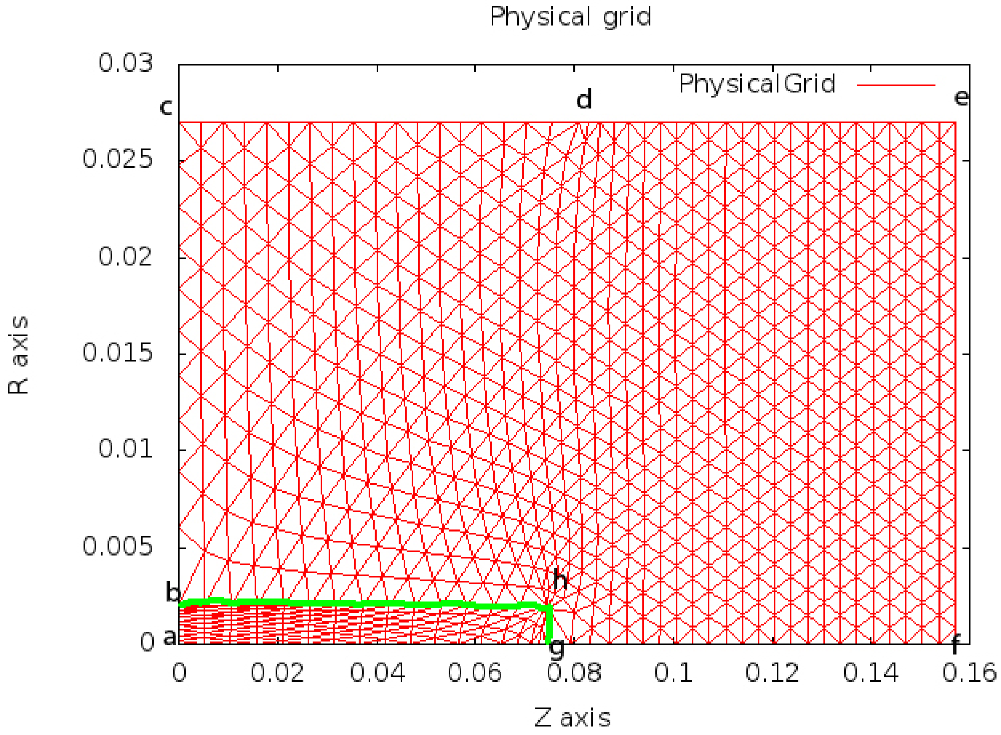

Figure 6.1: 1D near-cathode plasma calculation on green color with N1=20, N2=10, N3=18, N4=42, N5=24.

The Plasma Arc Attachment Edge on the Cathode Surface Factors

The factors of the plasma arc attachement edge on the cathode strongly depend on the work function, surface finishing, molecular structure of the cathode. Also, there are other factors i.e. shape of the current and magnetic field lines, the type of gas used, the space and orientation of magnetic field, the vacuum level within the vacuum chamber, the polarity of the direct current, the electrical voltage used, and the position of the electrods in the vacuum chamber that would have an effect to the plasma attachement area.

For this research, the electroarc edge is defined as the plasma arc attachment edge on the cathode surface. This electroarc edge locates at the minimum electric field value on the cathode surface as will be described later. In order to locate this edge, the cathode, plasma and the plasma sheath model is fully integrated and iterated until the system reaches steady state. While iterating the electric field values on the cathode surface will adjust simultaneously with the current density, temperature, potential. That is, for particular parameters set up shown in the next chapter will be the primary factor to consider the limit of plasma sheath attachment edge, electroarc edge, on the cathode surface. However, the other factors can be included to further study the effect of plasma attachment edge on the cathode surface.

Chapter 7

Results

In this chapter, the 1D and 2D cylindrical symmetry MPD thruster simulation calculates the cathode, the plasma regions, and the plasma sheath model simultaneously. This simulation balances the energy between two regions with the plasma sheath model until the simulation reaches steady state. The following detail explains the method for the 1D and 2D cylindrical symmetry MPD thruster.

7.1 Converged Region of Plasma Sheath Model

The plasma sheath model [3] has a specific converged region as can be seen.

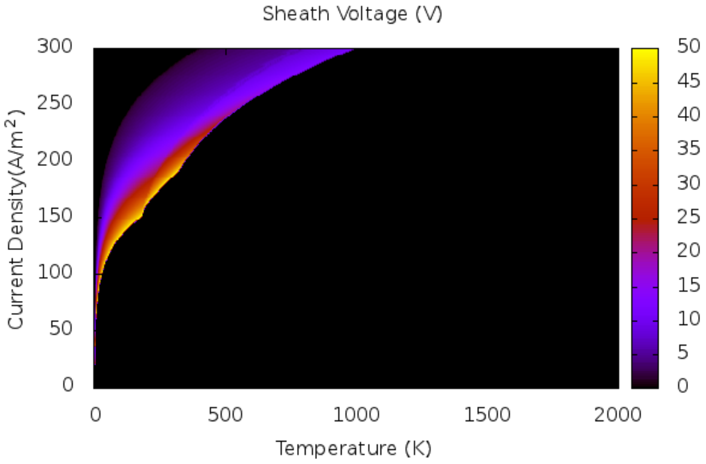

Figure 7.1: The converged area of sheath voltage of the plasma sheath model as a function of temperatureand current density.

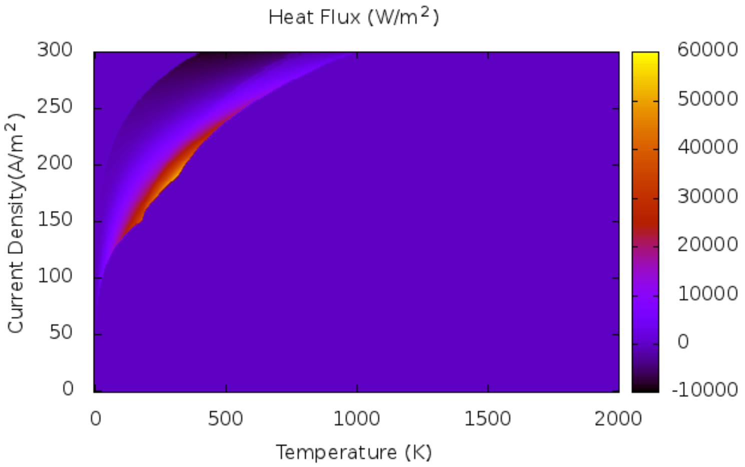

Figure 7.2: The converged area of heat flux of the plasma sheath model as a function of temperatureand current density.

As can be seen in Fig.7.1 and 7.2, only certain parameters of temperature and current density on cathode surface will provide the heat flux from plasma toward cathode regions. That is, the heat flux can be transfered from plasma to cathode regions or from cathode to plasma.

7.2 Fully Combined Cathode and Plasma Regions with Plasma Sheath Model in 1D MPD Thruster Simulation

From Eq.(4.19), this equation will be applied from grid the cathode base to NC-1 and NC+2 to the anode and it can be shown as;

$$\frac{dT_i}{dt} = \frac{1}{c_p(T_i)L_{s,i}}\left(J_0\frac{\phi_{i+1}-\phi_{i-1}}{2} + K(\bar{T}_{i+1})\frac{T_{i+1}-T_i}{L_{i+1}} - K(\bar{T}_i)\frac{T_i-T_{i-1}}{L_i}\right) \quad (7.1)$$

Using forward difference to the first derivative in time, which has a first order approximation $O(\Delta t)$ and it can be expresses as;

$$\frac{dT_i}{dt} = \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{7.2}$$

where Δt represents time step and the upper case n represents time step number n. By substituting into the equation, it can be expresses as;

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{1}{c_p(T_i^n)L_{s,i}}\left(J_0\frac{\phi_{i+1}^n - \phi_{i-1}^n}{2} + K(\bar{T}_{i+1}^n)\frac{T_{i+1}^n - T_i^n}{L_{i+1}} - K(\bar{T}_i^n)\frac{T_i^n - T_{i-1}^n}{L_i}\right) \tag{7.3}$$

rearrange for T_i^{n+1} as;

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{c_p(T_i^n)L_{s,i}}\left(J_0\frac{\phi_{i+1}^n - \phi_{i-1}^n}{2} + K(\bar{T}_{i+1}^n)\frac{T_{i+1}^n - T_i^n}{L_{i+1}} - K(\bar{T}_i^n)\frac{T_i^n - T_{i-1}^n}{L_i}\right) \tag{7.4}$$

also the potential from the cathode base to NC-1 and NC+2 to the anode and it can be shown as;

$$\phi_{i-1}^{n+1} = \phi_i^n - \frac{J_0 L_i}{\sigma(\bar{T}_i^n)} \tag{7.5}$$

Again, the plasma sheath must be included at NC+1 and it can be shown as;

$$\phi_{N_C} = \phi_{N_C+1} - V_F \tag{7.6}$$

where V_F is the cathode fall voltage or plasma sheath voltage which is a function of temperature and the current density.

Using the same method and applied to Eq.4.20 and 4.21, they can be expresseed as;

$$T_{NC}^{n+1} = T_{NC}^n + A\left(J_0\frac{\phi_{NC}^n - \phi_{NC-1}^n}{2} + \dot{Q}_S^n - K(\bar{T}_{NC}^n)\frac{T_{NC}^n - T_{NC-1}^n}{L_{NC}}\right) \tag{7.7}$$

$$T^{n+1}_{NC+1} = T^{n}_{NC+1} + B\Bigg(J_0 \frac{\phi^n_{NC+2} - \phi^n_{NC+1}}{2} - \dot{Q}^n_S - K(\bar{T}^n_{NC+2}) \frac{T^n_{NC+2} - T^n_{NC+1}}{L_{NC+1}} \Bigg) \tag{7.8}$$

where A = $\frac{\Delta t}{c_p(T^n_{NC}) L_{S,NC}}$, B = $\frac{\Delta t}{c_p(T^n_{NC+1}) L_{S,NC+1}}$ and $\dot{Q}^n_S$ is the heat transfer from the plasma sheath, which is a function of temperature and the current density.

7.2.1 Critical Time Increment of 1D MPD Thruster Simulation

Von Neumann criteria

$$\Delta t \leq \frac{c_p(T^n_i) L_{S,i} L_i}{2K(\bar{T}^n_{i+1})} \tag{7.9}$$

Ohmic heating criteria

$$\Delta t \leq \frac{c_p(T^n_i) L_{S,i}}{J_0} \tag{7.10}$$

As can be seen in the above equations, Von Neumann criteria composes of the thermal conductivity and the heat capacity terms; however, the Ohmic heating criteria relates to the current density term and the heat capacity terms. For explicit methode, Von Neumann and Ohmic heating criteria are required to compute stable and accurate results.

7.2.2 Outline of Algorithm of 1D MPD Thruster Simulation

In this section, the program will be described in several steps:

1. Set the physical length of the cathode and the plasma regions. Then, set the number of cells in each regions. After that, set the initial temperature and voltage at every grid points. For cathode region, the temperature can be between 2800-3500 K. For plasma region, only first ionization is required and the temperature

is around 0.9-1.0 eV. The cathode base temperature is 1500K. Then, the average temperature can be calculated.

2. Given the current density into the system as the current density is the same along the 1D MPD thruster simulation.

3. Calculate the critical time increment of Von Neumann and Ohmic. Set the number of step to calculate for the DO loop.

4. In the DO loop to solve for temperature and voltage (potential) at the grid points

4.1 Updated for the thermal conductivity and electrical conductivity in the cathode region.

4.2 Updated for the thermal conductivity and electrical conductivity in the plasma region.

4.3 Calculated for the voltage using Eqn.4.23 for the plasma region.

4.4 Called the subroutine of plasma sheath model to obtain the heat flux and the sheath voltage.

4.5 Added the sheath voltage to the cathode tip grid.

4.6 Calculated for for the voltage using Eqn.4.23 for the remaining cathode grids.

4.7 Solved for the temperature in plasma region.

4.7.1 At the plasma grid NC+1, using Eqn.4.21.

4.7.2 For the remaining plasma grids, using Eqn.4.19.

4.8 Solved for the temperature in cathode region.

4.8.1 At the cathode tip grid NC, using Eqn.4.20.

4.8.2 For the remaining cathode grids, using Eqn.4.19.

5. Print out the temperature, voltage values.

7.2.3 1D MPD Thruster Simulation Results

The cathode tip temperature of several initial temperature from 2700 to 3100 K reach the steady state after 900 s as can be seen in Fig.7.3 and the zoom in of Fig.7.4. The prediction of cathode tip at steady state is around 2950 K. That is, when the initial cathode tip temperature is below 2950 K, the plasma region supplies the heat from the plasma region to the cathode tip. However, if the initial temperature of the cathode tip is above 2950 K, the heat will transfer from the cathode tip to the plasma region.

The sheath voltage also approach a steady state after 900 s. and the sheath voltage value can be estimated to be around 6 V in Fig.7.5 and 7.6. With the increasing initial temperature, the initial value of sheath voltage decreases in Fig.7.5.

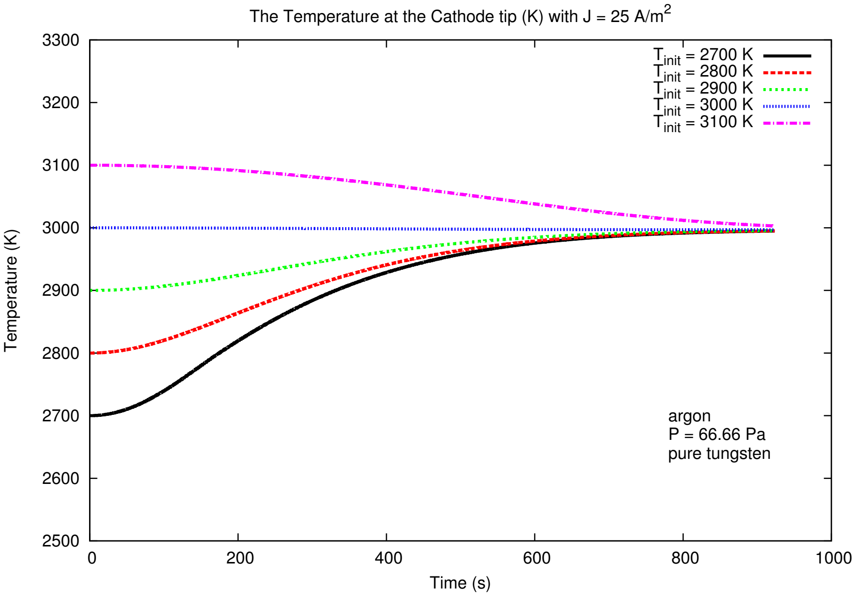

Figure 7.3: The cathode tip temperature as a function of time for a pressure of 66 Pa with initial temperatuer at 2700, 2800, 2900, 3000, and 3100 K as a parameter (pure tungsten).

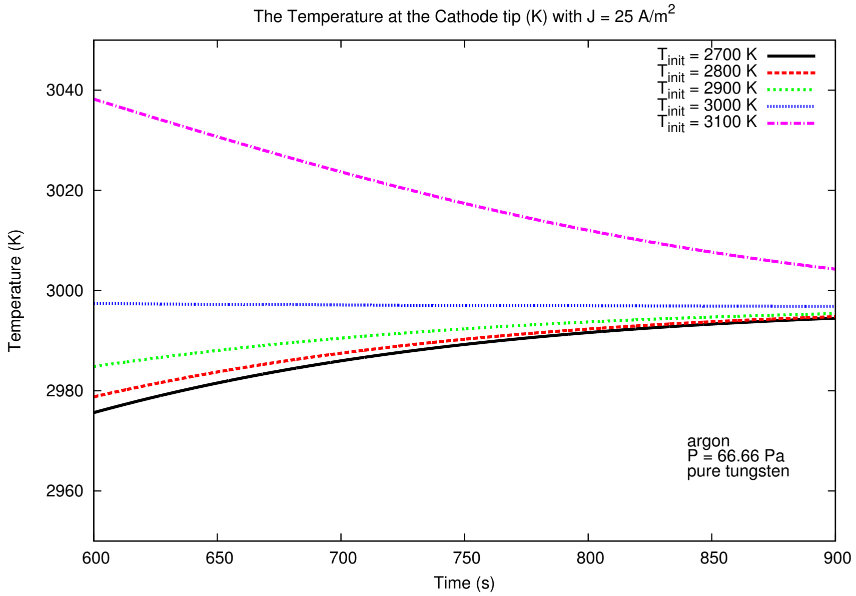

Figure 7.4: The cathode tip temperature [zoom in] as a function of time for a pressure of 66 Pa with initial temperatuer at 2700, 2800, 2900, 3000, and 3100 K as a parameter converged after 900 s.

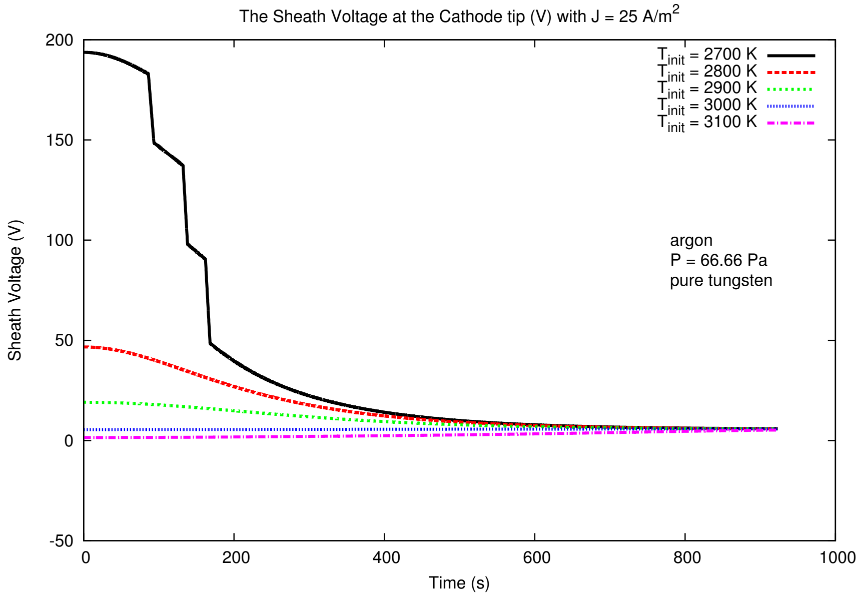

Figure 7.5: The sheath voltage at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 2700, 2800, 2900, 3000, and 3100 K as a parameter (pure tungsten).

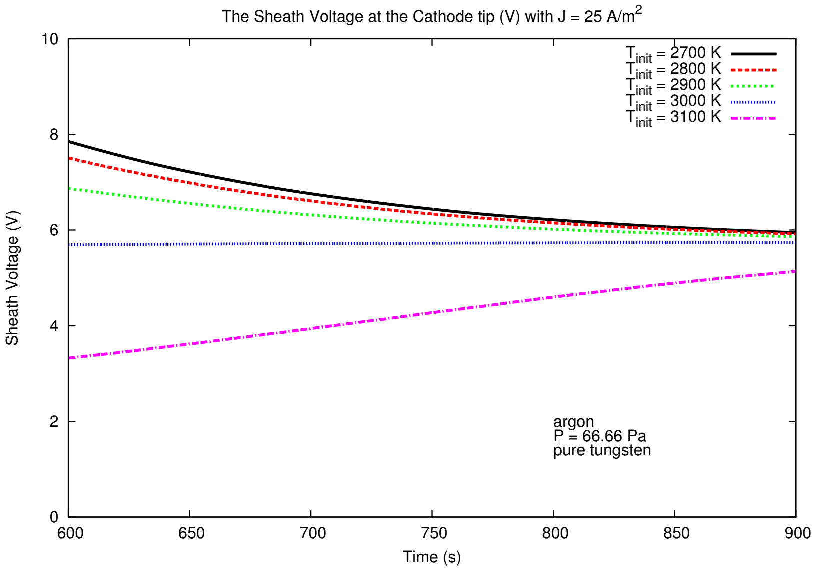

Figure 7.6: The sheath voltage [zoom in] at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 2700, 2800, 2900, 3000, and 3100 K as a parameter converged after 900 s.

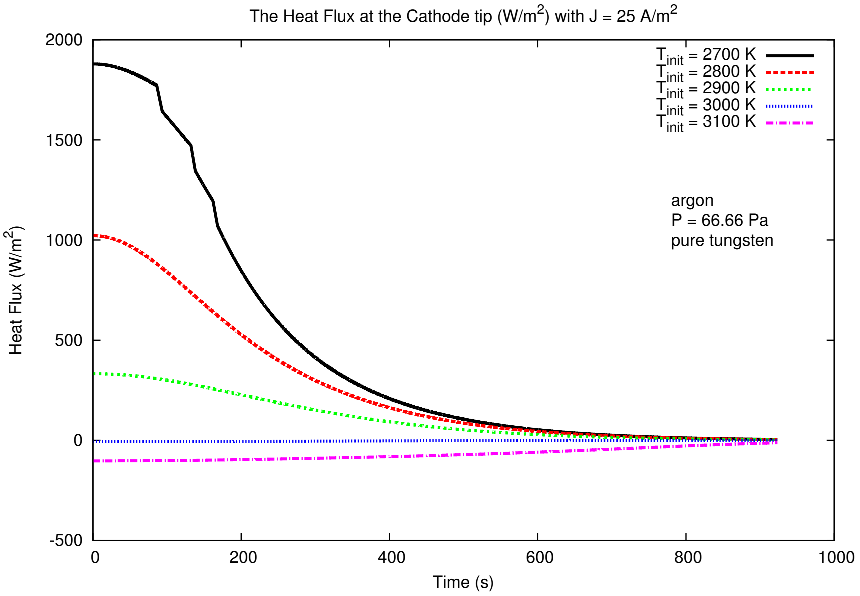

Figure 7.7: The heat flux at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature as a parameter (pure tungsten).

The heat flux with range of initial temperature from 2700 to 3100 K also converged a steady state after 900 s. The value of heat flux is nearly or less than zero, that means the cathode at steady state provide the heat flux to the plasma region. This process might be due to the ohmic heating from the cathode region that generated much rapidly than the process of heat conduction in the plasma region. The initial temperatue from 2700 - 2900 K provide the heat flux from the plasma region to the cathode region as the heat flux is positive. However, the heat flux becomes negative around 3000 K and above as can be seen Fig.7.7 and 7.8.

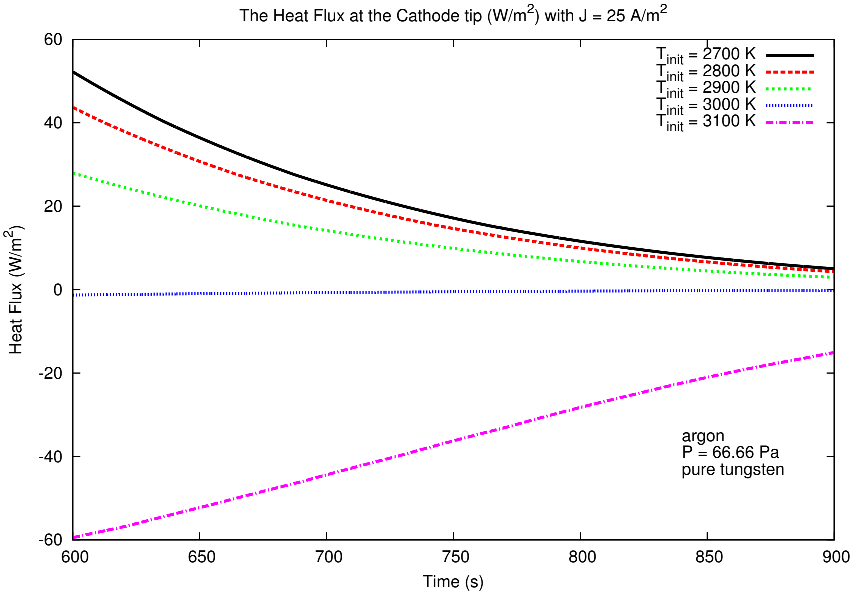

Figure 7.8: The heat flux [zoom in] at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature as a parameter start to converged after 900 s.

The cathode region temperature trends show in Fig.7.9 and 7.10 with initial temperature of 2800 K and current density of 25 A/m^2 and time ranged from 0, 300, and 900 s. As can be seen in Fig.7.10, the electrical conductivity in the cathode region conducts the electricity very well so the voltage in the cathode region is almost the same. The 1D MPD thruster potential presents in Fig.7.11 with the sheath voltage at the cathode tip. The anode is set to be 0 V and the voltage in plasma region increases with more negative value until at the grid NC + 1. The sheath voltage between NC and NC + 1 was included and at the cathode tip (NC). The zoom in for cathode region and plasma region can be viewed separately in Fig. 7.12 and 7.13.

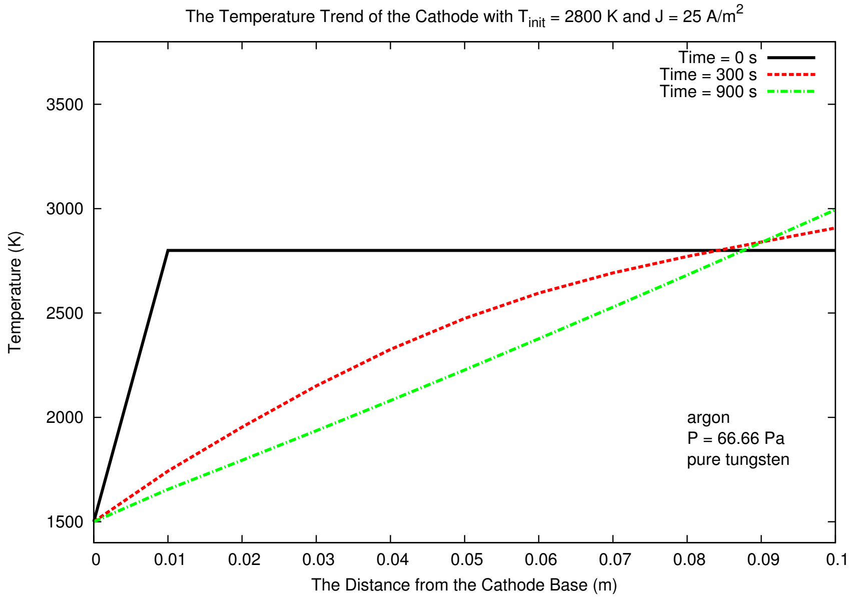

Figure 7.9: The cathode region temperature trends as a function of the distance from the cathode base with time as a parameter (pure tungsten) for initial temperature at 2800 K and current density at $25A/m^2$.

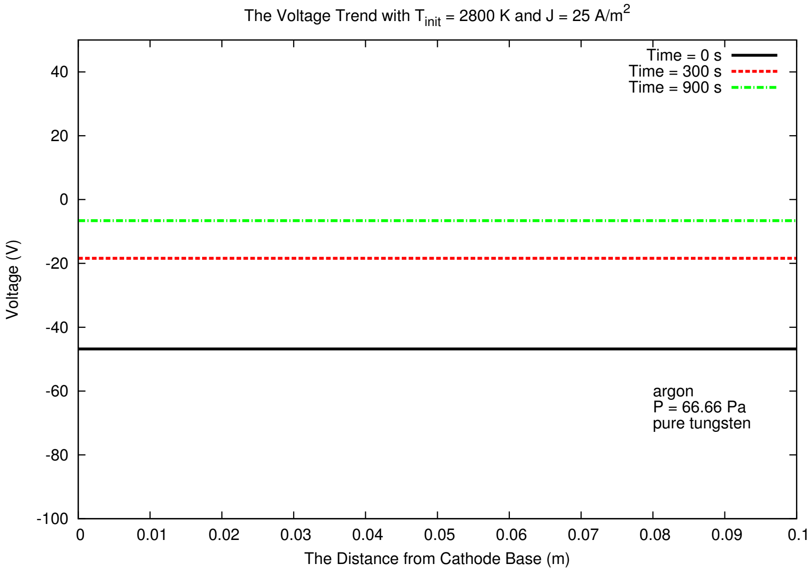

Figure 7.10: The cathode region voltage trends as a function of the distance from the cathode base with time as a parameter (pure tungsten) for initial temperature at 2800 K and current density at $25A/m^2$.

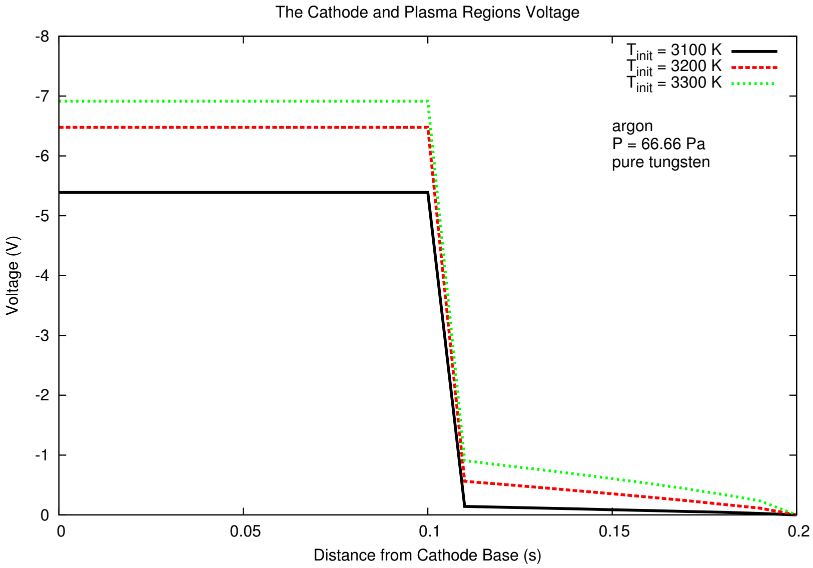

Figure 7.11: The voltage of 1D MPD thruster with the cathode and plasma regions as a function of the distance from the cathode base with initial temperature at 3100, 3200, and 3300 K and anode potential = 0 V.

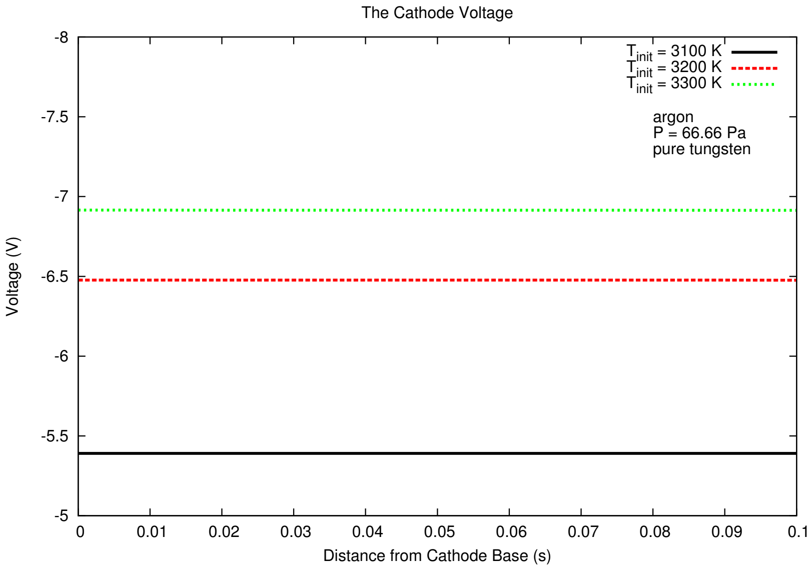

Figure 7.12: The steady state of cathode voltage region as a function of distance from the cathode base with initial temperature at 3100, 3200, and 3300 K with current density of 93, 169, and 297 A/m^2 repectively as a parameter.

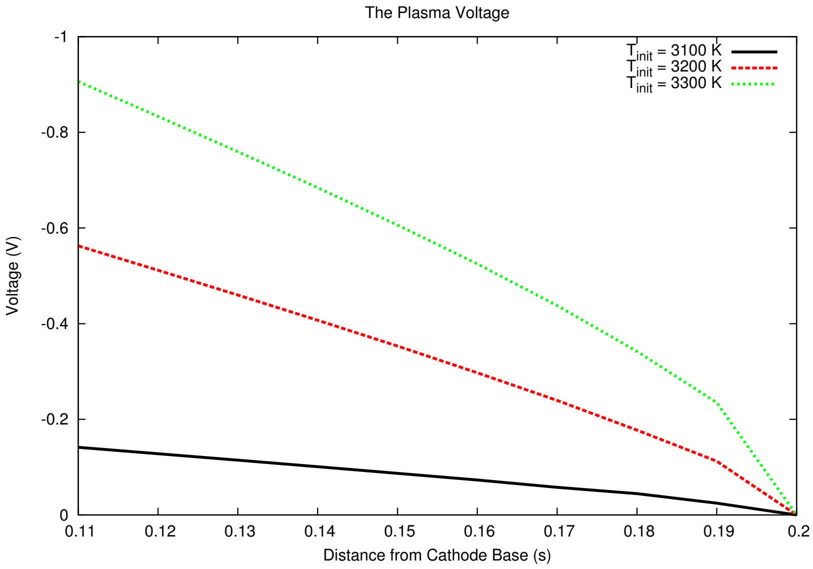

Figure 7.13: The steady state of plasma voltage region as a function of distance from the cathode base with initial temperature at 3100, 3200, and 3300 K with current density of 93, 169, and 297 A/m^2 repectively as a parameter.

The current density must precisely determine along with the initial temperature. The comparison between different initial temperature and current density are shown in Fig.7.14. That is, only change the current density 1 A/m^2, the heat flux changes significantly from positive to negative as can be seen in Fig.7.14 to 7.16.

The comparison of the sheath voltage with different initial temperature and current density are presented in Fig.7.17 to 7.19. As decribed before, the increased initial temperature increases the sheath voltage values.

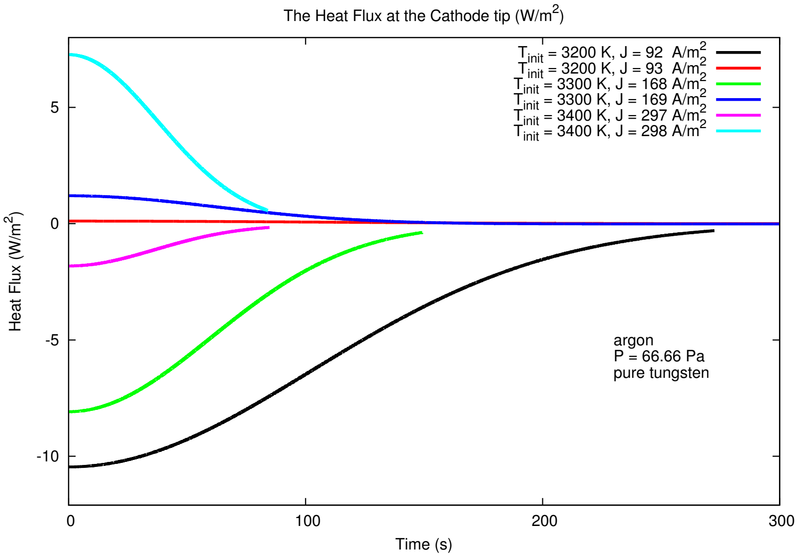

Figure 7.14: The heat flux at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 3200, 3300, 3400 K and current density as a parameter (pure tungsten).

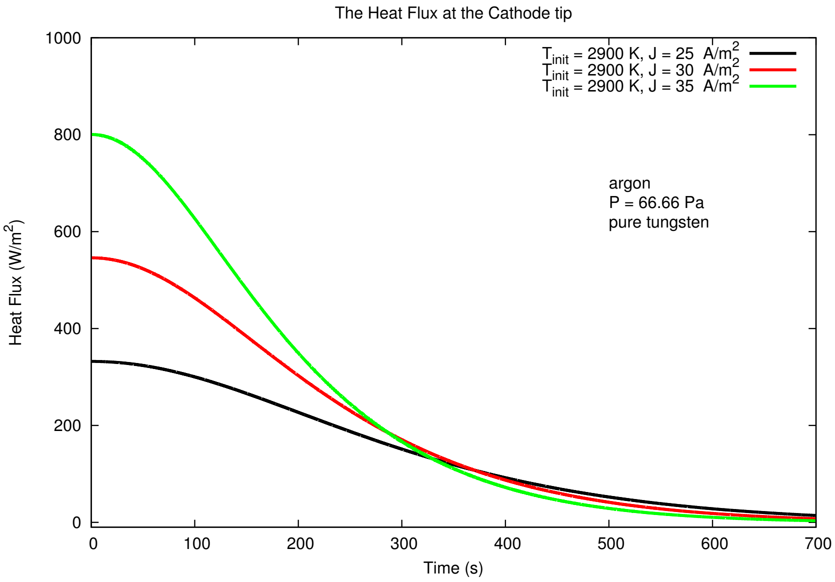

Figure 7.15: The heat flux at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 2900 K and current density as a parameter (pure tungsten).

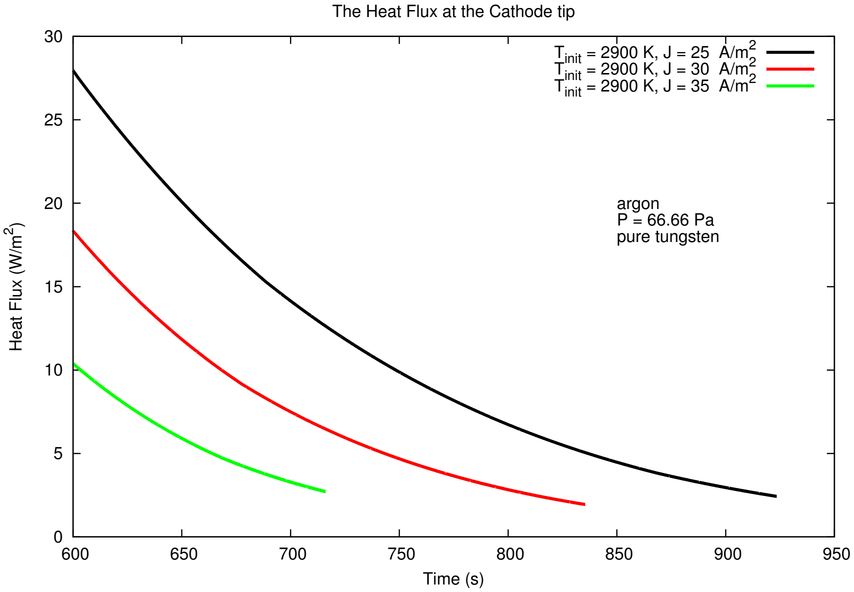

Figure 7.16: The heat flux [zoom in] at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 2900 K and current density as a parameter (pure tungsten).

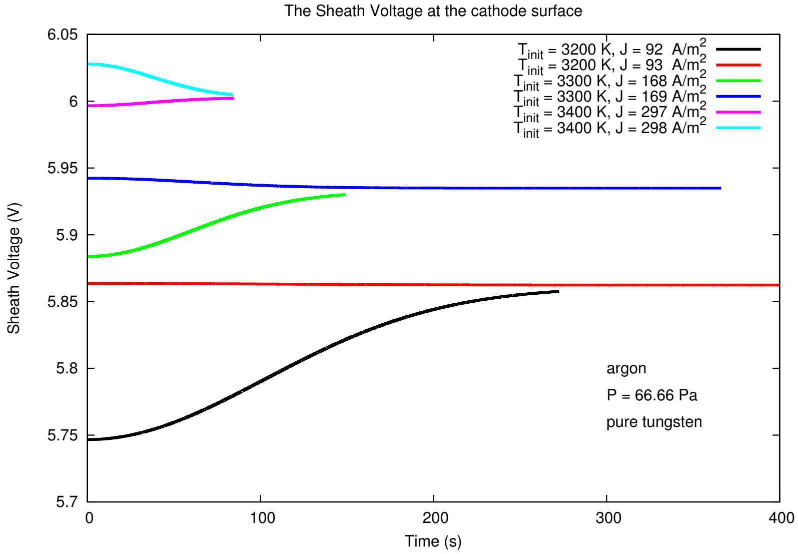

Figure 7.17: The sheath voltage at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 3200, 3000, and 3400 K and current density as a parameter (pure tungsten).

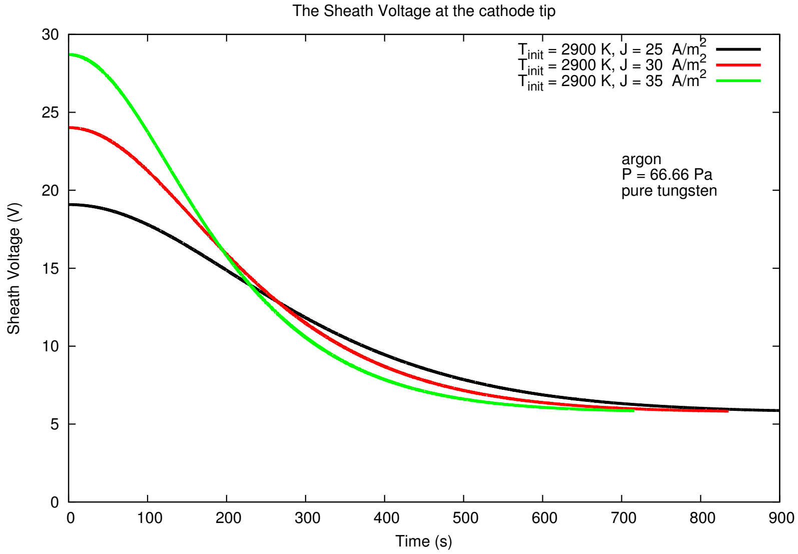

Figure 7.18: The sheath voltage at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 2900 K and current density as a parameter (pure tungsten).

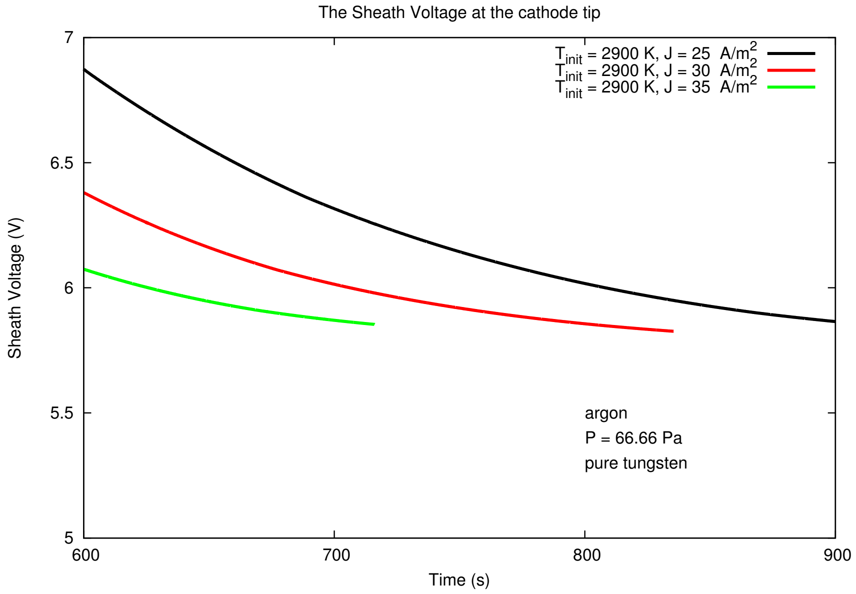

Figure 7.19: The sheath voltage [zoom in] at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 2900 K and current density as a parameter (pure tungsten).

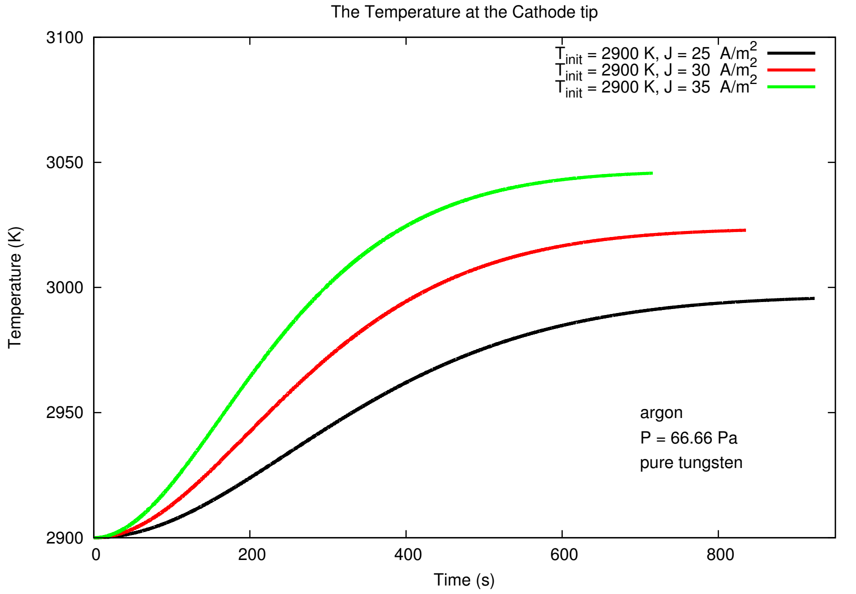

Figure 7.20: The temperature at the cathode tip as a function of time for a pressure of 66 Pa with initial temperature at 2900 K and current density as a parameter (pure tungsten).

In Fig.7.18, the current density increasing from 25, 30, and 35 A/m^2 also substantially changes the sheath voltage from 18, 23, and 29 V; however, the system reaches steady state with the sheath voltage around 6 V as can be viewed in zoom in Fig.7.19. By increasing the current density, the temperature significantly increases as shown in Fig.7.20.

7.3 Fully Combined Cathode and Plasma Regions with the Plasma Sheath Model in 2D Cylindrical Symmetry MPD Thruster

In chapter 4, the cartesian coordinate has been introduced and it can be applied for cylindrical symmetry. In this section, the detail of how to fully combined cathode and plasma regions together will be described.

The nonlinear heat diffusion equation can be applied with [1] for the 2D cylindrical symmetry and it can be written as

$$c_p(T)\frac{\partial T}{\partial t} = \frac{1}{r}\nabla \cdot \Big(rK(T)\nabla T\Big) + S \tag{7.11}$$

where S is the function of position or source term (i.e.thermionic heating), T is the temperature at the vertice of triangle, K(T) is a temperature dependent of thermal conductivity, $c_p(T)$ is a temperature dependent of heat capacity, s is the source terms i.e. Ohmic heating.

For steady state, we got

$$\nabla \cdot \Big(rK(T)\nabla T\Big) + S = 0 \tag{7.12}$$

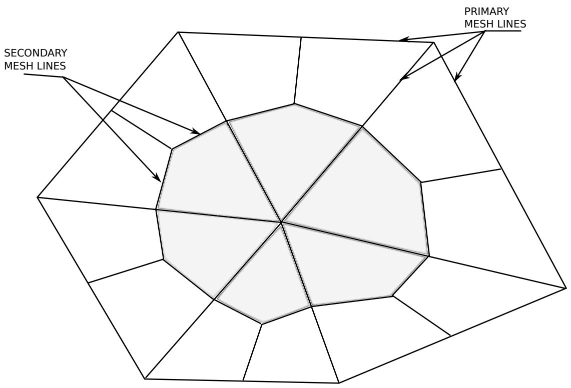

Figure 7.21: Primary and secondary mesh lines.

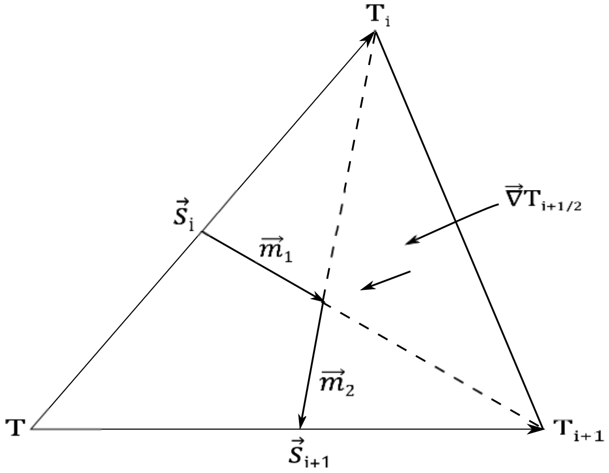

Figure 7.22: Vectors used in flux calculation.

The equation can be written in finite-difference approximation as

$$\frac{\Delta T}{\Delta t} = \frac{1}{G}\left[\sum_{i=1}^{6} w_i(T_i - T) + S\right] \tag{7.13}$$

For a steady state of the finite-difference analog of Eq.7.17 is

$$\sum_i w_i(T_i - T) + S = 0 \tag{7.14}$$

Again, using forward difference to the first derivative in time, which has a first order approximation $O(\Delta t)$ and it can be expresses as;

$$\frac{\Delta T_i}{\Delta t} = \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{7.15}$$

Then, Eq.7.17 can be written as

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{G}\left[\sum_{i=1}^{6} w_i(T_i - T) + S\right] \tag{7.16}$$

On the cathode surface (NC), the plasma sheath must be included so the grids can be written as

$$T_{NC}^{n+1} = T_{NC}^n + \frac{\Delta t}{G}\left[\sum_{i=1}^{6} w_i(T_i - T) + S + \dot{Q}a_{i+1/2}\right] \tag{7.17}$$

For the plasma grids next to the cathode surface(NC+1), the heat flux from the plasma sheath must be included as well so the grids can be written as

$$T_{NC+1}^{n+1} = T_{NC+1}^n + \frac{\Delta t}{G}\left[\sum_{i=1}^{6} w_i(T_i - T) + S - \dot{Q}a_{i+1/2}\right] \tag{7.18}$$

where

$$G = \textstyle\sum_{i=1}^{6} c_p(T)_{i+1/2}\bar{\bar{r}}_{i+1/2}a_{i+1/2},$$

$$S = \textstyle\sum_{i=1}^{6} j_{i+1/2}\bar{\bar{r}}_{i+1/2}a_{i+1/2},$$

$$\textstyle\sum_{i=1}^{6} w_i = \frac{1}{4}\sum_i \frac{K(T)_{i+1/2}}{A_{i+1/2}}\bar{\bar{r}}_{i+1/2}(\mathbf{s}_{i+1} - \mathbf{s}_i)^2,$$

$$\bar{\bar{r}}_{i+1/2} = \tfrac{7}{12}\bar{r}_{i+1/2} + \tfrac{5}{12}r,$$

$$\bar{r}_{i+1/2} = \tfrac{1}{3}(r + r_i + r_{i+1}).$$

7.3.1 Critical Time Increment of 2D Cylindrical Symmetry MPD Thruster Simulation

Von Neumann criteria

$$\Delta t \leq \frac{G}{2\sum_{i=1}^{6} w_i} \tag{7.19}$$

Ohmic heating criteria

$$\Delta t \leq \frac{G}{2S} \tag{7.20}$$

On the cathode surface, the heat flux critiria from the plasma sheath model must be included as can be shown below.

Heat Flux Sheath criteria

$$\Delta t \leq \frac{G}{2\dot{Q}a_{i+1/2}} \tag{7.21}$$

Convective criteria

$$\Delta t \leq \frac{G}{2h_c a_{i+1/2}} \tag{7.22}$$

Again, as can be seen in the above equations, Von Neumann criteria composes of the thermal conductivity and the heat capacity terms; however, the Ohmic heating criteria relates to the current density, electrical field, and the heat capacity terms. On the cathode surface, the heat flux sheath criteria relates to the heat capacity, the heat flux sheath and the area of the cells.

7.3.2 Outline of Algorithm of 2D Cylindrical Symmetry MPD Thruster Simulation

In this section, the 2D program will be described in several steps:

1. Given the initial temperature values of the cathode and plasma regions.

2. CALL the subroutines to obtain the electrical conductivity, thermal conductivity and heat capacity of both regions.

3. CALL the subroutine MAKEGRID to generate the cells as a triangular as decribed in chapter 4. Also, calculate for the average radius and the average radius of a quadrilateral. At this point, the area of each triangle can be obtained.

4. Given the total current of the system, the total current must be equal at the cathode base and anode as describe in chapter 4.

5. Set up the initial temperature at the grid points.

6. Calculate the critical time increment of the system.

7. Given the number of step to calculate in DO loop.

8. In the DO loop to solve for temperature

8.1 Calculate the average temperature at each triangular both in the cathode and in the plasma regions.

8.2 CALL the subroutine to update the electrical conductivity, thermal conductivity, and heat capacity of each triangle of both regions.

8.3 Calculate the total current for with the updated conductivity, thermal conductivity, and heat capacity.

8.4 Calculate the summation of electric field, current density, thermal conductivity, and heat capacity at the grid points.

8.5 Calculate the temperature using Eq.7.16 except for cathode surface grid points.

8.6 Calculate the temperature using Eq.7.17 for the cathode surface. The plasma sheath is included

9. Print out the temperature, voltage values.

7.3.3 2D Cylindrical Symmetry MPD Thruster Simulation Results

For the 2D cylindrical symmetry simulationl, the MPD system diameter parameters are set as followed: L_c = 7.5 m, L_a = 7.6 m, r_c = 2.0 m, r_a = 9.0 m, $T_{c_{init}}$ = 3300 K, $T_{e_{init}}$ = 9000 K, V_{init} = 100 V, I = 100 A. The first case has grids of N_1=6, N_2=4, N_3=2, N_4=4, N_5=2 and the second case has grids of N_1=24, N_2=8, N_3=10, N_4=20, N_5=8. However, some of these parameters can be changed as needed to see how an MPD thurster system respond.

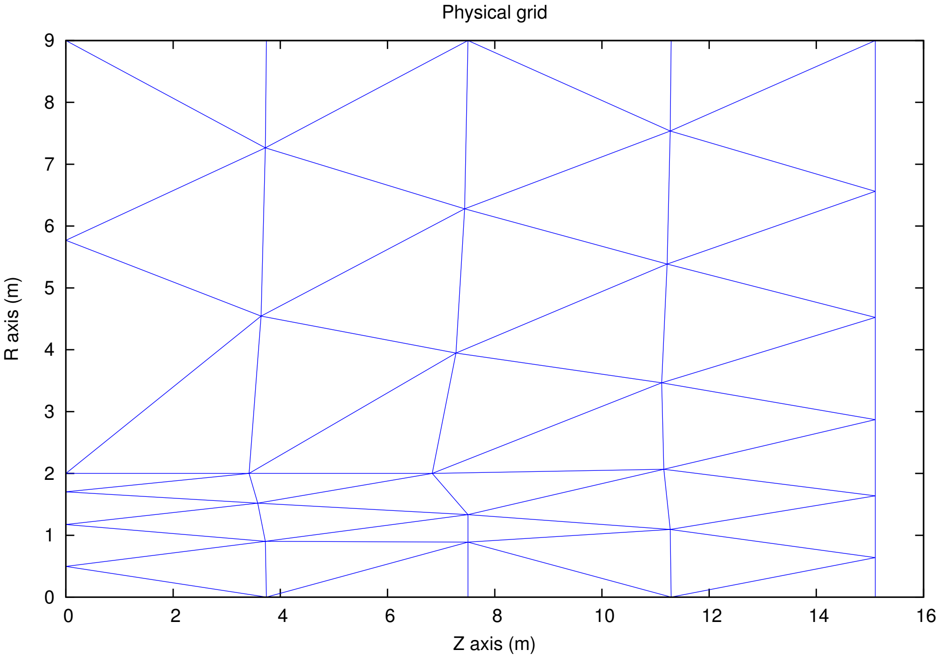

Figure 7.23: The physical grid of MPD Thruster with N_1=6, N_2=4, N_3=2, N_4=4, N_5=2.

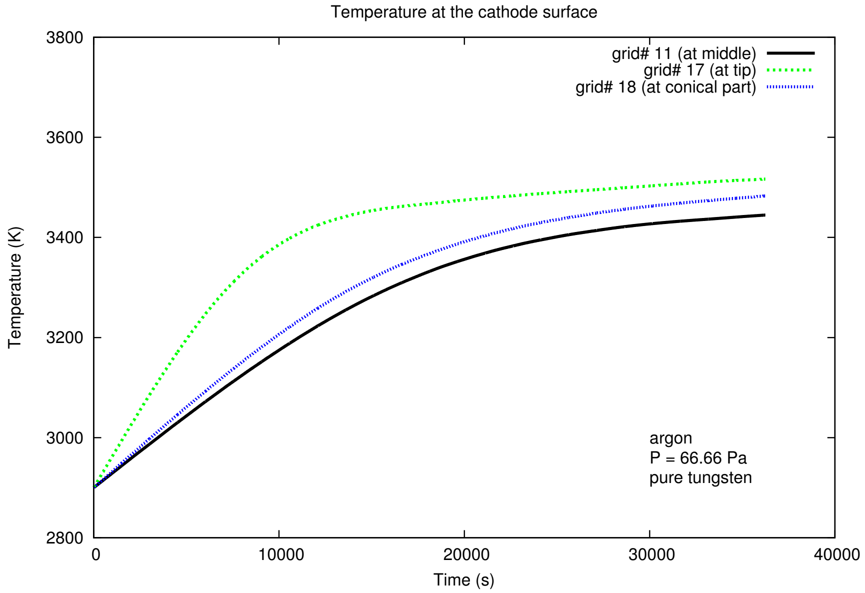

Figure 7.24: Temperature as a function of time with grid cathode surface location as a parameter in pressure of 66 Pa and pure tungsten.

The cathode surface temperature (at middle, conical tip, and tip) reach the plateau after 25000 s. As expected, the tip has a high temperature nearly 3500 K following at conical tip and at the middle of the cathode as can be viewed in Fig.7.24.

The maximum electric field value locates at the tip of the cathode and followed by at the conical tip and at the middle of the cathode. The electric field rapidly approach the steady state after 3000 s as in Fig.7.25.

The maximum current density value still locates at the cathode tip; however, the lowest current density value is at the conical tip. The reason of this result still unclear; however, it might be due to the distribution area of the grid at the conical tip.

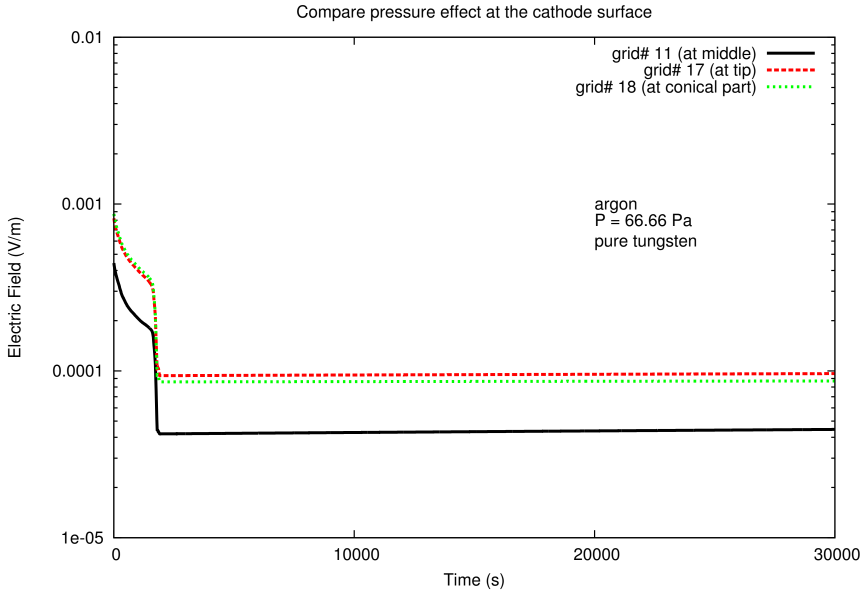

Figure 7.25: Electric field as a function of time with grid cathode surface location as a parameter in pressure of 66 Pa and pure tungsten.

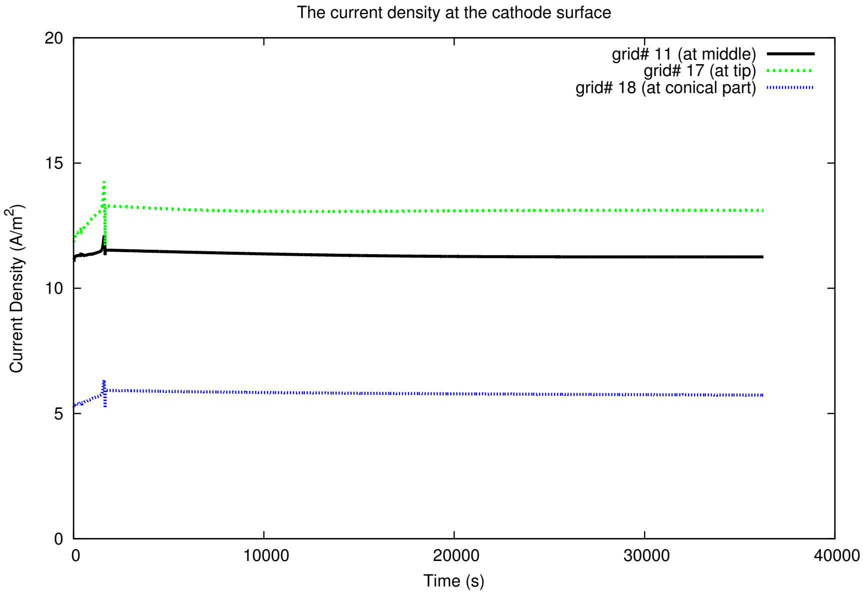

Figure 7.26: Current density as a function of time with grid cathode surface location as a parameter in pressure of 66 Pa and pure tungsten.

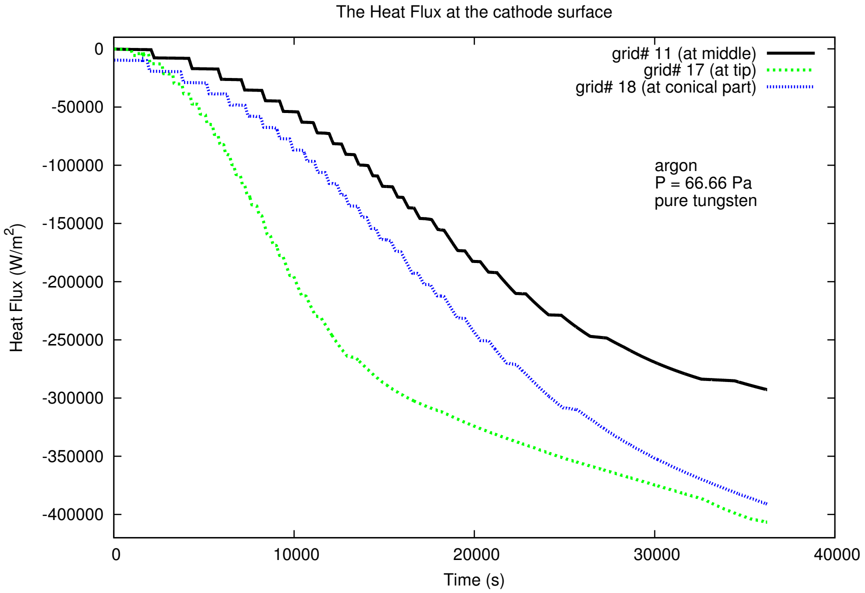

Figure 7.27: Heat flux as a function of time with grid cathode surface location as a parameter in pressure of 66 Pa and pure tungsten.

The heat flux at the cathode surface have a negative sign. This is due to the ohmic heating generating heat in cathode region must faster than the heat conduction from plasma to cathode regions. The heat flux has a direct relationship with the cathode temperature so the location of maximum to minimum are the same with temperature.

7.3.3.1 Convection Effect

The equation from previous section is still the same; however, the convective term carries the heat away from the cathode surface so this effect has a negative sign in Eq.7.17 and it becomes

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{G}\left[\sum_{i=1}^{6} w_i(T_i - T) + S + \dot{Q}a_{i+1/2} - h_c a_{i+1/2}(T_i^n - T_{inf}^n)\right] \quad (7.23)$$

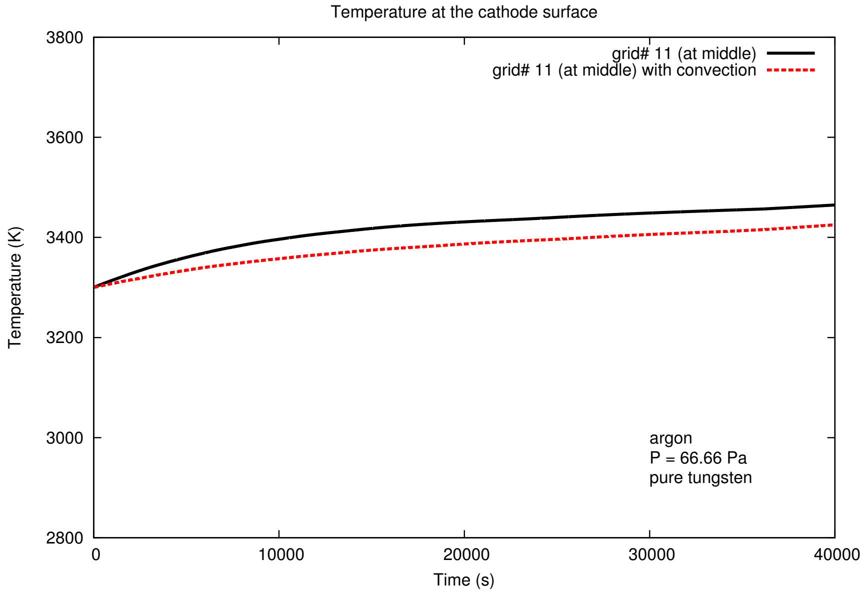

Figure 7.28: Temperature as a function of time with convection effect as a parameter in pressure of 66 Pa and pure tungsten.

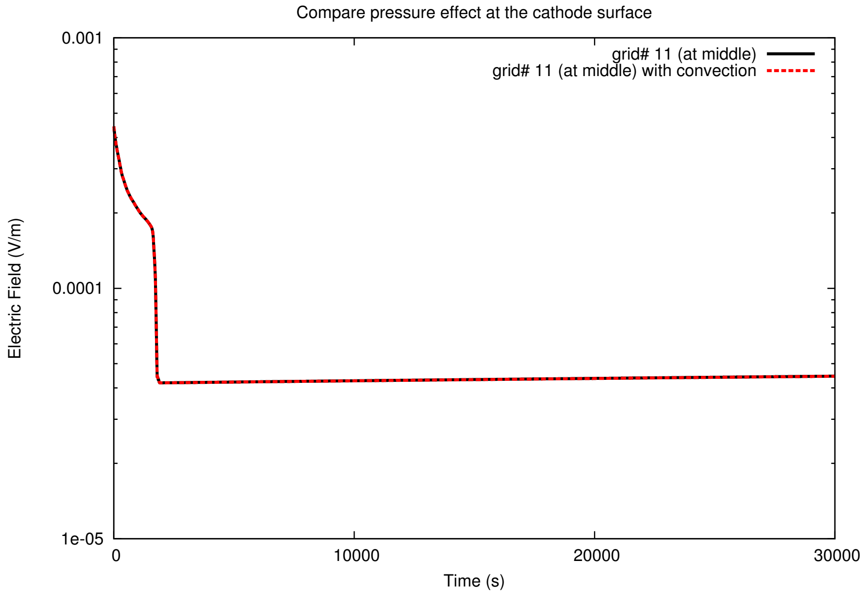

Figure 7.29: Electric field as a function of time with convection effect as a parameter in pressure of 66 Pa and pure tungsten.

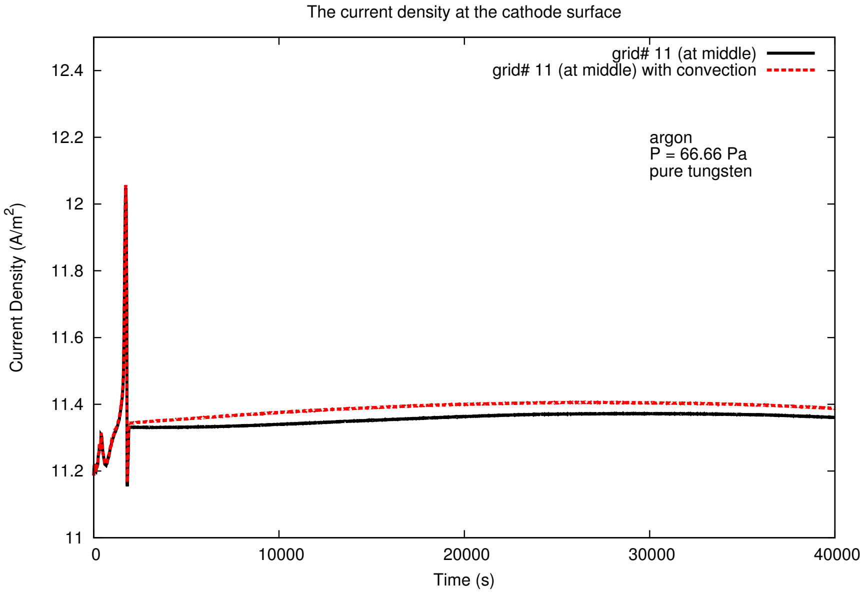

Figure 7.30: Current density as a function of time with convection effect as a parameter in pressure of 66 Pa and pure tungsten.

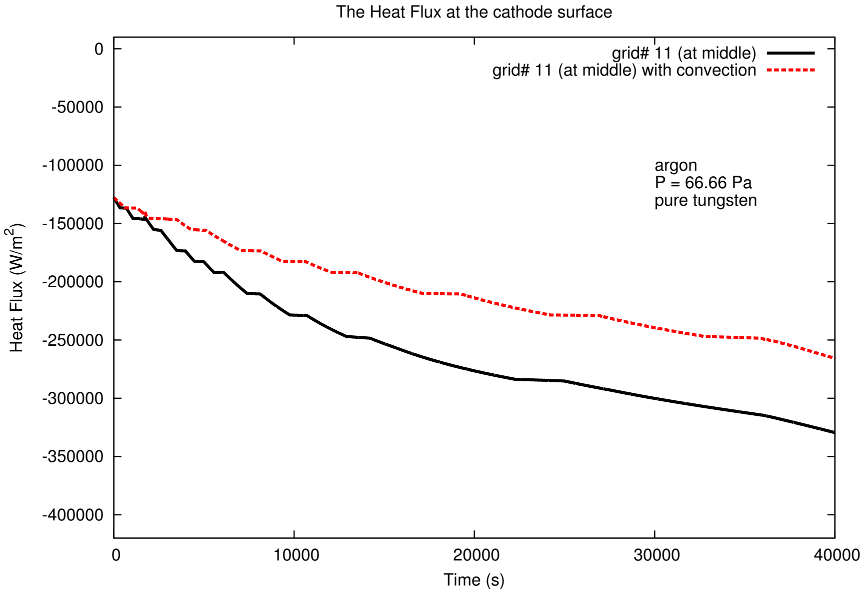

Figure 7.31: Heat flux as a function of time with convection effect as a parameter in pressure of 66 Pa and pure tungsten.

The convection effect tends to lower the temperature, and current density; however, electric field value does not change. As lower the cathode temperature, the heat flux also decreased as can be seen in Fig.7.28 - 7.31.

7.3.3.2 Pressure Effect

The plasma properties in different pressure have a significant impact on electrical conductivity, thermal conductivity and heat capacity respectively. We will compare pressure of 10^5 and 10^7 Pa with our pressure at 66 Pa to see the pressure effect. The properties of interested are temperature, electric field and the current density in our 2D cylindrical symmetry. The equation is thes same as in the case of without convection effect, Eq.7.17 but the electrical, thermal conductivities and heat capacity values will be adjusted for 10^5 and 10^7 Pa conditions.

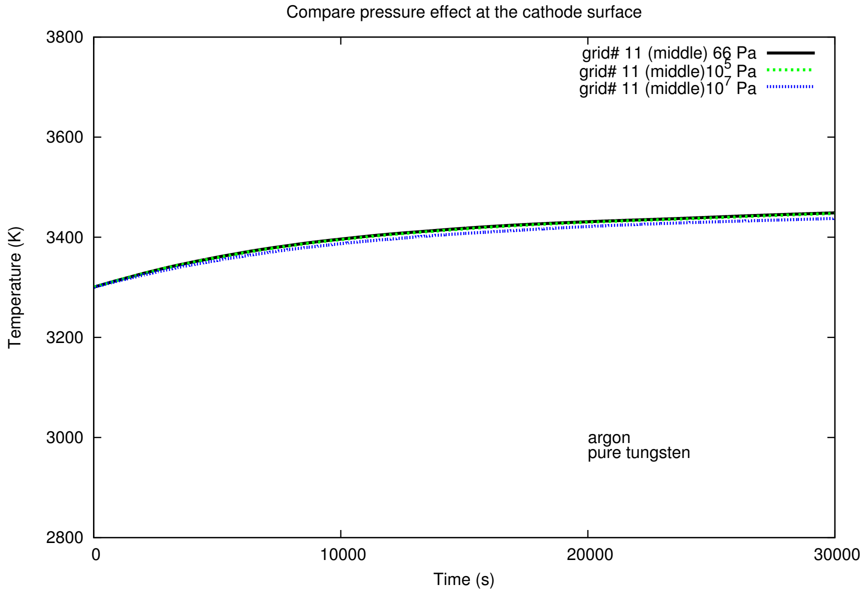

Figure 7.32: Temperature as a function of time with pressure as a parameter in pressure of 66 Pa and pure tungsten.

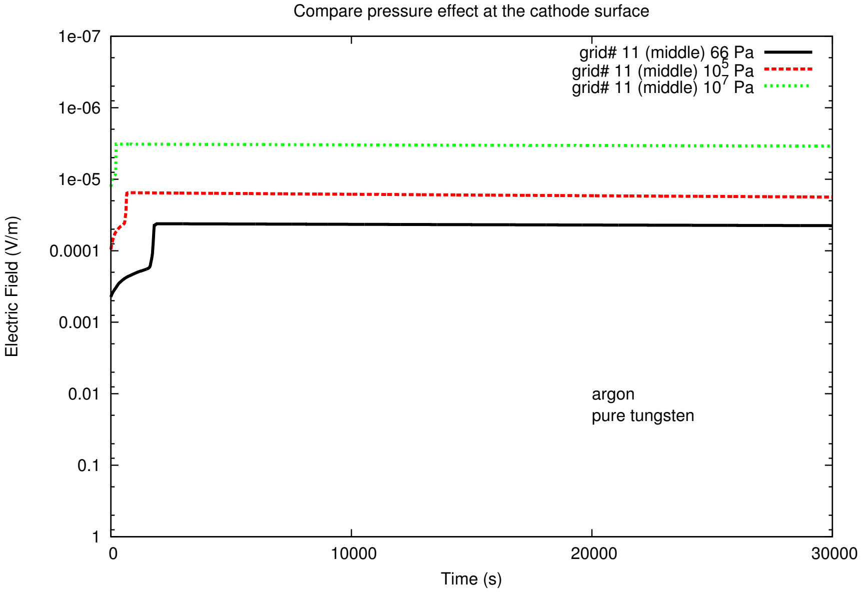

Figure 7.33: Electric field as a function of time with pressure as a parameter in pressure of 66 Pa and pure tungsten.

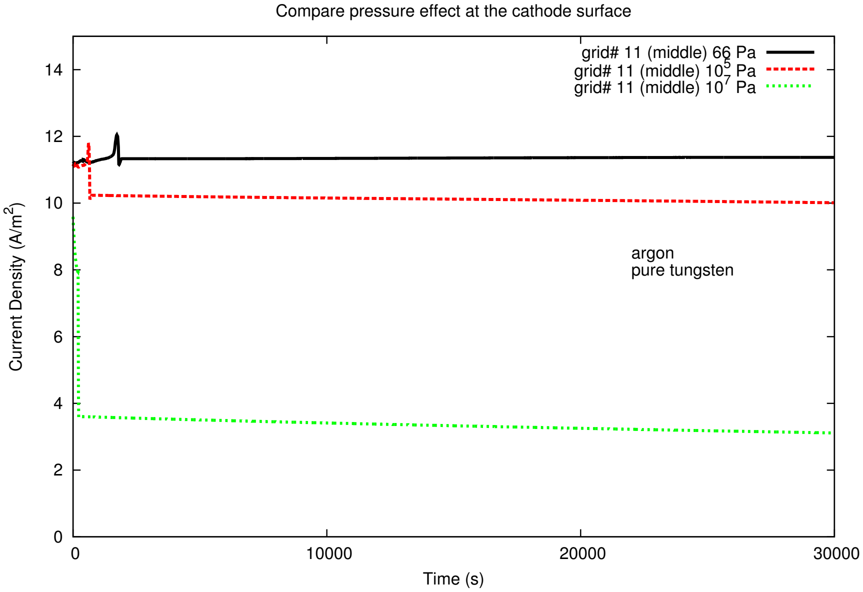

Figure 7.34: Current density as a function of time with pressure as a parameter in pressure of 66 Pa and pure tungsten.

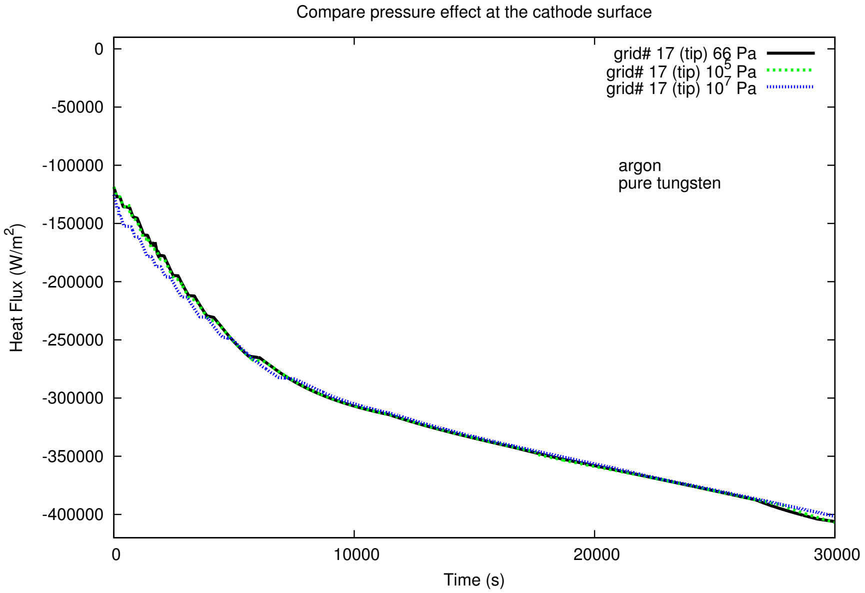

Figure 7.35: Heat flux as a function of time with pressure as a parameter in pressure of 66 Pa and pure tungsten.

The temperature and heat flux influence very small amount as pressure changed in Fig.7.32 and 7.35. The electric field and current density values reduced as the pressure increased as can be shown in Fig.7.33 and 7.34.

For N1=24, N2=8, N3=10, N4=20, N5=8, the physical grids, electrical, thermal conducitivity and heat capacity can be viewed as;

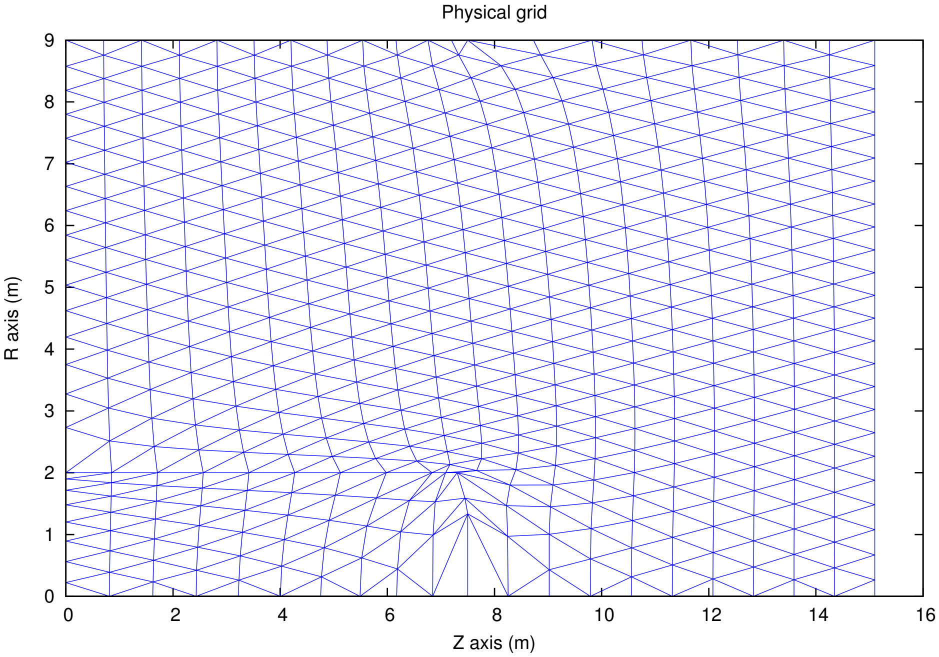

Figure 7.36: The physical grids space of MPD Thruster Simulation with N1=24, N2=8, N3=10, N4=20, N5=8.

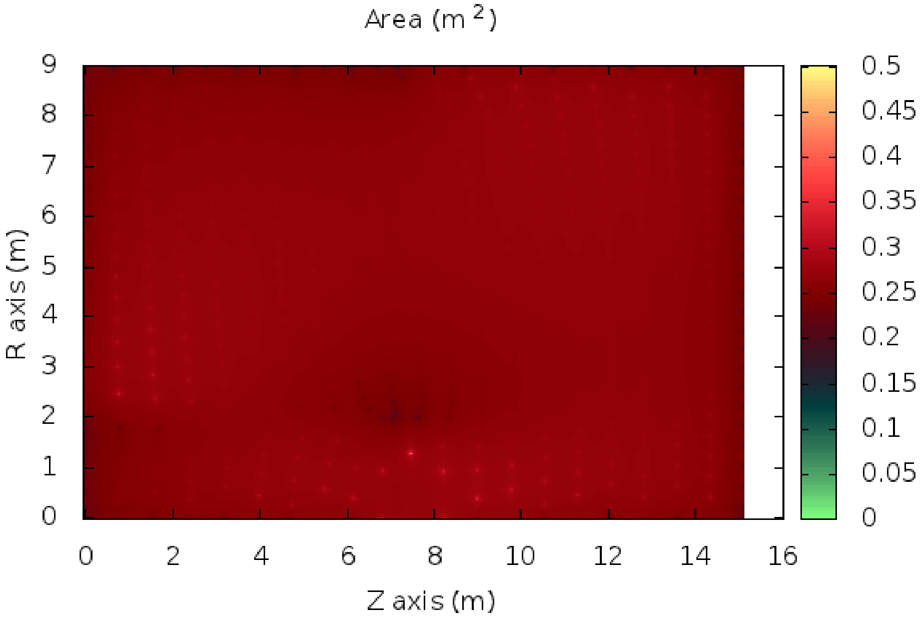

Figure 7.37: The area size distribution of physical grids space with N1=24, N2=8, N3=10, N4=20, N5=8.

The parameters size of MPD thruster are greater than the previous experimental data as to simulate the actual size of the MPD thruster to use in spacecraft. The area size distribution of the computation grids distributed almoste equally in the cathode and the plasma regions as can be seen in the color distribution. However, at the cathode tip in the plasma region, the size of the grids are smaller than others.

The temperature, electrical conductivity, thermal conductivity and heat capacity distribution of cathode and plasma regions shown in Fig.7.38 - 7.41. The electrical conductivity, thermal conductivity and heat capacity are as expected to has a greater value in the cathode region than in the plasma region.

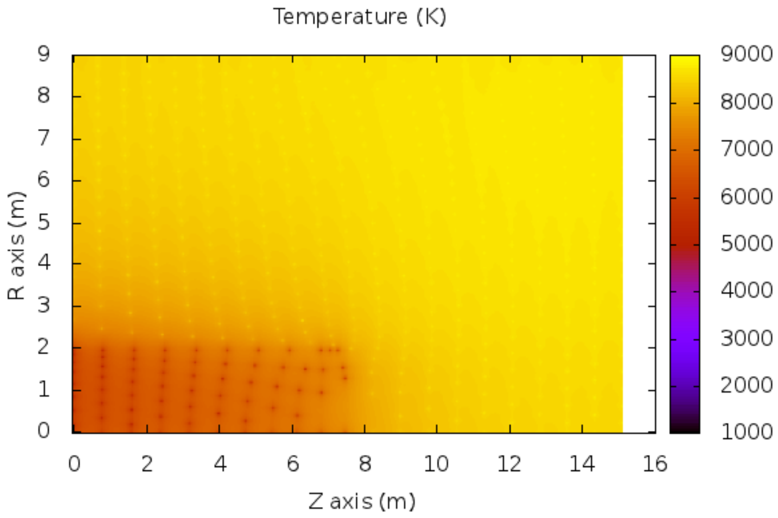

Figure 7.38: The temperatuer distribution of cathode and plasma regions in MPD thruster.

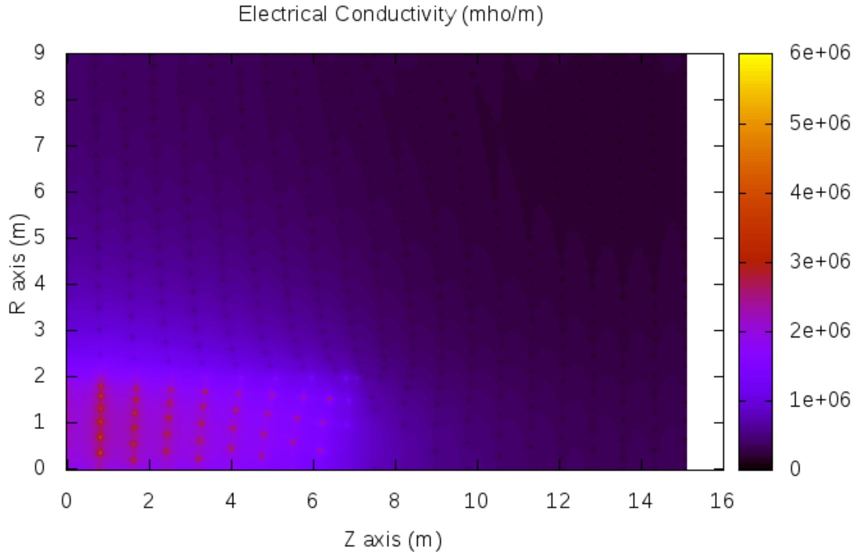

Figure 7.39: The electrical conductivity distribution of cathode and plasma regions in MPD thruster.

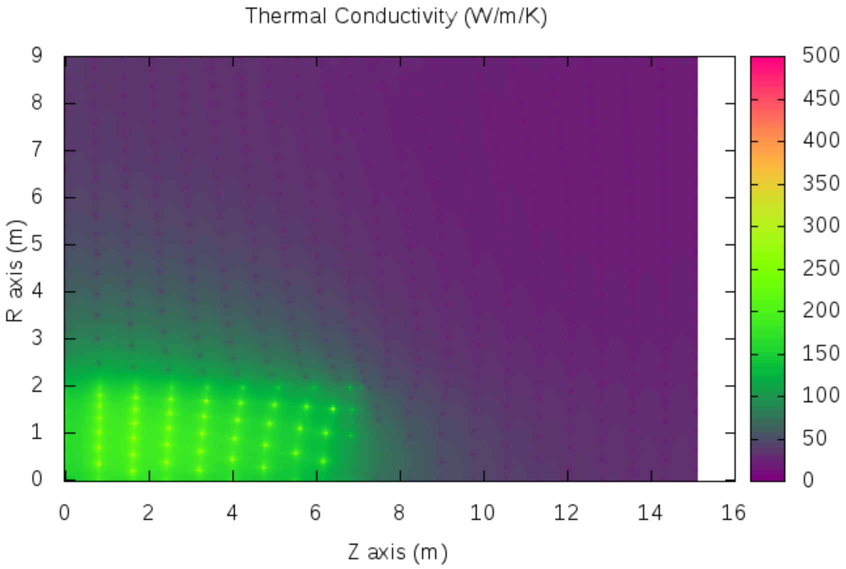

Figure 7.40: The thermal conductivity distribution of cathode and plasma regions in MPD thruster.

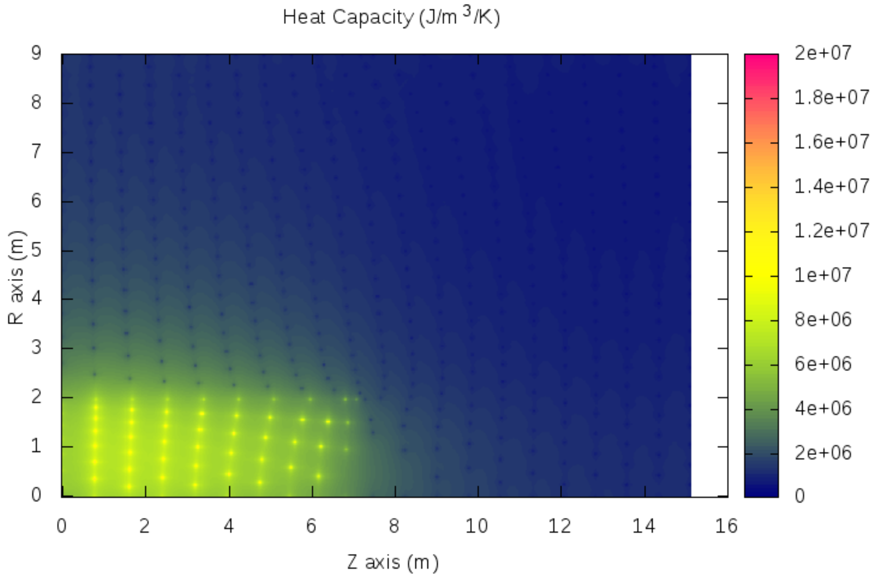

Figure 7.41: The heat capacity distribution of cathode and plasma region in MPD thruster.

7.3.3.3 Electroarc Edge

The electroarc edge is definded as the location where the plasma arc attach the cathode surface. To locate this point, the electric field along the cathode surface must be calculated to find the minimum values of electric field in Fig.7.42.

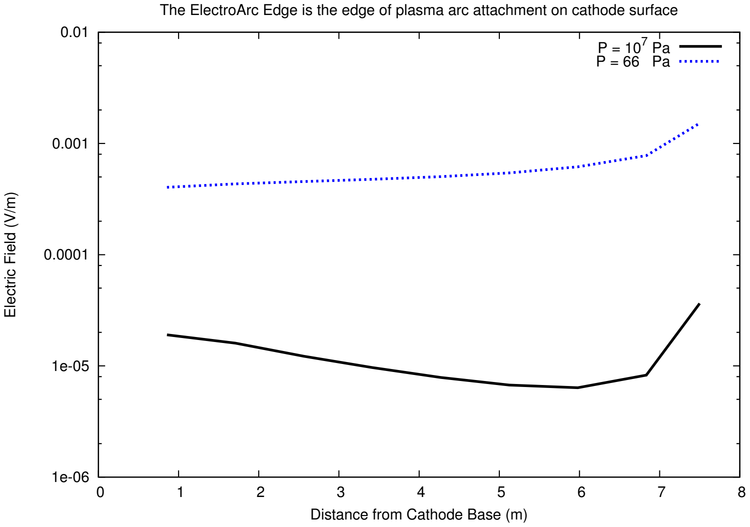

Figure 7.42: The minimum value of electric field on the cathode surface in MPD thruster is definded as the edge of plasma arc attachment, Electroarc Edge.

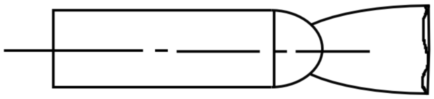

Figure 7.43: Illustration of Electroarc Edge in High Pressure [3].

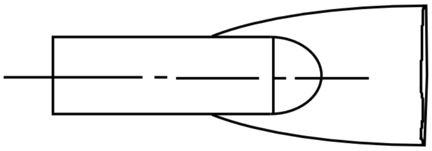

Figure 7.44: Illustration of Electroarc Edge in Low Pressure [3]

The experiment of MPD thruster showed the relationship between the location of this electroarc edge; however, the exact location could not be validated as can be seen in Fig.7.43 and 7.44. This research was be able to locate exactly where the electroarc edge would be. The electroarc edge is approximately at the cathode tip with pressure $10^7 Pa$ and the electroarc edge move toward the cathode base. These numerical simulation results agree well with the experimental observation [3].

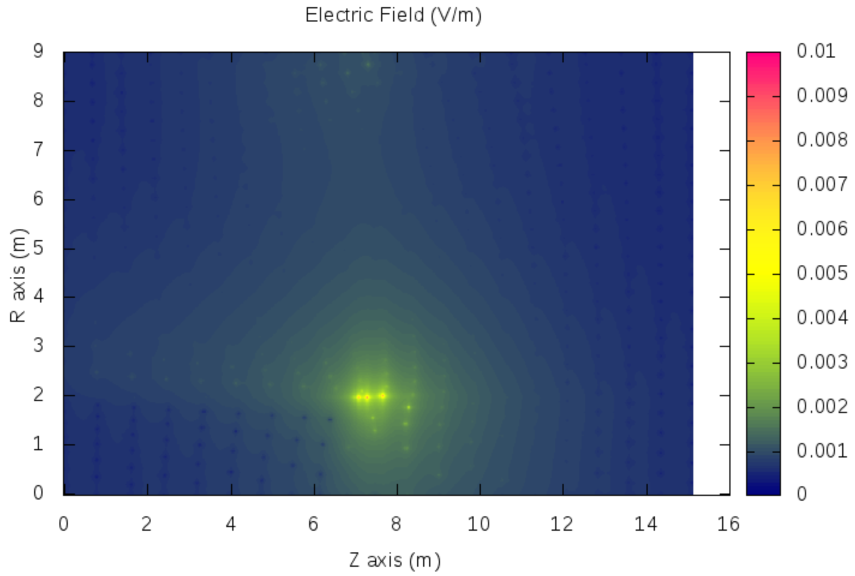

Figure 7.45: The electric field distribution of cathode and plasma regions in MPD thruster.

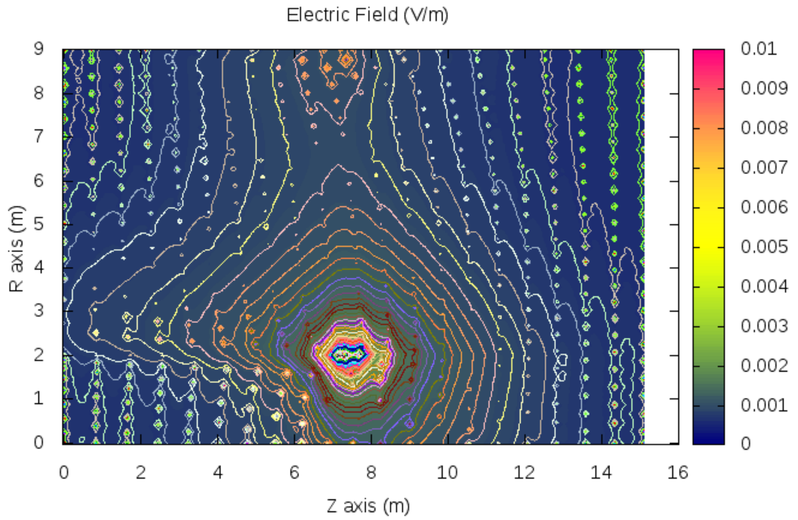

Figure 7.46: The electric field contour of cathode and plasma regions in MPD thruster.

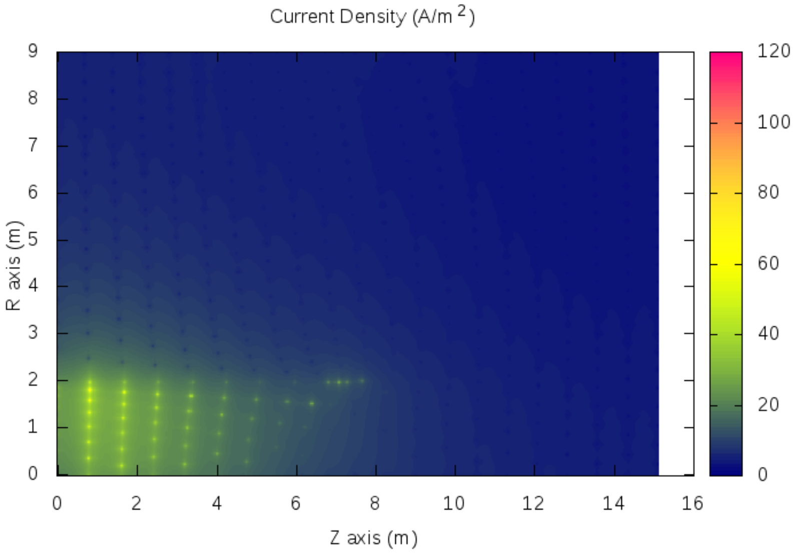

Figure 7.47: The current density distribution of cathode and plasma in MPD thruster.

The electric field distribution, electric field contour and current density distribution can be seen in Fig.7.45 - 7.47. The maximum of electric field value locates at the tip surface.

7.4 Recommendations for Future Work

The future study could improve the 2D cylindrical symmetry MPD thruster simulation to include the flow of particles such as electrons and ions as in particle-in-cell or Direct Simulation Monte Carlo (DSMC) within the system [57]. For this study, the time and financial were not allowed to calculate and include the flow of the particles model in 2D cylindrical simulation. In addition, the model could be further developed to included the erosion effects base on this 2D cylindrical symmetry. Furthermore, this 2D cylindrical symmetry MPD thruster simulation can be improved by calculating the magnetic field, and the thrust in an MPD thruster to fully see how the system response with this magnetic field.

Chapter 8

Conclusions

The objective of this work were to improve the computation simulations and explain the interaction between the cathode and the plasma regions with the plasma sheath model [3] in an MPD thruster. These simulations should help describing the experimental phenomenon as well as predicting how an MPD thruster respond to the operational changed.

In this study, a 1D, and a 2D cylindrical symmetry MPD thruster simulations have been developed to predicting the temperature, the voltage, the current density, the electric field with the surrounding plasma in an MPD thruster and estimated the plasma attachment edge, electroarc edge.

The 1D MPD thruster simulation consists with three regions: the cathode, the plasma, and the interface of these regions, the plasma sheath model. The interface provides the boundary values to the cathode in steady state. The criteria of the plasma sheath model is limited with certain temperature and current density. At the cathode temperature below 2500 K, the heat flux transfers from plasma to the cathode. Once the cathode reaches around 2700 K, the heat flux transfer from cathode to plasma region. In other words, the plasma region heats up by the cathode region. Moreover, the current density also significantly impact on the cathode temperature as increased the current density, the cathode can heat up rapidly.

As a results, we need to consider the criteria of the plasma sheath model to pin point where we need to obtain the cathode temperature and current density. This 1D MPD thruster simulation uses the plasma sheath mode, by given the cathode temperature and current density of the cathode surface, the plasma sheath model provides the heat flux and the plasma sheath to the cathode surface. Then, the 1D simulation balances the energy of cathode region with the plasma region until it reaches steady state for temperature, sheath voltage and heat flux.

For the 2D cylindrical symmetry MPD thruster simulation, temperature, potential, current density, electric field can be calculated and this 2D simulation also can estimated the electroarc edge, which is definded as the edge of the plasma arc attachement at the minimum electric field value on the cathode surface. In the numerical results of low pressure, the minimum value of electric field locates further to near the cathode base as obaserved in experiment [3]. In addition, the minimum electric field value moves toward the cathode tip for high pressure condition as decribed [3]. The electroarc edge results using the criteria of minimal value of electric field on the cathode surface agree well with the experimental observation.

In conclusive, this work was achieved the objectives that is to improve the MPD thruster simulations and explain the interaction of the cathode, the plasma with the plasma sheath model and predict the electroarc edge on the cathode surface in MPD thruster.

Bibliography

[1] A. M. Winslow, "Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh," *Journal of Computational Physics*, vol. 1, no. 2, 1966.

[2] D. A. Codron and D. Erwin, "Experimental studies on high current arc discharges for magnetoplasmadynamic thrusters," in *Aerospace Conference, 2012 IEEE*, IEEE, 2012.

[3] K. D. Goodfellow, *A theoretical and experimental investigation of cathode processes in electric thrusters.* PhD thesis, University of Southern California, 1996.

[4] W. F. Ahtye, *A critical evaluation of methods for calculating transport coefficients of partially and fully ionized gases.* National Aeronautics and Space Administration, 1965.

[5] E. Niewood and M. Martinez-Sanchez, "Quasi-one-dimensional numerical simulation of magnetoplasmadynamic thrusters," *Journal of Propulsion and Power*, vol. 8, no. 5, 1992.

[6] C. K. J. Hulston, P. J. Redlich, W. R. Jackson, M. Marshall, F. P. Larkins, R. C. Mehta, S. Andrews, and P. V. Ramachandran, "Thermal erosion of magnetoplasmadynamic thruster cathode," *International journal of heat and mass transfer*, vol. 39, no. 8, 1996.

[7] J. Rossignol, S. Clain, and M. Abbaoui, "The modelling of the cathode sheath of an electrical arc in vacuum," *Journal of Physics D: Applied Physics*, vol. 36, p. 1495, 2003.

[8] NASA, *Statement by NASA Administrator Charlie Bolden.* Public document, 2014.

[9] R. G. Jahn and W. von Jaskowsky, *Physics of electric propulsion*, vol. 288. McGraw-Hill, 1968.

[10] R. W. Humble, G. N. Henry, W. J. Larson, U. S. of Defense, U. S. Aeronautics, and S. Administration, *Space propulsion analysis and design*. McGraw-Hill, 1995.

[11] G. P. Sutton and O. Biblarz, *Rocket propulsion elements*. Wiley, 2011.

[12] J. D. Mattingly and H. von Ohain, *Elements of propulsion: gas turbines and rockets*. American Institute of Aeronautics and Astronautics, 2006.

[13] V. P. Friedensen, "Space nuclear power: Technology, policy, and risk considerations in human missions to Mars," *Acta astronautica*, vol. 42, no. 1-8, 1998.

[14] P. G. Mikellides, "Modeling and analysis of a megawatt-class magnetoplasmadynamic thruster," *Journal of propulsion and power*, vol. 20, no. 2, 2004.

[15] K. Toki, Y. Shimizu, and K. Kuriki, "Application of MPD thruster systems to interplanetary missions," *Journal of Propulsion and Power*, vol. 2, 1986.

[16] A. Sasoh, A. Solem, and Y. Arakawa, "10 kW steady-state MPD thruster," *Tokyo, University, Faculty of Engineering, Journal, Series B*, vol. 39, 1988.

[17] J. E. Polk, *Mechanisms of cathode erosion in plasma thrusters*. PhD thesis, Princeton University, 1996.

[18] M. I. Boulos, P. Fauchais, and E. Pfender, *Thermal plasmas: fundamentals and applications*, vol. 1. Springer, 1994.

[19] D. Pramod, *Numerical methods in engineering*. Chulalongkorn, 2012.

[20] D. A. Erwin and J. A. Kunc, "Electron temperature and ionization degree dependence of electron transport coefficients in monatomic gases," *Physics of Fluids*, vol. 28, p. 3349, 1985.

[21] D. A. Erwin and J. A. Kunc, "Scalar DC electrical conductivity of partially ionized gases," *Computer physics communications*, vol. 42, no. 1, 1986.

[22] S. Paik and E. Pfender, "Argon plasma transport properties at reduced pressures," *Plasma chemistry and plasma processing*, vol. 10, no. 2, 1990.

[23] G. J. Dunn and T. W. Eagar, "Calculation of electrical and thermal conductivities of metallurgical plasmas," *Bull./Welding research council*, 1990.

[24] H. Kawaguchi, K. Sasaki, H. Itoh, and T. Honma, "Numerical study of the thrust mechanism in a two-dimensional MPD thruster," *International Journal of Applied Electromagnetics and Mechanics*, vol. 6, no. 4, 1995.

[25] T. Miyasaka and T. Fujiwara, "Numerical analyses of 2-dimensional axisymmetric MHD flows satisfying sonic conditions in an MPD thruster," *Japan Society for Aeronautical and Space Sciences, Journal*, vol. 41, no. 133, 1998.

[26] J. Thompson, "Introduction to"Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh"," *Journal of Computational Physics*, vol. 135, 1997.

[27] C. K. Birdsall and A. B. Langdon, *Plasma physics via computer simulation.* Egully. com, 2004.

[28] M. Auweter-Kurtz, C. Boie, H. J. Kaeppeler, H. L. Kurtz, H. O. Schrade, P. C. Sleziona, H. P. Wagner, and T. Wegmann, "Magnetoplasmadynamic thrusters: design criteria and numerical simulation," *International Journal of Applied Electromagnetics in Materials*, vol. 4, 1994.

[29] S. Sawai, H. Igarashi, T. Honma, K. Toki, and K. Kuriki, "An electromagnetic field analysis of two-dimensional MPD arcjet thrusters using the boundary element method.," *INT. J. APPL. ELECTROMAGN.*, vol. 1, no. 2, 1990.

[30] K. Kubota, I. Funaki, and Y. Okuno, "Comparison of Simulated Plasma Flow Field in a Two-Dimensional Magnetoplasmadynamic Thruster With Experimental Data," *Plasma Science, IEEE Transactions on*, vol. 37, no. 12, 2009.

[31] T. Tajima, "Computational plasma physics. With applications to fusion and astrophysics.," *Frontiers in Physics, Vol. 72,*, vol. 1, 1989.

[32] M. M. Woolfson and G. J. Pert, *An introduction to computer simulation.* Oxford University Press, 1999.

[33] R. C. Weast, M. J. Astle, and W. H. Beyer, *CRC handbook of chemistry and physics*, vol. 69. CRC press Boca Raton, FL, 1988.

[34] C. L. Yaws, *Inorganic compounds and elements*, vol. 4. Gulf Publ. Co., 1996.

[35] J. P. Goedbloed and S. Poedts, *Principles of magnetohydrodynamics: With applications to laboratory and astrophysical plasmas.* Cambridge Univ Pr, 2004.

[36] C. E. Moore, "Atomic Energy Levels. As Derived From the Analyses of Optical Spectra. Volume 3," tech. rep., DTIC Document, 1958.

[37] R. T. Downey, *Theoretical and Experimental Investigation into High Current Hollow Cathode Arc Attachment.* PhD thesis, University of Southern California, 2008.

[38] A. H. Stroud and D. Secrest, *Gaussian quadrature formulas*, vol. 39. Prentice-Hall Englewood Cliffs, NJ, 1966.

[39] W. Taylor, "Algorithms and FORTRAN programs to calculate classical collision integrals for realistic intermolecular potential," *Rt. MLM-2661.-Mound Res. Corp*, 1979.

[40] R. A. Aziz and H. H. Chen, "An accurate intermolecular potential for argon," *The Journal of Chemical Physics*, vol. 67, no. 12, 1977.

[41] S. I. Sandler, *An introduction to applied statistical thermodynamics*. Wiley, 2010.

[42] G. Herzberg, *Atomic spectra and atomic structure*. Dover publications, 2010.

[43] A. B. Cambel, "Plasma physics and magnetofluid mechanics," 1963.

[44] T. K. Bose, "Thermophysical and transport properties of multicomponent gas plasmas at multiple temperatures," *Progress in Aerospace Sciences*, vol. 25, no. 1, 1988.

[45] J. O. Hirschfelder, C. F. Curtiss, R. B. Bird, and U. of Wisconsin. Theoretical Chemistry Laboratory, *Molecular theory of gases and liquids*, vol. 26. Wiley New York, 1954.

[46] R. S. Cohen, L. Spitzer, and P. M. R. Routly, "The electrical conductivity of an ionized gas," *Physical Review*, vol. 80, no. 2, p. 230, 1950.

[47] J. D. Cobine, *Gaseous conductors: theory and engineering applications*. Dover Publications New York, 1958.

[48] L. Spitzer, "Physics of fully ionized gases," *Interscience Tracts on Physics and Astronomy, New York: Interscience Publication, 1965, 2nd rev. ed.*, vol. 1, 1965.

[49] M. Mitchner and C. H. Kruger, *Partially ionized gases*, vol. 8. Wiley New York, 1973.

[50] T. K. Bose, *High temperature gas dynamics*. Springer, 2004.

[51] J. D. Anderson, "Hypersonic and high temperature gas dynamics," 1989.

[52] C. E. Moore, *Atomic energy levels as derived from the analyses of optical spectra*, vol. 1,2,3. US Dept. of Commerce, National Bureau of Standards, 1948.

[53] L. K. Nash, *Elements of statistical thermodynamics*. Dover publications, 2006.

[54] J. A. Kunc and D. A. Erwin. ,Lecture Note in ASTE501a,b "Physical Gas Dynamics", University of Southern California, LA, CA, 2011 [Unpublished].

[55] M. H. Fontana, "Monat: A fortran 63 program for computing thermodynamic properties of monatomic ideal gases.," tech. rep., Oak Ridge National Lab., Tenn., 1968.

[56] D. A. Erwin ,University of Southern California, LA, CA, 1990 [Unpublished].

[57] D. Pullin, "Direct simulation methods for compressible inviscid ideal-gas flow," *Journal of Computational Physics*, vol. 34, no. 2, pp. 231–244, 1980.

[58] K. A. Hoffman and S. T. Chiang, "Computational Fluid Dynamics–Volume II," *Engineering Education System*, 2000.

[59] K. A. Hoffmann, "Computational fluid dynamics for engineers(Book)," *Austin, TX: Engineering Education System, 1989.*, 1989.

Appendix A

Derivatives in the Computational Domain for further reading [58]. Consider a function f , where it is required to determine its first-and second order derivatives in the computational domain.

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial x} + \eta_x \frac{\partial}{\partial \eta} \tag{A-1}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial y} + \eta_y \frac{\partial}{\partial \eta} \tag{A-2}$$

Therefore,

$$\frac{\partial f}{\partial x} = f_x = \xi_x f_\xi + \eta_x f_\eta \tag{A-3}$$

$$\frac{\partial f}{\partial y} = f_y = \xi_y f_\xi + \eta_y f_\eta \tag{A-4}$$

$$f_x = J y_\eta f_\xi + (-J y_\xi) f_\eta = J(y_\eta f_\xi - y_\xi f_\eta) \tag{A-5}$$

$$f_y = J x_\eta f_\xi + (-J x_\xi) f_\eta = J(x_\eta f_\xi - x_\xi f_\eta) \tag{A-6}$$

To determine the second-order derivatives, f_{xx} and f_{yy}

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial}{\partial x}\left(\xi_x f_\xi + \eta_x f_\eta\right) = \left(\xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta}\right)\left(\xi_x f_\xi + \eta_x f_\eta\right) \quad \text{(A-7)}$$

$$\frac{\partial^2 f}{\partial x^2} = \xi_x \frac{\partial}{\partial \xi}\left(\xi_x f_\xi + \eta_x f_\eta\right) + \eta_x \frac{\partial}{\partial \eta}\left(\xi_x f_\xi + \eta_x f_\eta\right)$$

$$= \xi_x^2 f_{\xi\xi} + \xi_x f_\xi \frac{\partial}{\partial \xi}(\xi_x) + \xi_x \eta_x f_{\xi\eta} + \xi_x f_\eta \frac{\partial}{\partial \xi}(\eta_x) + \eta_x \xi_x f_{\xi\eta} + \eta_x f_\eta \frac{\partial}{\partial \eta}(\xi_x) + \eta_x^2 f_{\eta\eta} + \eta_x f_\eta \frac{\partial}{\partial \eta}(\eta_x)$$

where

$$\xi_x = J y_\eta, \xi_y = -J x_\eta, \eta_x = -J y_\xi, \eta_y = J x_\xi$$

Let

$$\xi = \triangle$$

then, we have

$$\frac{\partial^2 f}{\partial x^2} = J^2(y_\eta^2 f_{\triangle\triangle} - 2 y_\triangle y_\eta f_{\triangle\eta} + y_\triangle^2 f_{\eta\eta}) + J y_\eta \left[f_\triangle \frac{\partial}{\partial \triangle}(\triangle_x) + f_\eta \frac{\partial}{\partial \triangle}(\eta_x) \right]$$

$$+(-J y_\triangle)\left[f_\triangle \frac{\partial}{\partial \eta}(\triangle x) + f_\eta \frac{\partial}{\partial \eta}(\eta_x) \right]$$

At this point, the derivatives of the metrics are determined as follows:

$$\frac{\partial}{\partial \triangle}(\triangle x) = \frac{\partial}{\partial \triangle}(J y_\eta) = \frac{\partial}{\partial \triangle}\left(\frac{y_\eta}{x_\triangle y_\eta - x_\eta y_\triangle}\right)$$

$$\frac{\partial}{\partial\triangle}(\triangle x) = J^2\Bigg[y_\triangle\eta(x_\triangle y_\eta - x_\eta y_\triangle) - y_\eta(y_\eta x_{\triangle\triangle} + x_\triangle y_\triangle\eta - x_\eta y_{\triangle\triangle} - y_\triangle x_{\triangle\eta}\Bigg]$$

or

$$\frac{\partial}{\partial\triangle}(\triangle x) = J^2(x_\triangle y_\eta y_{\triangle\eta} - x_\eta y_\triangle y_{\triangle\eta} - y_\eta^2 x_{\triangle\triangle} - x_\triangle y_\eta y_{\triangle\eta} + x_\eta y_\eta y_{\triangle\triangle} + y_\triangle y_\eta x_{\triangle\eta}) \tag{A-8}$$

$$\frac{\partial}{\partial\triangle}(\eta x) = -J^2(x_\triangle y_\eta y_{\triangle\triangle} - x_\eta y_\triangle y_{\triangle\triangle} - y_\triangle y_\eta x_{\triangle\triangle} - x_\triangle y_\triangle y_{\triangle\eta} + x_\eta y_\triangle y_{\triangle\triangle} + y_\triangle^2 x_{\triangle\eta}) \tag{A-9}$$

$$\frac{\partial}{\partial\eta}(\triangle x) = J^2(x_\triangle y_\eta y_{\eta\eta} - x_\eta y_\triangle y_{\eta\eta} - x_\triangle y_\eta y_{\eta\eta} - y_\eta^2 x_{\triangle\eta} + y_\triangle y_\eta x_{\eta\eta} + x_\eta y_\eta y_{\triangle\eta}) \tag{A-10}$$

$$\frac{\partial}{\partial\eta}(\eta x) = -J^2(x_\triangle y_\eta y_{\triangle\eta} - x_\eta y_\triangle y_{\triangle\eta} - y_\triangle x_\triangle y_{\eta\eta} - y_\triangle y_\eta x_{\triangle\eta} + y_\triangle^2 x_{\eta\eta} + x_\eta y_\triangle y_{\triangle\eta}) \tag{A-11}$$

Substituting

$$\frac{\partial^2 f}{\partial x^2} = J^2(y_\eta^2 f_{\eta\eta} - 2y_\triangle y_\eta f_{\triangle\eta} + y_\triangle^2 f_{\eta\eta} + \tag{A-12}$$

$$+J^3\Bigg[(y_\eta^2 y_{\triangle\triangle} - 2y_\eta y_\triangle y_{\triangle\eta} + y_\triangle^2 y_{\eta\eta})(x_\eta f_\triangle - x_\triangle f_\eta) +$$

$$+(y_\eta^2 x_{\triangle\triangle} - 2y_\eta y_\triangle x_{\triangle\eta} + y_\triangle^2 x_{\eta\eta})(y_\triangle f_\eta - y_\eta f_\triangle)\Bigg]$$

$$\frac{\partial^2 f}{\partial y^2} = J^2(x_\eta^2 f_{\triangle\triangle} - 2x_\triangle x_\eta f_{\triangle\eta} + x_\triangle^2 f_{\eta\eta}) + \tag{A-13}$$

$$+J^3\Bigg[(x_\eta^2 y_{\triangle\triangle} - 2x_\triangle x_\eta y_{\triangle\eta} + x_\triangle^2 y_{\eta\eta})(x_\eta f_\triangle - x_\triangle f_\eta) +$$

$$+(x_\eta^2 x_{\triangle\triangle} - 2x_\triangle x_\eta x_{\triangle\eta} + x_\triangle^2 x_{\eta\eta})(y_\triangle f_\eta - y_\eta f_\triangle)\Bigg]$$

Now, consider the Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{A-14}$$

$$\nabla^2 f = J^2(af_{\triangle\triangle} - 2bf_{\triangle\eta} + cf_{\eta\eta}) +$$

$$+J^3\Bigg[(ay_{\triangle\triangle} - 2by_{\triangle\eta} + cy_{\eta\eta})(x_\eta f_\triangle - x_\triangle f_\eta)$$

$$+(ax_{\triangle\triangle} - 2bx_{\triangle\eta} + cx_\eta\eta)(y_\triangle f_\eta - y_\eta f_\triangle)\Bigg]$$

where

$$x_\eta^2 + y_\eta^2 = a$$

$$x_\triangle x_\eta + y_\triangle y_\eta = b$$

$$x_\triangle^2 + y_\triangle^2 = c$$

and finally

$$\nabla^2 f = J^2(af_{\triangle\triangle} - 2bf_{\triangle\eta} + cf_{\eta\eta} + df_\eta + ef_\triangle) \tag{A-15}$$

where

$$d = J(y_{\triangle}\alpha - x_{\triangle}\beta)$$

$$e = J(x_{\triangle}\beta - y_{\eta}\alpha)$$

$$\alpha = ax_{\triangle\triangle} - 2bx_{\triangle\eta} + cx_{\eta\eta}$$

$$\beta = ay_{\triangle\triangle} - 2by_{\triangle\eta} + cy_{\eta\eta}$$

For elliptic system

$$\nabla^2\xi = 0$$

$$\nabla^2\eta = 0$$

let

$$f = \xi$$

$$\xi_\xi = \frac{\partial\xi}{\partial\xi} = 1$$

$$\xi_\eta = 0$$

$$\xi_{\xi\xi} = \frac{\partial}{\partial\xi}\left(\frac{\partial\xi}{\partial\xi}\right) = 0$$

$$\xi_{\eta\eta} = 0$$

$$\xi_{\xi\eta} = 0$$

A-15 yield :

$$\nabla^2\xi = 0, \quad \nabla^2\eta = 0$$

$$J^2 e = 0, \quad J^2 d = 0$$

$$J^3(x_\eta\beta - y_\eta\alpha) = 0, \quad J^3(y_\xi\alpha - x_\xi\beta) = 0$$

Since

$$J \neq 0$$

then

$$x_\eta \beta - y_\eta \alpha = 0$$

$$y_\eta \alpha - x_\xi \beta = 0$$

Eliminating α yield

$$\beta(x_\xi y_\eta - x_\eta y_\xi) = 0$$

But

$$x_\xi y_\eta - x_\eta y_\xi = \frac{1}{J}$$

Thus,

$$\frac{1}{J}\beta = 0$$

Since $J \neq 0$, then

$$\beta = 0$$

or

$$a y_{\xi\xi} - 2b y_{\xi\eta} + c y_{\eta\eta} = 0$$

We show that $\beta = 0$ and , therefore, α must also be zero, which result in

$$a x_{\xi\xi} - 2b x_{\xi\eta} + c x_{\eta\eta} = 0$$

Appendix B

Transformation of the Governing Parial differential Equations [59]

Now, define the following relation between the physical and computational spaces

$$\xi = \xi(x, y) \tag{B-16}$$

$$\eta = \eta(x, y) \tag{B-17}$$

The chain rule for partial differentiation

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} \tag{B-18}$$

let

$$\frac{\partial \xi}{\partial x} = \xi_x$$

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} \tag{B-19}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} \tag{B-20}$$

Now consider a model PDE such as

$$\frac{\partial U}{\partial x} + a\frac{\partial U}{\partial y} = 0 \tag{B-21}$$

Transforming from physical spcae to computational space

$$\xi_x \frac{\partial U}{\partial \xi} + \eta_x \frac{\partial U}{\partial \eta} + a\left(\xi_y \frac{\partial U}{\partial \xi} + \eta_y \frac{\partial U}{\partial \eta}\right) = 0 \tag{B-22}$$

which may be rearranged as

$$\left(\xi_x + a\xi_y\right)\frac{\partial U}{\partial \xi} + \left(\eta_x + a\eta_y\right)\frac{\partial U}{\partial \eta} = 0 \tag{B-23}$$

Matrics and the Jacobian of Transformation

From C-19 and C-20

$$\xi_x = \frac{\partial \xi}{\partial x} \cong \frac{\triangle \xi}{\triangle x}$$

This ratio of arc length in the computational space to the physical space

From C-16 and C-17

$$d\xi = \xi_x dx + \xi_y dy$$

$$d\eta = \eta_x dx + \eta_y dy$$

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}$$

Reverse the role of independent variables, we get

$$dx = x_\xi d_\xi + x_\eta d_\eta$$

$$dy = y_\xi d_\xi + y_\eta d_\eta$$

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}$$

it can be in the form that

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}^{-1}$$

from which

$$\xi_x = J y_\eta \tag{B-24}$$

$$\xi_y = -J x_\eta \tag{B-25}$$

$$\eta_x = -J y_\xi \tag{B-26}$$

$$\eta_y = J x_\xi \tag{B-27}$$

The Jacobian Transfromation is

$$J = \frac{1}{x_\xi y_\eta - y_\xi x_\eta} \tag{B-28}$$

Appendix C

```
C This mainprogram 1DMPD thruster includes the plasma sheath model
C By given the cathode temperature and the current density:
C the plasma sheath model provides the heat flux and the sheath voltage.
C The mainprogram will stop when it reaches the steady state.
C Written by: Thada Suksila
C Date: Sep 27, 2014
C UPDATE: Oct 20, 2014
C
      PROGRAM mainprogram
      INCLUDE 'common_declare.f'
      OPEN(93, FILE = 'mainprogram.txt', STATUS = 'UNKNOWN')
      OPEN(94, FILE = 'mainprogram_C.txt',STATUS= 'UNKNOWN')
      OPEN(95, FILE = 'mainprogram_P.txt',STATUS= 'UNKNOWN')
C
      CALL grid
      CALL tinit
      CALL vinit

      NSTEPS = 7000
      TIME   = 0.0
      TOL    = 5E-3
C check critical time 1.Von Neumann at cathode and 2. Ohmic Heating at plasma
C Cathode
      T = TAVG_C(NPC)
      CALL KTH_C(T,XKTH_C)
      T = TC
      CALL Cp_C(T,CAP_C)
      DT_VON_C = CAP_C*XLSC(NNPC)*XNLC/(2*XKTH_C)
      DT_OHM_C = CAP_C*XLSC(NNPC)/XJ
C      WRITE(93,*) DT_VON_C, DT_OHM_C
C Plasma
      T = TAVG_P(1)
      CALL KTH_P(T,XKTH_P)
```

```
       T = TE
       CALL Cp_P(T,CP_NOR)      ! CP_NOR = Z*CP/R
       CALL Z(T, XZ)
       Rho   = PR/(Rar*T)
       CAP_P = CP_NOR/XZ*Rar*1000
       CAP_P = CAP_P*Rho
       XCAP_P(1) = CAP_P
       DT_VON_P = CAP_P*XLSP(1)*XNLC/(2*XKTH_P)
       DT_OHM_P = CAP_P*XLSP(1)/XJ
C       WRITE(93,*) DT_VON_P, DT_OHM_P
C choose DT cathode
       IF(DT_VON_C.GT.DT_OHM_C)THEN
          DTIME_C = DT_OHM_C
       ELSE
          DTIME_C = DT_VON_C
       ENDIF
C choose DT plasma
       IF(DT_VON_P.GT.DT_OHM_P)THEN
          DTIME_P = DT_OHM_P
       ELSE
          DTIME_P = DT_VON_P
       ENDIF
C choose between cathode and plasma
       IF(DTIME_C.GT.DTIME_P)THEN
          DTIME = DTIME_P
       ELSE
          DTIME = DTIME_C
       ENDIF
       DTIME = DTIME/5
C Fixed temperature at the cathode base and anode
       TEMP_C(1)    = TCB
       TEMP_P(NNPP) = TEA
C Solve for temperature and potential
       DO 500 ISTEP = 1, NSTEPS
          DTT = ISTEP*DTIME
C #1 conductivities at cathode region
          DO 1 I = 1, NPC
             T = TAVG_C(I)
             CALL KTH_C(T,XKTH_C)
             CALL SIGMA_C(T,SIG_C)
             XKAPA_C(I) = XKTH_C
             XSIGMA_C(I)= SIG_C
 1        CONTINUE
C #2 conductivities at plasma region
          DO 2 I = 1, NPP
             T = TAVG_P(I)
             CALL KTH_P(T,XKTH_P)
             CALL SIGMA_P(T,SIG_P)
             XKAPA_P(I) = XKTH_P
             XSIGMA_P(I)= SIG_P
 2        CONTINUE
C Solve for Potential (V)
C #3 plasma region
          DO 3 I = NNPP-1, 1, -1
             XPHINEW_P(I) = XPHI_P(I+1) - ((XJ*XNLP)/(XSIGMA_P(I)))
             XPHI_P(I)    = XPHINEW_P(I)
 3        CONTINUE
C #4 Call for the plasma sheath model
          T = TEMP_C(NNPC)                    !****************
          J =  XJ
          CALL CMODEL( T, J, VC, Q )
          Q = Q
          WRITE(93,*)  TIME, T,  J,  Vc,  Q
C #5 Include voltage drop (plasma sheath, VC)
          XPHI_C(NNPC) = XPHI_P(1) - VC
C #6 cathode region
          DO 4 I = NNPC-1, 1, -1
             XPHINEW_C(I) = XPHI_C(I+1) - ((XJ*XNLC)/(XSIGMA_C(I)))
             XPHI_C(I)    = XPHINEW_C(I)
 4        CONTINUE
C Solve for Temperature (K)
C #7 plasma region
C #7.1 at the NC+1
          T = TEMP_P(1)
          CALL Cp_P(T,CP_NOR) ! CP_NOR = Z*CP/R
          CALL Z(T, XZ)
          Rho   = PR/(Rar*T)
          CAP_P = CP_NOR/XZ*Rar*1000
          CAP_P = CAP_P*Rho
          XCAP_P(1) = CAP_P
          ALPHA = DTT/(XCAP_P(1)*XLSP(1))
          BETA1 = ALPHA*XJ/2
          BETA2 = ALPHA*XKAPA_P(1)/XNLP
          A     = BETA1*(XPHI_P(2)-XPHI_P(1))
          B     = ALPHA*Q
          C     = BETA2*(TEMP_P(2)-TEMP_P(1))
          TNEW_P(1) = TEMP_P(1) + A - B + C  ! Eq.4.21
          TEMP_P(1) = TNEW_P(1)
          TAVG_P(1) = (TEMP_P(1)+TEMP_P(2))/2
C #7.2 the rest except the anode
          DO 5 I = 2, NNPP-1
             T = TEMP_P(I)
             CALL Cp_P(T,CP_NOR) ! CP_NOR = Z*CP/R
             CALL Z(T, XZ)
             Rho   = PR/(Rar*T)
             CAP_P = CP_NOR/XZ*Rar*1000
             CAP_P = CAP_P*Rho
             XCAP_P(I) = CAP_P
             ALPHA = DTT/(XCAP_P(I)*XLSP(I))
             BETA1 = ALPHA*XJ/2
             BETA2 = ALPHA*XKAPA_P(I)/XNLP
             BETA3 = ALPHA*XKAPA_P(I-1)/XNLP
             A     = BETA1*(XPHI_P(I+1)-XPHI_P(I-1))
             B     = BETA2*(TEMP_P(I+1)-TEMP_P(I))
             C     = BETA3*(TEMP_P(I)  -TEMP_P(I-1))
             TNEW_P(I)    = TEMP_P(I) + A + B - C   ! Eq.4.19
             TEMP_P(I)    = TNEW_P(I)
             TAVG_P(I)    = (TEMP_P(I)+TEMP_P(I+1))/2
 5        CONTINUE
C #8 cathode region
C #8.1 at the cathode tip
          T = TEMP_C(NNPC)
          CALL Cp_C(T,CAP_C)
          XCAP_C(NNPC) = CAP_C
          ALPHA = DTT/(XCAP_C(NNPC)*XLSC(NNPC))
          BETA1 = ALPHA*XJ/2
          BETA2 = ALPHA*XKAPA_C(NNPC)/XNLC
          A = BETA1*(XPHI_C(NNPC)-XPHI_C(NNPC-1))
          B = ALPHA*Q
          C = BETA2*(TEMP_C(NNPC) -TEMP_C(NNPC-1))
          TNEW_C(NNPC) = TEMP_C(NNPC) + A + B - C  !Eq. 4.20
          TEMP_C(NNPC) = TNEW_C(NNPC)
          TAVG_C(NNPC) = (TEMP_C(NNPC)+TEMP_C(NNPC-1))/2
C #8.2 the rest except cathode base
          DO 6 I = 2, NNPC-1
             T = TEMP_C(I)
             CALL Cp_C(T,CAP_C)
             XCAP_C(I) = CAP_C
             ALPHA = DTT/(XCAP_C(I)*XLSC(I))
             BETA1 = ALPHA*XJ/2
             BETA2 = ALPHA*XKAPA_C(I)/XNLC
             BETA3 = ALPHA*XKAPA_C(I-1)/XNLC
             A     = BETA1*(XPHI_C(I+1)-XPHI_C(I-1))
             B     = BETA2*(TEMP_C(I+1)-TEMP_C(I))
             C     = BETA3*(TEMP_C(I)  -TEMP_C(I-1))
             TNEW_C(I)    = TEMP_C(I) + A + B - C  !Eq.4.19
             TEMP_C(I)    = TNEW_C(I)
             TAVG_C(I)    = (TEMP_C(I)+TEMP_C(I+1))/2
 6        CONTINUE
C print output
          IP = ISTEP - (ISTEP/1)*1
          IF(IP.EQ.0)THEN
             WRITE(94,300) TIME, (TEMP_C(I),I = 1, NNPC),(XPHI_C(I),I = 1,NNPC),Q,VC
             WRITE(95,300) TIME, (TEMP_P(I),I = 1, NNPP),(XPHI_P(I),I = 1,NNPP),Q,VC
          ENDIF
 300      FORMAT(F15.6,2X,11(F15.4,2X),11(F15.4,2X),2X,F15.4,2X,F15.4)
          TIME = TIME + DTIME


 500   CONTINUE
 501   STOP
       END PROGRAM

C ---------------------- SUBROUTINE HEAT CAPACITY ---------------------
C  subroutine to calculate the temperature dependent Heat capacity
C  of the Cathode and Anode
C ---------------------------------------------------------------------
      SUBROUTINE Cp_C( T, CAP_C )
      INCLUDE 'common_declare.f'
C     Compute heat capacity of tungsten in J/(K m^3)

      RHOMOL_C = RHO_C / XMAMU_C      ! gmol/m^3

      AVOLJ = AMOLCAL_C * XJPERCAL * RHOMOL_C
      BVOLJ = BMOLCAL_C * XJPERCAL * RHOMOL_C

      CAP_C = AVOLJ + 1E-3*T*BVOLJ
C Convert from J/(K m^3) to J/(Kg K) by 19.25E3 (density at room temperature (Wiki)
C      CAP_C = CAP_C / (RHO_C*1E-3)

      RETURN
      END

C ---------------- SUBROUTINE CTCOND ---------------------------
C  subroutine to calculate the temperature dependent thermal
C  conductivity of Cathode(Tungsten)
C  TUNGSTEN VALUES solid and liquid  [W/m/K]
C --------------------------------------------------------------
      SUBROUTINE KTH_C(T, XKTH_C)
      INCLUDE 'common_declare.f'
      IF ( T .LE. 3670 ) THEN
         XKTH_C = A0 + A1*EXP(-A2*T) + A3*EXP(-A4*T)
      ELSE
         XKTH_C = B0 + B1*T + B3*T*T
      ENDIF

      RETURN
      END
```

```
C ------------------------ SUBROUTINE SIGMA_C -------------------------
C This subroutine calculates the temperature dependent Electrical
C conductivity of cathode(Tungsten) [mho/m]
C A linear curve fit of the electrical resistivity in page 99 eq.3.10 Goodfellow
C --------------------------------------------------------------------
      SUBROUTINE SIGMA_C(T,SIG_C)
      INCLUDE 'common_declare.f'

      XRHOE_C  = (C0 + C1*T)*0.00000001
      SIG_C = 1/XRHOE_C

      RETURN
      END
C------------------------- SUBROUTINE CP Normalized --------------------
C This subroutine calculates the Cp of argon without any unit
C Written By: Thada Suksila
C----------------------------------------------------------------------
      SUBROUTINE Cp_P(T,CP_NOR)
      DIMENSION X(200), FX(200,200)
      OPEN(11,FILE= 'P0.0006_5000-5500.txt', STATUS = 'OLD')
      OPEN(12,FILE= 'P0.0006_5500-5750.txt', STATUS = 'OLD')
      OPEN(13,FILE= 'P0.0006_5750-6000.txt', STATUS = 'OLD')
      OPEN(14,FILE= 'P0.0006_6000-6500.txt',STATUS = 'OLD')
      OPEN(15,FILE= 'P0.0006_6500-7000.txt',STATUS= 'OLD')
      OPEN(16,FILE= 'P0.0006_7000-7250.txt',STATUS= 'OLD')
      OPEN(17,FILE= 'P0.0006_7250-7400.txt',STATUS= 'OLD')
      OPEN(18,FILE= 'P0.0006_7400-7600.txt',STATUS= 'OLD')
      OPEN(19,FILE= 'P0.0006_7600-7800.txt',STATUS= 'OLD')
      OPEN(20,FILE= 'P0.0006_7800-8000.txt',STATUS= 'OLD')
      OPEN(21,FILE= 'P0.0006_8000-8100.txt',STATUS= 'OLD')
      OPEN(22,FILE= 'P0.0006_8100-8200.txt',STATUS= 'OLD')
      OPEN(23,FILE= 'P0.0006_8200-8300.txt',STATUS= 'OLD')
      OPEN(24,FILE= 'P0.0006_8300-8600.txt',STATUS= 'OLD')
      OPEN(25,FILE= 'P0.0006_8600-8800.txt',STATUS= 'OLD')
      OPEN(26,FILE= 'P0.0006_8800-8900.txt',STATUS= 'OLD')
      OPEN(27,FILE= 'P0.0006_8900-9000.txt',STATUS= 'OLD')
      OPEN(28,FILE= 'P0.0006_9000-9100.txt',STATUS= 'OLD')
      OPEN(29,FILE= 'P0.0006_9100-9200.txt',STATUS= 'OLD')
      OPEN(30,FILE= 'P0.0006_9200-9300.txt',STATUS= 'OLD')
      OPEN(31,FILE= 'P0.0006_9300-9400.txt',STATUS= 'OLD')
      OPEN(32,FILE= 'P0.0006_9400-9500.txt',STATUS= 'OLD')
      OPEN(33,FILE= 'P0.0006_9500-9600.txt',STATUS= 'OLD')
      OPEN(34,FILE= 'P0.0006_9600-9700.txt',STATUS= 'OLD')
      OPEN(35,FILE= 'P0.0006_9700-9800.txt',STATUS= 'OLD')
      OPEN(36,FILE= 'P0.0006_9800-9900.txt',STATUS= 'OLD')
      OPEN(37,FILE= 'P0.0006_9900-10000.txt',STATUS= 'OLD')
      OPEN(38,FILE= 'P0.0006_10000-10100.txt',STATUS= 'OLD')
      OPEN(39,FILE= ’P0.0006_10100-10200.txt’,STATUS= ’OLD’)
      OPEN(40,FILE= ’P0.0006_10200-10300.txt’,STATUS= ’OLD’)
      OPEN(41,FILE= ’P0.0006_10300-10400.txt’,STATUS= ’OLD’)
      OPEN(42,FILE= ’P0.0006_10400-10500.txt’,STATUS= ’OLD’)
      OPEN(43,FILE= ’P0.0006_10500-10600.txt’,STATUS= ’OLD’)
      OPEN(44,FILE= ’P0.0006_10600-10700.txt’,STATUS= ’OLD’)
      OPEN(45,FILE= ’P0.0006_10700-10800.txt’,STATUS= ’OLD’)
      OPEN(46,FILE= ’P0.0006_10800-10900.txt’,STATUS= ’OLD’)
      OPEN(47,FILE= ’P0.0006_10900-11000.txt’,STATUS= ’OLD’)
      OPEN(48,FILE= ’P0.0006_11000-11200.txt’,STATUS= ’OLD’)
      OPEN(49,FILE= ’P0.0006_11200-11400.txt’,STATUS= ’OLD’)
      OPEN(50,FILE= ’P0.0006_11400-11600.txt’,STATUS= ’OLD’)
      OPEN(51,FILE= ’P0.0006_11600-11800.txt’,STATUS= ’OLD’)
      OPEN(52,FILE= ’P0.0006_11800-12000.txt’,STATUS= ’OLD’)
      OPEN(53,FILE= ’P0.0006_12000-12300.txt’,STATUS= ’OLD’)
      OPEN(54,FILE= ’P0.0006_12300-12600.txt’,STATUS= ’OLD’)
      OPEN(55,FILE= ’P0.0006_12600-13000.txt’,STATUS= ’OLD’)
C given the temperature
      XX  = T
      IF(XX.LE.5000)THEN
         FF = 2.378552973
         GOTO 101
      ELSEIF(XX.GT.13000)THEN
         FF = 0.00017581*XX + 3.69888312
         GOTO 101
      ENDIF
      IF(XX.GT.5000.AND.XX.LE.5500) THEN
         N = 7
         DO IROW = 1,N
            READ(11,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.5500.AND.XX.LE.5750)THEN
         N = 10
         DO IROW = 1,N
            READ(12,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.5750.AND.XX.LE.6000)THEN
         N = 8
         DO IROW = 1,N
            READ(13,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.6000.AND.XX.LE.6500)THEN
         N = 12
         DO IROW = 1,N
            READ(14,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.6500.AND.XX.LE.7000)THEN
```

```
   N = 12
   DO IROW = 1,N
      READ(15,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7000.AND.XX.LE.7250)THEN
   N = 7
   DO IROW = 1,N
      READ(16,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7250.AND.XX.LE.7400)THEN
   N = 8
   DO IROW = 1,N
      READ(17,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7400.AND.XX.LE.7600)THEN
   N = 8
   DO IROW = 1,N
      READ(18,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7600.AND.XX.LE.7800)THEN
   N = 8
   DO IROW = 1,N
      READ(19,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7800.AND.XX.LE.8000)THEN
   N = 8
   DO IROW = 1,N
      READ(20,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8000.AND.XX.LE.8100)THEN
   N = 8
   DO IROW = 1,N
      READ(21,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8100.AND.XX.LE.8200)THEN
   N = 8
   DO IROW = 1,N
      READ(22,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8200.AND.XX.LE.8300)THEN
   N = 8
   DO IROW = 1,N
      READ(23,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8300.AND.XX.LE.8600)THEN
   N = 9
   DO IROW = 1,N
```

```
      READ(24,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8600.AND.XX.LE.8800)THEN
   N = 8
   DO IROW = 1,N
      READ(25,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8800.AND.XX.LE.8900)THEN
   N = 7
   DO IROW = 1,N
      READ(26,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8900.AND.XX.LE.9000)THEN
   N = 8
   DO IROW = 1,N
      READ(27,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9000.AND.XX.LE.9100)THEN
   N = 8
   DO IROW = 1,N
      READ(28,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9100.AND.XX.LE.9200)THEN
   N = 8
   DO IROW = 1,N
      READ(29,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9200.AND.XX.LE.9300)THEN
   N = 8
   DO IROW = 1,N
      READ(30,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9300.AND.XX.LE.9400)THEN
   N = 8
   DO IROW = 1,N
      READ(31,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9400.AND.XX.LE.9500)THEN
   N = 8
   DO IROW = 1,N
      READ(32,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9500.AND.XX.LE.9600)THEN
   N = 8
   DO IROW = 1,N
      READ(33,*) X(IROW),FX(IROW,1)
   ENDDO
```

```
ELSEIF(XX.GT.9600.AND.XX.LE.9700)THEN
   N = 8
   DO IROW = 1,N
      READ(34,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9700.AND.XX.LE.9800)THEN
   N = 8
   DO IROW = 1,N
      READ(35,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9800.AND.XX.LE.9900)THEN
   N = 8
   DO IROW = 1,N
      READ(36,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9900.AND.XX.LE.10000)THEN
   N = 8
   DO IROW = 1,N
      READ(37,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10000.AND.XX.LE.10100)THEN
   N = 8
   DO IROW = 1,N
      READ(38,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10100.AND.XX.LE.10200)THEN
   N = 8
   DO IROW = 1,N
      READ(39,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10200.AND.XX.LE.10300)THEN
   N = 8
   DO IROW = 1,N
      READ(40,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10300.AND.XX.LE.10400)THEN
   N = 8
   DO IROW = 1,N
      READ(41,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10400.AND.XX.LE.10500)THEN
   N = 8
   DO IROW = 1,N
      READ(42,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10500.AND.XX.LE.10600)THEN
   N = 8
   DO IROW = 1,N
      READ(43,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10600.AND.XX.LE.10700)THEN
   N = 8
   DO IROW = 1,N
      READ(44,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10700.AND.XX.LE.10800)THEN
   N = 8
   DO IROW = 1,N
      READ(45,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10800.AND.XX.LE.10900)THEN
   N = 8
   DO IROW = 1,N
      READ(46,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10900.AND.XX.LE.11000)THEN
   N = 8
   DO IROW = 1,N
      READ(47,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11000.AND.XX.LE.11200)THEN
   N = 9
   DO IROW = 1,N
      READ(48,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11200.AND.XX.LE.11400)THEN
   N = 8
   DO IROW = 1,N
      READ(49,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11400.AND.XX.LE.11600)THEN
   N = 7
   DO IROW = 1,N
      READ(50,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11600.AND.XX.LE.11800)THEN
   N = 8
   DO IROW = 1,N
      READ(51,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11800.AND.XX.LE.12000)THEN
   N = 7
   DO IROW = 1,N
      READ(52,*) X(IROW),FX(IROW,1)
```

```
         ENDDO
      ELSEIF(XX.GT.12000.AND.XX.LE.12300)THEN
         N = 8
         DO IROW = 1,N
            READ(53,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12300.AND.XX.LE.12600)THEN
         N = 9
         DO IROW = 1,N
            READ(54,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12600.AND.XX.LE.13000)THEN
         N = 10
         DO IROW = 1,N
            READ(55,*) X(IROW),FX(IROW,1)
         ENDDO
      ENDIF
C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
         CP_NOR = FF
 40   CONTINUE
 101  WRITE(89,100) XX, FF
 100  FORMAT(’value of F(X) at x = ’, E10.4,’is’, E16.7)
C Close all open files
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      CLOSE(26)
      CLOSE(27)
      CLOSE(28)
      CLOSE(29)
      CLOSE(30)
      CLOSE(31)
      CLOSE(32)
      CLOSE(33)
      CLOSE(34)
      CLOSE(35)
      CLOSE(36)
      CLOSE(37)
      CLOSE(38)
      CLOSE(39)
      CLOSE(40)
      CLOSE(41)
      CLOSE(42)
      CLOSE(43)
      CLOSE(44)
      CLOSE(45)
      CLOSE(46)
      CLOSE(47)
      CLOSE(48)
      CLOSE(49)
      CLOSE(50)
      CLOSE(51)
      CLOSE(52)
      CLOSE(53)
      CLOSE(54)
      CLOSE(55)
      RETURN
      END
C ------------------ SUBROUTINE Z (compressibility factor ----------
C This pressure is 0.0006atm to calculate the compressibility factor of gas
C Program for computing F(X) at a given X
C using Newton’s divided-difference interpolating polynomials
C Written By: Thada Suksila
C Date: October 27, 2013
C Calculate for Z at pressure 0.0006 atm
      SUBROUTINE Z(XI,FF)
      DIMENSION X(200), FX(200,200)
      OPEN(11,FILE= 'Z0.0006_6000-6500.txt', STATUS = 'OLD')
      OPEN(12,FILE= 'Z0.0006_6500-7000.txt', STATUS = 'OLD')
      OPEN(13,FILE= 'Z0.0006_7000-7500.txt', STATUS = 'OLD')
      OPEN(14,FILE= 'Z0.0006_7500-8000.txt',STATUS = 'OLD')
      OPEN(15,FILE= 'Z0.0006_8000-8500.txt',STATUS= 'OLD')
      OPEN(16,FILE= 'Z0.0006_8500-9000.txt',STATUS= 'OLD')
      OPEN(17,FILE= 'Z0.0006_9000-9500.txt',STATUS= 'OLD')
      OPEN(18,FILE= 'Z0.0006_9500-10000.txt',STATUS= 'OLD')
      OPEN(19,FILE= 'Z0.0006_10000-10500.txt',STATUS= 'OLD')
      OPEN(20,FILE= 'Z0.0006_10500-11000.txt',STATUS= 'OLD')
      OPEN(21,FILE= 'Z0.0006_11000-11500.txt',STATUS= 'OLD')
      OPEN(22,FILE= 'Z0.0006_11500-12000.txt',STATUS= 'OLD')
      OPEN(23,FILE= 'Z0.0006_12000-12500.txt',STATUS= 'OLD')
      OPEN(24,FILE= 'Z0.0006_12500-13000.txt',STATUS= 'OLD')
      OPEN(25,FILE= 'Z0.0006_13000-13500.txt',STATUS= 'OLD')
      OPEN(26,FILE= 'Z0.0006_13500-14000.txt',STATUS= 'OLD')
C Start to calculate using given temperature
      XX  = XI
      IF(XX.LE.6000)THEN
         FF = 1.0
         GOTO 101
      ELSEIF(XX.GT.14000)THEN
         FF = 2.0
         GOTO 101
      ENDIF
      IF(XX.GT.6000.AND.XX.LE.6500) THEN
         N = 5
         DO IROW = 1,N
            READ(11,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.6500.AND.XX.LE.7000)THEN
         N = 7
         DO IROW = 1,N
            READ(12,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.7000.AND.XX.LE.7500)THEN
         N = 7
         DO IROW = 1,N
            READ(13,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.7500.AND.XX.LE.8000)THEN
         N = 6
         DO IROW = 1,N
            READ(14,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.8000.AND.XX.LE.8500)THEN
```

```
   N = 7
   DO IROW = 1,N
      READ(15,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8500.AND.XX.LE.9000)THEN
   N = 7
   DO IROW = 1,N
      READ(16,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9000.AND.XX.LE.9500)THEN
   N = 7
   DO IROW = 1,N
      READ(17,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9500.AND.XX.LE.10000)THEN
   N = 6
   DO IROW = 1,N
      READ(18,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10000.AND.XX.LE.10500)THEN
   N = 7
   DO IROW = 1,N
      READ(19,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10500.AND.XX.LE.11000)THEN
   N = 6
   DO IROW = 1,N
      READ(20,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11000.AND.XX.LE.11500)THEN
   N = 6
   DO IROW = 1,N
      READ(21,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11500.AND.XX.LE.12000)THEN
   N = 6
   DO IROW = 1,N
      READ(22,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.12000.AND.XX.LE.12500)THEN
   N = 6
   DO IROW = 1,N
      READ(23,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.12500.AND.XX.LE.13000)THEN
   N = 6
   DO IROW = 1,N
```

```
            READ(24,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.13000.AND.XX.LE.13500)THEN
         N = 6
         DO IROW = 1,N
            READ(25,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.13500.AND.XX.LE.14000)THEN
         N = 4
         DO IROW = 1,N
            READ(26,*) X(IROW),FX(IROW,1)
         ENDDO
      ENDIF
C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
 40   CONTINUE
 101  WRITE(89,100) XX, FF
 100  FORMAT('value of F(X) at x = ', E10.4,'is', E16.7)
C Close all open files
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      CLOSE(26)
      RETURN
      END
C--------------------------------------------------------------------------
      SUBROUTINE KTH_P(T,XKTH_P)
C The pressure is 0.00065atm for thermal conductivity of argon
C Program for computing F(X) at a given X
C Written By: Thada Suksila
C using Newton's divided-difference interpolating polynomials
      DOUBLE PRECISION X, FX
      DIMENSION X(506), FX(506,506)
      OPEN(12,FILE= 'ThP0.0006atm7-8.txt', STATUS = 'OLD')
      OPEN(13,FILE= 'ThP0.0006atm8-9.txt', STATUS = 'OLD')
      OPEN(14,FILE= 'ThP0.0006atm9-10.txt',STATUS = 'OLD')
      OPEN(15,FILE= 'ThP0.0006atm10-11.txt',STATUS= 'OLD')
      OPEN(16,FILE= 'ThP0.0006atm11-12.txt',STATUS= 'OLD')
      OPEN(17,FILE= 'ThP0.0006atm12-13.txt',STATUS= 'OLD')
      OPEN(18,FILE= 'ThP0.0006atm13-15.txt',STATUS= 'OLD')
      OPEN(19,FILE= 'ThP0.0006atm15-17.txt',STATUS= 'OLD')
      OPEN(20,FILE= 'ThP0.0006atm17-19.txt',STATUS= 'OLD')
      OPEN(21,FILE= 'ThP0.0006atm19-21.txt',STATUS= 'OLD')
      OPEN(22,FILE= 'ThP0.0006atm21-23.txt',STATUS= 'OLD')
      OPEN(23,FILE= 'ThP0.0006atm23-25.txt',STATUS= 'OLD')
      OPEN(24,FILE= 'ThP0.0006atm25-27.txt',STATUS= 'OLD')
      OPEN(25,FILE= 'ThP0.0006atm27-28.txt',STATUS= 'OLD')
C given the temperature
      XX = T
      IF(XX.LT.7000) THEN
         FF = 0.0171*1.5951
         GOTO 101
      ELSEIF(XX.GE.7000.AND.XX.LE.8000)THEN
         N = 14
         DO IROW = 1,N
            READ(12,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.8000.AND.XX.LE.9000)THEN
         N = 14
         DO IROW = 1,N
            READ(13,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.9000.AND.XX.LE.10000)THEN
         N = 14
         DO IROW = 1,N
            READ(14,*) X(IROW),FX(IROW,1)
         ENDDO
```

```
ELSEIF(XX.GT.10000.AND.XX.LE.11000)THEN
   N = 14
   DO IROW = 1,N
      READ(15,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11000.AND.XX.LE.12000)THEN
   N = 14
   DO IROW = 1,N
      READ(16,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.12000.AND.XX.LE.13000)THEN
   N = 14
   DO IROW = 1,N
      READ(17,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.13000.AND.XX.LE.15000)THEN
   N = 14
   DO IROW = 1,N
      READ(18,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.15000.AND.XX.LE.17000)THEN
   N = 14
   DO IROW = 1,N
      READ(19,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.17000.AND.XX.LE.19000)THEN
   N = 14
   DO IROW = 1,N
      READ(20,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.19000.AND.XX.LE.21000)THEN
   N = 14
   DO IROW = 1,N
      READ(21,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.21000.AND.XX.LE.23000)THEN
   N = 14
   DO IROW = 1,N
      READ(22,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.23000.AND.XX.LE.25000)THEN
   N = 14
   DO IROW = 1,N
      READ(23,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.25000.AND.XX.LE.27000)THEN
   N = 14
```

```
         DO IROW = 1,N
            READ(24,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.27000.AND.XX.LE.28000)THEN
         N = 14
         DO IROW = 1,N
            READ(25,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.28000)THEN
         FF = 225143.0861
         GOTO 101
      ENDIF
C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
C      WRITE(5,*)'Input the temperature (K):'
C      READ(5,*) XX
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
         TH  = FF
 40   CONTINUE
 101  WRITE(2,100) XX, FF,FF*0.00001
C Convert from erg/(K cm s) to Watt/(m K) by time 1E-5
      XKTH_P = FF*0.00001

 100  FORMAT('Temp =',E10.4,'is', E16.7,'or',E16.7)
 200  FORMAT(1X,E21.16,4X,E21.16)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      RETURN
      END
C---------------------------------------------------------------
      SUBROUTINE SIGMA_P(T,SIG_P)
C This pressure is 0.00065atm Electrical Conductivity
C Program for computing F(X) at a given X
C Written By: Thada Suksila
C using Newton’s divided-difference interpolating polynomials
      DOUBLE PRECISION X, FX
      DIMENSION X(506), FX(506,506)
      OPEN(10,FILE= ’ElP0.00065atm5.5-6.txt’,STATUS= ’OLD’)
      OPEN(11,FILE= ’ElP0.00065atm6-7.txt’, STATUS = ’OLD’)
      OPEN(12,FILE= ’ElP0.00065atm7-8.txt’, STATUS = ’OLD’)
      OPEN(13,FILE= ’ElP0.00065atm8-9.txt’, STATUS = ’OLD’)
      OPEN(14,FILE= ’ElP0.00065atm9-10.txt’,STATUS = ’OLD’)
      OPEN(15,FILE= ’ElP0.00065atm10-11.txt’,STATUS= ’OLD’)
      OPEN(16,FILE= ’ElP0.00065atm11-12.txt’,STATUS= ’OLD’)
      OPEN(17,FILE= ’ElP0.00065atm12-13.txt’,STATUS= ’OLD’)
      OPEN(18,FILE= ’ElP0.00065atm13-15.txt’,STATUS= ’OLD’)
      OPEN(19,FILE= ’ElP0.00065atm15-17.txt’,STATUS= ’OLD’)
      OPEN(20,FILE= ’ElP0.00065atm17-19.txt’,STATUS= ’OLD’)
      OPEN(21,FILE= ’ElP0.00065atm19-21.txt’,STATUS= ’OLD’)
      OPEN(22,FILE= ’ElP0.00065atm21-23.txt’,STATUS= ’OLD’)
      OPEN(23,FILE= ’ElP0.00065atm23-25.txt’,STATUS= ’OLD’)
      OPEN(24,FILE= ’ElP0.00065atm25-27.txt’,STATUS= ’OLD’)
      OPEN(25,FILE= ’ElP0.00065atm27-30.txt’,STATUS= ’OLD’)
C given the temperature
      XX = T
C Start Newton’s interpolation polynomials
      IF(XX.LT.5500)THEN
         FF = 0.5*(8.52E10+6.7666E11)
         GOTO 101
      ELSEIF(XX.GE.5500.AND.XX.LT.6000) THEN
         N = 14
         DO IROW = 1,N
            READ(10,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GE.6000.AND.XX.LE.7000) THEN
         N = 14
         DO IROW = 1,N
            READ(11,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.7000.AND.XX.LE.8000)THEN
         N = 14
         DO IROW = 1,N
            READ(12,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.8000.AND.XX.LE.9000)THEN
         N = 14
         DO IROW = 1,N
            READ(13,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.9000.AND.XX.LE.10000)THEN
         N = 14
         DO IROW = 1,N
            READ(14,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.10000.AND.XX.LE.11000)THEN
         N = 14
         DO IROW = 1,N
            READ(15,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.11000.AND.XX.LE.12000)THEN
         N = 14
         DO IROW = 1,N
            READ(16,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12000.AND.XX.LE.13000)THEN
         N = 14
         DO IROW = 1,N
            READ(17,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.13000.AND.XX.LE.15000)THEN
         N = 14
         DO IROW = 1,N
            READ(18,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.15000.AND.XX.LE.17000)THEN
         N = 14
         DO IROW = 1,N
            READ(19,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.17000.AND.XX.LE.19000)THEN
         N = 14
         DO IROW = 1,N
            READ(20,*) X(IROW),FX(IROW,1)
         ENDDO
```

```
      ELSEIF(XX.GT.19000.AND.XX.LE.21000)THEN
         N = 14
         DO IROW = 1,N
            READ(21,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.21000.AND.XX.LE.23000)THEN
         N = 14
         DO IROW = 1,N
            READ(22,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.23000.AND.XX.LE.25000)THEN
         N = 14
         DO IROW = 1,N
            READ(23,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.25000.AND.XX.LE.27000)THEN
         N = 14
         DO IROW = 1,N
            READ(24,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.27000.AND.XX.LE.28000)THEN
         N = 14
         DO IROW = 1,N
            READ(25,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.28000) THEN
         FF = 6.24869E13
         GOTO 101
      ENDIF

C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
C      WRITE(5,*)'Input the temperature (K):'
C      READ(5,*) XX
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
```

```
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
 40   CONTINUE
 101  WRITE(2,100) XX, FF,FF*1.1126535E-10
C convert from (stat mho)/cm to mho/m by time 1.126535E-10
      SIG_P = FF*1.1126535E-10

 100  FORMAT('Temp =',E10.4,'is', E16.7,'or',E16.7)

      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      RETURN
      END
C
C     Model of cathode sheath
C     Provides sheath voltage and heat flux
C     as functions of cathode temperature and current density
C
C     Inputs:
C     T (K): Cathode surface temperature
C     J (A/cm^2): Current density
C
C     Outputs:
C     VC (V): Cathode sheath voltage
C     Q (W/cm^2): Heat flux to cathode surface
C
      SUBROUTINE CMODEL( T, J, VC, Q )
C      INCLUDE 'common_declare.f'
C      IMPLICIT REAL( J )


      SAVE
      LOGICAL FIRST /.TRUE./, DONE

C     Note: Following three lines must be consistent with each other!
      PARAMETER( NT=301, NJ=2001 )
      PARAMETER( TLOW=2500, THIGH=4000, TSTEP=5 )
      PARAMETER( JLOW=0, JHIGH=10000, JSTEP=5 )

      DIMENSION VCDAT( NT,NJ ), QDAT( NT,NJ ), TVALS( NT ), JVALS( NJ )
      DIMENSION IJMIN( NT ), IJMAX( NT )

C      write(*,*) ' CMODEL called with T=',T,' J=',J

      IF( FIRST )  THEN
C     First time called: Read the data files
         OPEN( 8, FILE='Vc.dat', STATUS='UNKNOWN' )
         OPEN( 9, FILE='q.dat', STATUS='UNKNOWN' )
         READ(8,*) NR, NC
C         WRITE(*,*) 'In file Vc.dat, got ',NR, ' rows -- should be ', NT
         READ(9,*) NR, NC
C         WRITE(*,*) 'In file q.dat, got ',NR, ' rows -- should be ', NT

         DO IT=1, NT
C            write(*,*)'Reading Vc, line ', IT
            READ(8,*) ( VCDAT(IT,IJ), IJ=1, NJ )
C            write(*,*)'Reading q, line ', IT
            READ(9,*) ( QDAT(IT,IJ), IJ=1, NJ )
         ENDDO

C         WRITE(*,*) 'Finished reading Vc, Q values from data files'

C     Set the temperature and current density values
         DO IT=1, NT
            TVALS(IT) = TLOW + (IT-1) * TSTEP
         ENDDO
         DO IJ=1, NJ
            JVALS(IJ) = JLOW + (IJ-1) * JSTEP
         ENDDO
C
C     For each temperature value, find region of convergence of model
C
C     Outside region of convergence: Extrapolate
C     Sheath voltage: Increase rapidly for too-high values of J
C     Heat flux:
C       Increase rapidly for too-high values of J
C       Decrease rapidly for too-low values of J
C
C     IJMIN, IJMAX: For each T value, the smallest and largest values
```

```
C     of IJ for which QDAT is nonzero
C
         DO IT=1, NT
            DO IJ=1, NJ
               IF( QDAT(IT,IJ) .NE. 0 )  THEN
                  IJMAX(IT) = IJ
               ENDIF
            ENDDO
            DO IJ=NJ,1,-1
               IF( QDAT(IT,IJ) .NE. 0 )  THEN
                  IJMIN(IT) = IJ
               ENDIF
            ENDDO
         ENDDO
C
C         write(*,*) 'Finished finding limits of convergence'
         VCSLOPE = 10
         QSLOPE = 100
C
         FIRST = .FALSE.
      ENDIF
C
C     Check values -- for debugging
         DONE = .FALSE.
c         DO WHILE( .NOT. DONE )
         DO WHILE( .false. )
            WRITE(*,200)
 200        FORMAT( 'Enter IT, IJ (0 0 to quit): ', $ )
            READ(*,*) IT, IJ
            IF( IT .EQ. 0 ) THEN
               DONE = .TRUE.
            ELSE
               WRITE(*,*) 'Vc(',IT,IJ,') = ', VCDAT(IT,IJ)
               WRITE(*,*) 'q(',IT,IJ,') = ', QDAT(IT,IJ)
            ENDIF
         ENDDO

C     Check values -- for debugging
         DONE = .FALSE.
c         DO WHILE( .NOT. DONE )
         DO WHILE( .false. )
            WRITE(*,300)
 300        FORMAT( 'Enter IT (0 to quit): ', $ )
            READ(*,*) IT
            IF( IT .EQ. 0 ) THEN
               DONE = .TRUE.
            ELSE
```

```
               WRITE(*,*) ’IJMIN(’,IT,’) = ’, IJMIN(IT)
               WRITE(*,*) ’IJMAX(’,IT,’) = ’, IJMAX(IT)
            ENDIF
         ENDDO
C
C     Find location of input T, J in data files
C
C     XT: continuous, ranges from 0 to just below NT-1
C     IT1, IT2: Integer values below and above XT
C     PT: continuous, ranges from 0 to 1: how far across T step we are
      XT = MAX( 0.0, MIN( ( T - TLOW ) / TSTEP, NT-1.01 ) )
      IT1 = INT(XT) + 1
      IT2 = IT1 + 1
      PT = XT - INT(XT)
c      WRITE(*,*) ’Temperature ’, T, ’ is between IT = ’, IT1, IT2
c      WRITE(*,*) ’Fraction ’,PT, ’ across temperature increment’

      XJ = MAX( 0.0, MIN( ( REAL(J) - JLOW ) / JSTEP, NJ-1.01 ) )
      IJ1 = INT(XJ) + 1
      IJ2 = IJ1 + 1
      PJ = XJ - INT(XJ)
c      WRITE(*,*) ’Current density ’, J, ’ is between IJ = ’, IJ1, IJ2
c      WRITE(*,*) ’Fraction ’,PJ, ’ across current density increment’

c      write(*,*) ’IJMAX of ’,IT1,’ is ’, IJMAX(IT1)

      IF( IJ2 .GT. IJMAX(IT1) )  THEN
C     Current too high for model -- extrapolate up
c         write(*,*) ’IT1: Current too high -- ijmax is ’,IJMAX(IT1)
         IF( IJMAX(IT1) .EQ. 0 )  THEN
C     Model not converged at all for this T
c            WRITE(*,*) ’Model not converged at all for this T’
c            write(*,*) ’vcslope ’,VCSLOPE,’ qslope ’,QSLOPE
            VC1 = VCSLOPE*J
            Q1 = QSLOPE*J
         ELSE
            JMAX = JVALS(IJMAX(IT1))
c            write(*,*) ’JMAX is ’,JMAX
            VC1 = VCDAT(IT1,IJMAX(IT1)) + VCSLOPE*(J-JMAX)
            Q1 = QDAT(IT1,IJMAX(IT1)) + QSLOPE*(J-JMAX)
         ENDIF
      ELSEIF( IJ1 .LT. IJMIN(IT1) )  THEN
C     Current too low for model -- extrapolate down
c         write(*,*) ’IT1: Current too low -- ijmin is ’,IJMIN(IT1)
         JMIN = JVALS(IJMIN(IT1))
c         write(*,*) ’JMIN is ’,JMIN
         VC1 = 0
         Q1 = QDAT(IT1,IJMIN(IT1)) + QSLOPE*(J-JMIN)
      ELSE
C     Model converged here -- interpolate model data
c         write(*,*) ’IT1: Model is converged’
         VC1 = VCDAT(IT1,IJ1)*(1-PJ) + VCDAT(IT1,IJ2)*PJ
         Q1 = QDAT(IT1,IJ1)*(1-PJ) + QDAT(IT1,IJ2)*PJ
      ENDIF
c      write(*,*) ’Got VC1 = ’,VC1,’ Q1 = ’,Q1

      IF( IJ2 .GT. IJMAX(IT2) )  THEN
C     Current too high for model -- extrapolate up
         IF( IJMAX(IT2) .EQ. 0 )  THEN
C     Model not converged at all for this T
c            WRITE(*,*) ’Model not converged at all for this T’
            VC2 = VCSLOPE*J
            Q1 = QSLOPE*J
         ELSE
            JMAX = JVALS(IJMAX(IT2))
            VC2 = VCDAT(IT2,IJMAX(IT2)) + VCSLOPE*(J-JMAX)
            Q2 = QDAT(IT2,IJMAX(IT2)) + QSLOPE*(J-JMAX)
         ENDIF
      ELSEIF( IJ1 .LT. IJMIN(IT2) )  THEN
C     Current too low for model -- extrapolate down
         JMIN = JVALS(IJMIN(IT2))
         VC2 = 0
         Q2 = QDAT(IT2,IJMIN(IT2)) + QSLOPE*(J-JMIN)
      ELSE
C     Model converged here -- interpolate model data
         VC2 = VCDAT(IT2,IJ1)*(1-PJ) + VCDAT(IT2,IJ2)*PJ
         Q2 = QDAT(IT2,IJ1)*(1-PJ) + QDAT(IT2,IJ2)*PJ
      ENDIF

C     Now interpolate in the T direction
      VC = VC1*(1-PT) + VC2*PT
      Q = Q1*(1-PT) + Q2*PT

      RETURN

      END
```

Appendix D

```
C Main file for program CURRENT for 2D Cylindrical Symmetry MPD Thruster
C Originally written by: Dr.Daniel Erwin
C Revised & Updated to calculate temperature by: Thada Suksila
C Date: July 16, 2014
C Included New Sheath Model
C Updated: October 28, 2014

      INCLUDE ’common.f’

      OPEN (93,FILE = ’NSTEPS.txt’ , STATUS = ’UNKNOWN’)
      OPEN (94,FILE = ’0.5_NSTEPS.txt’, STATUS = ’UNKNOWN’)
      OPEN (95,FILE = ’main_T.txt’,STATUS = ’UNKNOWN’)
      OPEN (96,FILE = ’main_Phi.txt’, STATUS = ’UNKNOWN’)
      OPEN (97,FILE = ’main_Xj.txt’, STATUS = ’UNKNOWN’)
      OPEN (98,FILE = ’main_E.txt’, STATUS = ’UNKNOWN’)
      OPEN (99,FILE = ’mainprogram.txt’, STATUS=’UNKNOWN’)
      OPEN(100,FILE = ’11.txt’, STATUS = ’UNKNOWN’)
      OPEN(101,FILE = ’16.txt’, STATUS = ’UNKNOWN’)
      OPEN(102,FILE = ’17.txt’, STATUS = ’UNKNOWN’)
      OPEN(103,FILE = ’18.txt’, STATUS = ’UNKNOWN’)
C initial values for electrical conductivity of tungsten and plasma (mho/m)
      T = TC      ! cathode
      CALL SIGMA_C(T,SIG_C)
      CALL KTH_C(T,XKTH_C)
      CALL Cp_C(T,CAP_C)
      AA  = SIG_C
      AA1 = XKTH_C
      AA2 = CAP_C

      T = TP      ! plasma
      CALL SIGMA_P(T,SIG_P)
      CALL KTH_P(T,XKTH_P)
      CALL Cp_P(T,CP_NOR)
      CALL Z(T,XZ)
      RHO   = PR/(Rar*T)
      CAP_P = CP_NOR/XZ*Rar*1000
      CAP_P = CAP_P*RHO
      BB  = SIG_P
      BB1 = XKTH_P
      BB2 = CAP_P

C start makegrid and calculate current

      CALL MAKEGRID(AA,AA1,AA2,BB,BB1,BB2)
      CALL SUB_CURRENT            !
C      CALL ASSERT(NP.GT.0, 'Back from MAKEGRID, NP is zero !')
      CALL initial_temp
C
C check critical time 1.Von Neumann and 2. Ohmic Heating
      A = (XCAP_NODE(1)*R2BAR_NODE(1)*TAREA_NODE(1))
      B = 0.25 * XKTH_NODE(N1+3)*R2BAR_NODE(N1+3)*SSQRT_NODE(N1+3)/(3*TAREA_NODE(N1+3))
      C = Xj_NODE(N1+3)*E_NODE(N1+3)*R2BAR_NODE(N1+3)*TAREA_NODE(N1+3)
C
      DT_VON = A/(2*B)
      DT_OHM = A/(2*C)
C choose DT
       IF(DT_VON.GT.DT_OHM)THEN
          DTIME = DT_OHM
       ELSE
          DTIME = DT_VON
       ENDIF
C       WRITE(99,*) DTIME, DT_VON, DT_OHM
C divided by safty factor
       DTIME = DTIME/5000
C       WRITE(99,*) DTIME, DT_OHM_P,DT_VON_P,DT_OHM_C,DT_VON_C
C
C      NSTEPS = 100000
       NSTEPS = 1
      TOL    = 5E-3
      TIME   = 0.0
C
C Solve for Temperature at the grid points
      DO 500 ISTEP = 1, NSTEPS
         DTT = ISTEP * DTIME
C #2 Calculates for average triangle temperature
         DO IT = 1, NT
            TEMP_TRI(IT) = (TEMP(ICTP(1,IT))+TEMP(ICTP(2,IT))+TEMP(ICTP(3,IT)))/3
C            WRITE(99,*) IT, TEMP_TRI(IT), ICTP(1,IT),ICTP(2,IT), ICTP(3,IT)
         ENDDO
C #3 case N1=6,N2=4,N3=2,N4=4,N5=2
         IF(N1.EQ.6.AND.N2.EQ.4.AND.N3.EQ.2.AND.N4.EQ.4.AND.N5.EQ.2)THEN
```

```
            DO IT = 1, NT
               IF(IT.LE.N1+1)THEN    !column1
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  CALL KTH_C(T,XKTH_C)
                  CALL Cp_C(T,CAP_C)
                  SIGMA(IT) = SIG_C
                  XKTH(IT) = XKTH_C
                  XCAP(IT) = CAP_C
C                  WRITE(99,*) T,IT, SIGMA(IT), 'column 1'
               ELSEIF(IT.GE.N1*2+1.AND.IT.LE.(N1*2+1)+((N2-1)*2)-1) THEN ! column 2
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  CALL KTH_C(T,XKTH_C)
                  CALL Cp_C(T,CAP_C)
                  SIGMA(IT) = SIG_C
                  XKTH(IT) = XKTH_C
                  XCAP(IT) = CAP_C
C                  WRITE(99,*) T,IT, SIGMA(IT), 'column 2'
               ELSE                  ! plasma
                  T = TEMP_TRI(IT)
                  CALL SIGMA_P(T,SIG_P)
                  CALL KTH_P(T,XKTH_P)
                  CALL Cp_P(T,CP_NOR)
                  CALL Z(T,XZ)
                  RHO   = PR/(Rar*T)
                  CAP_P = CP_NOR/XZ*Rar*1000
                  CAP_P = CAP_P*RHO
                  SIGMA(IT) = SIG_P
                  XKTH(IT) = XKTH_P
                  XCAP(IT) = CAP_P
C                  WRITE(99,*) T,IT, SIGMA(IT), 'plasma'
               ENDIF
            ENDDO
C case N1=24,N2=8,N3=2,N4=10,N5=8
         ELSEIF(N1.EQ.24.AND.N2.EQ.8.AND.N3.EQ.10.AND.N4.EQ.20.AND.N5.EQ.8)THEN
            DO IT = 1, NT
               IF(IT.GE.1.AND.IT.LE.2*N2-1)THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSEIF(IT.GE.2*N1+1.AND.IT.LE.(2*N1+1)+(2*N2-3))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
```

```
               ELSEIF(IT.GE.4*N1+1.AND.IT.LE.(4*N1+1)+(2*N2-4))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSEIF(IT.GE.6*N1+1.AND.IT.LE.(6*N1+1)+(2*N2-5))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSEIF(IT.GE.8*N1+1.AND.IT.LE.(8*N1+1)+(2*N2-6))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSEIF(IT.GE.10*N1+1.AND.IT.LE.(10*N1+1)+(2*N2-7))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSEIF(IT.GE.12*N1+1.AND.IT.LE.(12*N1+1)+(2*N2-8))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSEIF(IT.GE.14*N1+1.AND.IT.LE.(14*N1+1)+(2*N2-9))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSEIF(IT.GE.16*N1+1.AND.IT.LE.(16*N1+1)+(2*N2-10))THEN
                  T = TEMP_TRI(IT)
                  CALL SIGMA_C(T,SIG_C)
                  SIGMA(IT) = SIG_C
C                  WRITE(99,*) IT, SIGMA(IT), 'cathode'
               ELSE
                  T = TEMP_TRI(IT)
                  CALL SIGMA_P(T,SIG_P)
                  SIGMA(IT) = SIG_P
C                  WRITE(99,*) IT, SIGMA(IT), 'plasma'
               ENDIF
            ENDDO
         ELSE
            GOTO 66
         ENDIF
C
C
```

```
C #4 calculates the Xj, E, SSQRT etc.
         CALL SUB_CURRENT
C
C #5 Updated Xj_NODE,E_NODE (at the nodes) and calculates average
C temperature at the nodes
         DO I = 1, NP
            DO IS=1, 6
               IPS = ICPP(IS,I)    ! Point at other end of side IS
               ISP = MOD(IS,6) + 1 ! Adjacent side (counterclockwise)
               IITP = IS           ! Adjacent triangle index (counterclockwise)
               ITP = ICPT(IITP,I)  ! Adjacent triangle
               E_NODE(I) = (E_tot(ICPT(1,I))+E_tot(ICPT(2,I))+E_tot(ICPT(3,I))+
     &                    E_tot(ICPT(4,I))+E_tot(ICPT(5,I))+E_tot(ICPT(6,I)))
               Xj_NODE(I)    = (Xj_TOT(ICPT(1,I))+Xj_TOT(ICPT(2,I))+Xj_TOT(ICPT(3,I))+
     &              Xj_TOT(ICPT(4,I))+Xj_TOT(ICPT(5,I))+Xj_TOT(ICPT(6,I)))
               SIGMA_NODE(I) = (SIGMA(ICPT(1,I))+SIGMA(ICPT(2,I))+SIGMA(ICPT(3,I))+
     &              SIGMA(ICPT(4,I))+SIGMA(ICPT(5,I))+SIGMA(ICPT(6,I)))
               XKTH_NODE(I)  = (XKTH(ICPT(1,I))+XKTH(ICPT(2,I))+XKTH(ICPT(3,I))+
     &              XKTH(ICPT(4,I))+XKTH(ICPT(5,I))+XKTH(ICPT(6,I)))
               XCAP_NODE(I)  = (XCAP(ICPT(1,I))+XCAP(ICPT(2,I))+XCAP(ICPT(3,I))+
     &              XCAP(ICPT(4,I))+XCAP(ICPT(5,I))+XCAP(ICPT(6,I)))
            ENDDO
         ENDDO
C# 5.1 Calculate sum of (Ti-T)
         DO I = 1, NP
            T1 = ABS(TEMP(ICPP(1,I)) - TEMP(I))
            T2 = ABS(TEMP(ICPP(2,I)) - TEMP(I))
            T3 = ABS(TEMP(ICPP(3,I)) - TEMP(I))
            T4 = ABS(TEMP(ICPP(4,I)) - TEMP(I))
            T5 = ABS(TEMP(ICPP(5,I)) - TEMP(I))
            T6 = ABS(TEMP(ICPP(6,I)) - TEMP(I))

            TSUM(I) = T1 + T2 + T3 + T4 + T5 + T6

C            WRITE(99,*) I,T1,T2, T3, T4, T5, T6, TSUM(I)
         ENDDO

C
C #6 calculates for temperature
         DO IP = 1, NP
            IF(ICOMPUTE(IP).EQ.0)THEN
               IF(ISH(IP).EQ.1)THEN ! at the cathode surface
                  T = TEMP(IP)
                  J = Xj_NODE(IP)
                  CALL CMODEL( T, J, VC, QQ )

                  VCC(IP) = VC
                  QQQ(IP) = QQ*TAREA_NODE(IP)

                  A = DTIME/(XCAP_NODE(IP)*R2BAR_NODE(IP)*TAREA_NODE(IP))
                  B = 0.25 * XKTH_NODE(IP)*R2BAR_NODE(IP)*SSQRT_NODE(IP)
     & /(3*TAREA_NODE(IP))
                  C = Xj_NODE(IP)*E_NODE(IP)*R2BAR_NODE(IP)*TAREA_NODE(IP)
                  D = QQQ(IP)

                  TNEW(IP) = TEMP(IP) + A* ( B*TSUM(IP) + C + QQQ(IP) ) ! Eq.7.21

                  TEMP(IP) = TNEW(IP)

               ELSE
                  A = DTIME/(XCAP_NODE(IP)*R2BAR_NODE(IP)*TAREA_NODE(IP))
                  B = 0.25 * XKTH_NODE(IP)*R2BAR_NODE(IP)*SSQRT_NODE(IP)
     & /(3*TAREA_NODE(IP))
                  C = Xj_NODE(IP)*E_NODE(IP)*R2BAR_NODE(IP)*TAREA_NODE(IP)

                  TNEW(IP) = TEMP(IP) + A* ( B*TSUM(IP) + C ) ! Eq.7.20

                  TEMP(IP) = TNEW(IP)
               ENDIF
            ENDIF
C Set cathoe base back to 1500 K
            IF(IP.LE.N2+1)THEN
               TNEW(IP) = TB
               TEMP(IP) = TNEW(IP)
            ENDIF
         ENDDO

C Print out at every step 4 steps
         IPP = ISTEP - (ISTEP/20)*20
         IF(IPP.EQ.0)THEN
            WRITE(95,300) TIME, (TEMP(I), I = 1, NP)
c$$$C           WRITE(96,300) TIME, (PHIVAL(I),I = 1,NP)
c$$$C           WRITE(97,300) TIME, (Xj_NODE(I),I=1,NP)
c$$$C           WRITE(98,300) TIME, (E_NODE(IP), I  = 1, NP)
            WRITE(100,*) TIME,11, TEMP(11), Xj_NODE(11), QQQ(11),VCC(11),'s'
            WRITE(101,*) TIME,16, TEMP(16), Xj_NODE(16), QQQ(16),VCC(16),'s'
            WRITE(102,*) TIME,17, TEMP(17), Xj_NODE(17), QQQ(17),VCC(17),'s'
            WRITE(103,*) TIME,18, TEMP(18), Xj_NODE(18), QQQ(18),VCC(18),'s'

         ENDIF
c$$$
         TIME = TIME + DTIME
c$$$
```

```
c$$$C        WRITE(99,*) ISTEP,TIME, IP, TEMP(11),TEMP(16),TEMP(17),TEMP(18), QQ, VC, J

 500  CONTINUE
 300  FORMAT(F15.6,3X,35(F15.4,2X))

      CLOSE(93)
      CLOSE(94)
      CLOSE(95)
      CLOSE(96)
      CLOSE(97)
      CLOSE(98)
      CLOSE(99)
      CLOSE(100)
      CLOSE(101)
      CLOSE(102)
      CLOSE(103)
 66   STOP
      END
C---------------------------------------------------------------------
C ---------------------- SUBROUTINE HEAT CAPACITY ---------------------
C  subroutine to calculate the temperature dependent Heat capacity
C  of the Cathode and Anode
C ---------------------------------------------------------------------
      SUBROUTINE Cp_C( T, CAP_C )
      INCLUDE 'common.f'
C     Compute heat capacity of tungsten in J/(K m^3)

      RHOMOL_C = RHO_C / XMAMU_C      ! gmol/m^3

      AVOLJ = AMOLCAL_C * XJPERCAL * RHOMOL_C
      BVOLJ = BMOLCAL_C * XJPERCAL * RHOMOL_C

      CAP_C = AVOLJ + 1E-3*T*BVOLJ
C Convert from J/(K m^3) to J/(Kg K) by 19.25E3 (density at room temperature (Wiki)
C      CAP_C = CAP_C / (RHO_C*1E-3)

      RETURN
      END

C ---------------- SUBROUTINE CTCOND ---------------------------
C  subroutine to calculate the temperature dependent thermal
C  conductivity of Cathode(Tungsten)
C  TUNGSTEN VALUES solid and liquid  [W/m/K]
C --------------------------------------------------------------
      SUBROUTINE KTH_C(T, XKTH_C)
      INCLUDE 'common.f'
      IF ( T .LE. 3670 ) THEN
```

```
         XKTH_C = A0 + A1*EXP(-A2*T) + A3*EXP(-A4*T)
      ELSE
         XKTH_C = B0 + B1*T + B3*T*T
      ENDIF

      RETURN
      END
C ------------------------ SUBROUTINE SIGMA_C -------------------------
C This subroutine calculates the temperature dependent Electrical
C conductivity of cathode(Tungsten) [mho/m]
C A linear curve fit of the electrical resistivity in page 99 eq.3.10 Goodfellow
C --------------------------------------------------------------------
      SUBROUTINE SIGMA_C(T,SIG_C)
      INCLUDE 'common.f'

      XRHOE_C  = (C0 + C01*T)*0.00000001
      SIG_C = 1/XRHOE_C

      RETURN
      END
C------------------------- SUBROUTINE CP Normalized --------------------
C This subroutine calculates the Cp of argon without any unit
C----------------------------------------------------------------------
      SUBROUTINE Cp_P(T,CP_NOR)
      DIMENSION X(200), FX(200,200)
      OPEN(11,FILE= 'P0.0006_5000-5500.txt', STATUS = 'OLD')
      OPEN(12,FILE= 'P0.0006_5500-5750.txt', STATUS = 'OLD')
      OPEN(13,FILE= 'P0.0006_5750-6000.txt', STATUS = 'OLD')
      OPEN(14,FILE= 'P0.0006_6000-6500.txt',STATUS = 'OLD')
      OPEN(15,FILE= 'P0.0006_6500-7000.txt',STATUS= 'OLD')
      OPEN(16,FILE= 'P0.0006_7000-7250.txt',STATUS= 'OLD')
      OPEN(17,FILE= 'P0.0006_7250-7400.txt',STATUS= 'OLD')
      OPEN(18,FILE= 'P0.0006_7400-7600.txt',STATUS= 'OLD')
      OPEN(19,FILE= 'P0.0006_7600-7800.txt',STATUS= 'OLD')
      OPEN(20,FILE= 'P0.0006_7800-8000.txt',STATUS= 'OLD')
      OPEN(21,FILE= 'P0.0006_8000-8100.txt',STATUS= 'OLD')
      OPEN(22,FILE= 'P0.0006_8100-8200.txt',STATUS= 'OLD')
      OPEN(23,FILE= 'P0.0006_8200-8300.txt',STATUS= 'OLD')
      OPEN(24,FILE= 'P0.0006_8300-8600.txt',STATUS= 'OLD')
      OPEN(25,FILE= 'P0.0006_8600-8800.txt',STATUS= 'OLD')
      OPEN(26,FILE= 'P0.0006_8800-8900.txt',STATUS= 'OLD')
      OPEN(27,FILE= 'P0.0006_8900-9000.txt',STATUS= 'OLD')
      OPEN(28,FILE= 'P0.0006_9000-9100.txt',STATUS= 'OLD')
      OPEN(29,FILE= 'P0.0006_9100-9200.txt',STATUS= 'OLD')
      OPEN(30,FILE= 'P0.0006_9200-9300.txt',STATUS= 'OLD')
      OPEN(31,FILE= 'P0.0006_9300-9400.txt',STATUS= 'OLD')
      OPEN(32,FILE= 'P0.0006_9400-9500.txt',STATUS= 'OLD')
      OPEN(33,FILE= ’P0.0006_9500-9600.txt’,STATUS= ’OLD’)
      OPEN(34,FILE= ’P0.0006_9600-9700.txt’,STATUS= ’OLD’)
      OPEN(35,FILE= ’P0.0006_9700-9800.txt’,STATUS= ’OLD’)
      OPEN(36,FILE= ’P0.0006_9800-9900.txt’,STATUS= ’OLD’)
      OPEN(37,FILE= ’P0.0006_9900-10000.txt’,STATUS= ’OLD’)
      OPEN(38,FILE= ’P0.0006_10000-10100.txt’,STATUS= ’OLD’)
      OPEN(39,FILE= ’P0.0006_10100-10200.txt’,STATUS= ’OLD’)
      OPEN(40,FILE= ’P0.0006_10200-10300.txt’,STATUS= ’OLD’)
      OPEN(41,FILE= ’P0.0006_10300-10400.txt’,STATUS= ’OLD’)
      OPEN(42,FILE= ’P0.0006_10400-10500.txt’,STATUS= ’OLD’)
      OPEN(43,FILE= ’P0.0006_10500-10600.txt’,STATUS= ’OLD’)
      OPEN(44,FILE= ’P0.0006_10600-10700.txt’,STATUS= ’OLD’)
      OPEN(45,FILE= ’P0.0006_10700-10800.txt’,STATUS= ’OLD’)
      OPEN(46,FILE= ’P0.0006_10800-10900.txt’,STATUS= ’OLD’)
      OPEN(47,FILE= ’P0.0006_10900-11000.txt’,STATUS= ’OLD’)
      OPEN(48,FILE= ’P0.0006_11000-11200.txt’,STATUS= ’OLD’)
      OPEN(49,FILE= ’P0.0006_11200-11400.txt’,STATUS= ’OLD’)
      OPEN(50,FILE= ’P0.0006_11400-11600.txt’,STATUS= ’OLD’)
      OPEN(51,FILE= ’P0.0006_11600-11800.txt’,STATUS= ’OLD’)
      OPEN(52,FILE= ’P0.0006_11800-12000.txt’,STATUS= ’OLD’)
      OPEN(53,FILE= ’P0.0006_12000-12300.txt’,STATUS= ’OLD’)
      OPEN(54,FILE= ’P0.0006_12300-12600.txt’,STATUS= ’OLD’)
      OPEN(55,FILE= ’P0.0006_12600-13000.txt’,STATUS= ’OLD’)
C given the temperature
      XX  = T
      IF(XX.LE.5000)THEN
         FF = 2.378552973
         GOTO 101
      ELSEIF(XX.GT.13000)THEN
         FF = 0.00017581*XX + 3.69888312
         GOTO 101
      ENDIF
      IF(XX.GT.5000.AND.XX.LE.5500) THEN
         N = 7
         DO IROW = 1,N
            READ(11,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.5500.AND.XX.LE.5750)THEN
         N = 10
         DO IROW = 1,N
            READ(12,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.5750.AND.XX.LE.6000)THEN
         N = 8
         DO IROW = 1,N
            READ(13,*) X(IROW),FX(IROW,1)
         ENDDO
```

```
ELSEIF(XX.GT.6000.AND.XX.LE.6500)THEN
   N = 12
   DO IROW = 1,N
      READ(14,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.6500.AND.XX.LE.7000)THEN
   N = 12
   DO IROW = 1,N
      READ(15,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7000.AND.XX.LE.7250)THEN
   N = 7
   DO IROW = 1,N
      READ(16,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7250.AND.XX.LE.7400)THEN
   N = 8
   DO IROW = 1,N
      READ(17,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7400.AND.XX.LE.7600)THEN
   N = 8
   DO IROW = 1,N
      READ(18,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7600.AND.XX.LE.7800)THEN
   N = 8
   DO IROW = 1,N
      READ(19,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7800.AND.XX.LE.8000)THEN
   N = 8
   DO IROW = 1,N
      READ(20,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8000.AND.XX.LE.8100)THEN
   N = 8
   DO IROW = 1,N
      READ(21,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8100.AND.XX.LE.8200)THEN
   N = 8
   DO IROW = 1,N
      READ(22,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8200.AND.XX.LE.8300)THEN
   N = 8
   DO IROW = 1,N
      READ(23,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8300.AND.XX.LE.8600)THEN
   N = 9
   DO IROW = 1,N
      READ(24,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8600.AND.XX.LE.8800)THEN
   N = 8
   DO IROW = 1,N
      READ(25,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8800.AND.XX.LE.8900)THEN
   N = 7
   DO IROW = 1,N
      READ(26,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8900.AND.XX.LE.9000)THEN
   N = 8
   DO IROW = 1,N
      READ(27,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9000.AND.XX.LE.9100)THEN
   N = 8
   DO IROW = 1,N
      READ(28,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9100.AND.XX.LE.9200)THEN
   N = 8
   DO IROW = 1,N
      READ(29,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9200.AND.XX.LE.9300)THEN
   N = 8
   DO IROW = 1,N
      READ(30,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9300.AND.XX.LE.9400)THEN
   N = 8
   DO IROW = 1,N
      READ(31,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9400.AND.XX.LE.9500)THEN
   N = 8
   DO IROW = 1,N
      READ(32,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9500.AND.XX.LE.9600)THEN
   N = 8
   DO IROW = 1,N
      READ(33,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9600.AND.XX.LE.9700)THEN
   N = 8
   DO IROW = 1,N
      READ(34,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9700.AND.XX.LE.9800)THEN
   N = 8
   DO IROW = 1,N
      READ(35,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9800.AND.XX.LE.9900)THEN
   N = 8
   DO IROW = 1,N
      READ(36,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9900.AND.XX.LE.10000)THEN
   N = 8
   DO IROW = 1,N
      READ(37,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10000.AND.XX.LE.10100)THEN
   N = 8
   DO IROW = 1,N
      READ(38,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10100.AND.XX.LE.10200)THEN
   N = 8
   DO IROW = 1,N
      READ(39,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10200.AND.XX.LE.10300)THEN
   N = 8
   DO IROW = 1,N
      READ(40,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10300.AND.XX.LE.10400)THEN
   N = 8
   DO IROW = 1,N
      READ(41,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10400.AND.XX.LE.10500)THEN
```

```
   N = 8
   DO IROW = 1,N
      READ(42,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10500.AND.XX.LE.10600)THEN
   N = 8
   DO IROW = 1,N
      READ(43,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10600.AND.XX.LE.10700)THEN
   N = 8
   DO IROW = 1,N
      READ(44,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10700.AND.XX.LE.10800)THEN
   N = 8
   DO IROW = 1,N
      READ(45,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10800.AND.XX.LE.10900)THEN
   N = 8
   DO IROW = 1,N
      READ(46,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10900.AND.XX.LE.11000)THEN
   N = 8
   DO IROW = 1,N
      READ(47,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11000.AND.XX.LE.11200)THEN
   N = 9
   DO IROW = 1,N
      READ(48,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11200.AND.XX.LE.11400)THEN
   N = 8
   DO IROW = 1,N
      READ(49,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11400.AND.XX.LE.11600)THEN
   N = 7
   DO IROW = 1,N
      READ(50,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11600.AND.XX.LE.11800)THEN
   N = 8
   DO IROW = 1,N
```

```
            READ(51,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.11800.AND.XX.LE.12000)THEN
         N = 7
         DO IROW = 1,N
            READ(52,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12000.AND.XX.LE.12300)THEN
         N = 8
         DO IROW = 1,N
            READ(53,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12300.AND.XX.LE.12600)THEN
         N = 9
         DO IROW = 1,N
            READ(54,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12600.AND.XX.LE.13000)THEN
         N = 10
         DO IROW = 1,N
            READ(55,*) X(IROW),FX(IROW,1)
         ENDDO
      ENDIF
C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
         CP_NOR = FF
 40   CONTINUE
C 101  WRITE(89,100) XX, FF
 101  ZAP = 1
 100  FORMAT('value of F(X) at x = ', E10.4,'is', E16.7)
C Close all open files
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      CLOSE(26)
      CLOSE(27)
      CLOSE(28)
      CLOSE(29)
      CLOSE(30)
      CLOSE(31)
      CLOSE(32)
      CLOSE(33)
      CLOSE(34)
      CLOSE(35)
      CLOSE(36)
      CLOSE(37)
      CLOSE(38)
      CLOSE(39)
      CLOSE(40)
      CLOSE(41)
      CLOSE(42)
      CLOSE(43)
      CLOSE(44)
      CLOSE(45)
      CLOSE(46)
      CLOSE(47)
      CLOSE(48)
      CLOSE(49)
      CLOSE(50)
      CLOSE(51)
      CLOSE(52)
      CLOSE(53)
      CLOSE(54)
      CLOSE(55)
      RETURN
      END
C ------------------ SUBROUTINE Z (compressibility factor ----------
C This pressure is 0.0006atm to calculate the compressibility factor of gas
C Program for computing F(X) at a given X
C using Newton's divided-difference interpolating polynomials
C Written By: Thada Suksila
C Date: October 27, 2013
C Calculate for Z at pressure 0.0006 atm
      SUBROUTINE Z(XI,FF)
      DIMENSION X(200), FX(200,200)
      OPEN(11,FILE= 'Z0.0006_6000-6500.txt', STATUS = 'OLD')
      OPEN(12,FILE= 'Z0.0006_6500-7000.txt', STATUS = 'OLD')
      OPEN(13,FILE= 'Z0.0006_7000-7500.txt', STATUS = 'OLD')
      OPEN(14,FILE= 'Z0.0006_7500-8000.txt',STATUS = 'OLD')
      OPEN(15,FILE= 'Z0.0006_8000-8500.txt',STATUS= 'OLD')
      OPEN(16,FILE= 'Z0.0006_8500-9000.txt',STATUS= 'OLD')
      OPEN(17,FILE= 'Z0.0006_9000-9500.txt',STATUS= 'OLD')
      OPEN(18,FILE= 'Z0.0006_9500-10000.txt',STATUS= 'OLD')
      OPEN(19,FILE= 'Z0.0006_10000-10500.txt',STATUS= 'OLD')
      OPEN(20,FILE= 'Z0.0006_10500-11000.txt',STATUS= 'OLD')
      OPEN(21,FILE= 'Z0.0006_11000-11500.txt',STATUS= 'OLD')
      OPEN(22,FILE= 'Z0.0006_11500-12000.txt',STATUS= 'OLD')
      OPEN(23,FILE= 'Z0.0006_12000-12500.txt',STATUS= 'OLD')
      OPEN(24,FILE= 'Z0.0006_12500-13000.txt',STATUS= 'OLD')
      OPEN(25,FILE= 'Z0.0006_13000-13500.txt',STATUS= 'OLD')
      OPEN(26,FILE= 'Z0.0006_13500-14000.txt',STATUS= 'OLD')
C Start to calculate using given temperature
      XX  = XI
      IF(XX.LE.6000)THEN
         FF = 1.0
         GOTO 101
      ELSEIF(XX.GT.14000)THEN
         FF = 2.0
         GOTO 101
      ENDIF
      IF(XX.GT.6000.AND.XX.LE.6500) THEN
         N = 5
         DO IROW = 1,N
            READ(11,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.6500.AND.XX.LE.7000)THEN
         N = 7
         DO IROW = 1,N
            READ(12,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.7000.AND.XX.LE.7500)THEN
         N = 7
         DO IROW = 1,N
            READ(13,*) X(IROW),FX(IROW,1)
```

```
   ENDDO
ELSEIF(XX.GT.7500.AND.XX.LE.8000)THEN
   N = 6
   DO IROW = 1,N
      READ(14,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8000.AND.XX.LE.8500)THEN
   N = 7
   DO IROW = 1,N
      READ(15,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8500.AND.XX.LE.9000)THEN
   N = 7
   DO IROW = 1,N
      READ(16,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9000.AND.XX.LE.9500)THEN
   N = 7
   DO IROW = 1,N
      READ(17,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9500.AND.XX.LE.10000)THEN
   N = 6
   DO IROW = 1,N
      READ(18,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10000.AND.XX.LE.10500)THEN
   N = 7
   DO IROW = 1,N
      READ(19,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10500.AND.XX.LE.11000)THEN
   N = 6
   DO IROW = 1,N
      READ(20,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11000.AND.XX.LE.11500)THEN
   N = 6
   DO IROW = 1,N
      READ(21,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11500.AND.XX.LE.12000)THEN
   N = 6
   DO IROW = 1,N
      READ(22,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.12000.AND.XX.LE.12500)THEN
```

```
         N = 6
         DO IROW = 1,N
            READ(23,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12500.AND.XX.LE.13000)THEN
         N = 6
         DO IROW = 1,N
            READ(24,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.13000.AND.XX.LE.13500)THEN
         N = 6
         DO IROW = 1,N
            READ(25,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.13500.AND.XX.LE.14000)THEN
         N = 4
         DO IROW = 1,N
            READ(26,*) X(IROW),FX(IROW,1)
         ENDDO
      ENDIF
C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
 40   CONTINUE
C 101  WRITE(89,100) XX, FF
 101  ZAP = 1
 100  FORMAT(’value of F(X) at x = ’, E10.4,’is’, E16.7)
C Close all open files
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      CLOSE(26)
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE KTH_P(T,XKTH_P)
C The pressure is 0.00065atm for thermal conductivity of argon
C Program for computing F(X) at a given X
C using Newton's divided-difference interpolating polynomials
      DOUBLE PRECISION X, FX
      DIMENSION X(506), FX(506,506)
      OPEN(12,FILE= 'ThP0.0006atm7-8.txt', STATUS = 'OLD')
      OPEN(13,FILE= 'ThP0.0006atm8-9.txt', STATUS = 'OLD')
      OPEN(14,FILE= 'ThP0.0006atm9-10.txt',STATUS = 'OLD')
      OPEN(15,FILE= 'ThP0.0006atm10-11.txt',STATUS= 'OLD')
      OPEN(16,FILE= 'ThP0.0006atm11-12.txt',STATUS= 'OLD')
      OPEN(17,FILE= 'ThP0.0006atm12-13.txt',STATUS= 'OLD')
      OPEN(18,FILE= 'ThP0.0006atm13-15.txt',STATUS= 'OLD')
      OPEN(19,FILE= 'ThP0.0006atm15-17.txt',STATUS= 'OLD')
      OPEN(20,FILE= 'ThP0.0006atm17-19.txt',STATUS= 'OLD')
      OPEN(21,FILE= 'ThP0.0006atm19-21.txt',STATUS= 'OLD')
      OPEN(22,FILE= 'ThP0.0006atm21-23.txt',STATUS= 'OLD')
      OPEN(23,FILE= 'ThP0.0006atm23-25.txt',STATUS= 'OLD')
      OPEN(24,FILE= 'ThP0.0006atm25-27.txt',STATUS= 'OLD')
      OPEN(25,FILE= 'ThP0.0006atm27-28.txt',STATUS= 'OLD')
C given the temperature
      XX = T
      IF(XX.GT.0.AND.XX.LT.7000) THEN
         FF = 0.0171*XX**1.5951
         GOTO 101
      ELSEIF(XX.GE.7000.AND.XX.LE.8000)THEN
         N = 14
         DO IROW = 1,N
            READ(12,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.8000.AND.XX.LE.9000)THEN
         N = 14
         DO IROW = 1,N
            READ(13,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.9000.AND.XX.LE.10000)THEN
         N = 14
         DO IROW = 1,N
            READ(14,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.10000.AND.XX.LE.11000)THEN
         N = 14
         DO IROW = 1,N
            READ(15,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.11000.AND.XX.LE.12000)THEN
         N = 14
         DO IROW = 1,N
            READ(16,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.12000.AND.XX.LE.13000)THEN
         N = 14
         DO IROW = 1,N
            READ(17,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.13000.AND.XX.LE.15000)THEN
         N = 14
         DO IROW = 1,N
            READ(18,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.15000.AND.XX.LE.17000)THEN
         N = 14
         DO IROW = 1,N
            READ(19,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.17000.AND.XX.LE.19000)THEN
         N = 14
         DO IROW = 1,N
            READ(20,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.19000.AND.XX.LE.21000)THEN
         N = 14
         DO IROW = 1,N
            READ(21,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.21000.AND.XX.LE.23000)THEN
         N = 14
         DO IROW = 1,N
            READ(22,*) X(IROW),FX(IROW,1)
         ENDDO
```

```
      ELSEIF(XX.GT.23000.AND.XX.LE.25000)THEN
         N = 14
         DO IROW = 1,N
            READ(23,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.25000.AND.XX.LE.27000)THEN
         N = 14
         DO IROW = 1,N
            READ(24,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.27000.AND.XX.LE.28000)THEN
         N = 14
         DO IROW = 1,N
            READ(25,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.28000)THEN
         FF = 225143.0861
         GOTO 101
      ENDIF
C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
C      WRITE(5,*)'Input the temperature (K):'
C      READ(5,*) XX
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
         TH  = FF
 40   CONTINUE
C 101  WRITE(2,100) XX, FF,FF*0.00001
 101  ZAP = 1
C Convert from erg/(K cm s) to Watt/(m K) by time 1E-5
      XKTH_P = FF*0.00001

 100  FORMAT('Temp =',E10.4,'is', E16.7,'or',E16.7)
 200  FORMAT(1X,E21.16,4X,E21.16)
```

```
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      RETURN
      END
C---------------------------------------------------------------
      SUBROUTINE SIGMA_P(T,SIG_P)
C This pressure is 0.00065atm Electrical Conductivity
C Program for computing F(X) at a given X
C using Newton’s divided-difference interpolating polynomials
      DOUBLE PRECISION X, FX
      DIMENSION X(506), FX(506,506)
      OPEN(10,FILE= ’ElP0.00065atm5.5-6.txt’,STATUS= ’OLD’)
      OPEN(11,FILE= ’ElP0.00065atm6-7.txt’, STATUS = ’OLD’)
      OPEN(12,FILE= ’ElP0.00065atm7-8.txt’, STATUS = ’OLD’)
      OPEN(13,FILE= ’ElP0.00065atm8-9.txt’, STATUS = ’OLD’)
      OPEN(14,FILE= ’ElP0.00065atm9-10.txt’,STATUS = ’OLD’)
      OPEN(15,FILE= ’ElP0.00065atm10-11.txt’,STATUS= ’OLD’)
      OPEN(16,FILE= ’ElP0.00065atm11-12.txt’,STATUS= ’OLD’)
      OPEN(17,FILE= ’ElP0.00065atm12-13.txt’,STATUS= ’OLD’)
      OPEN(18,FILE= ’ElP0.00065atm13-15.txt’,STATUS= ’OLD’)
      OPEN(19,FILE= ’ElP0.00065atm15-17.txt’,STATUS= ’OLD’)
      OPEN(20,FILE= ’ElP0.00065atm17-19.txt’,STATUS= ’OLD’)
      OPEN(21,FILE= ’ElP0.00065atm19-21.txt’,STATUS= ’OLD’)
      OPEN(22,FILE= ’ElP0.00065atm21-23.txt’,STATUS= ’OLD’)
      OPEN(23,FILE= ’ElP0.00065atm23-25.txt’,STATUS= ’OLD’)
      OPEN(24,FILE= ’ElP0.00065atm25-27.txt’,STATUS= ’OLD’)
      OPEN(25,FILE= ’ElP0.00065atm27-30.txt’,STATUS= ’OLD’)
C given the temperature
      XX = T
C Start Newton’s interpolation polynomials
      IF(XX.LT.300) THEN
         FF = 8.34E-12
         GOTO 101
      ELSEIF(XX.GE.300.AND.XX.LT.500) THEN
```

```
   FF = 0.5*(8.34E-12+8.27E-8)
   GOTO 101
ELSEIF(XX.GE.500.AND.XX.LT.700) THEN
   FF = 0.5*(8.27E-8+3.55E-5)
   GOTO 101
ELSEIF(XX.GE.700.AND.XX.LT.900) THEN
   FF = 0.5*(3.55E-5+3.28E-3)
   GOTO 101
ELSEIF(XX.GE.900.AND.XX.LT.1100)THEN
   FF = 0.5*(3.28E-3+1.22E-1)
   GOTO 101
ELSEIF(XX.GE.1100.AND.XX.LT.1300)THEN
   FF = 0.5*(1.22E-1+2.47E0)
   GOTO 101
ELSEIF(XX.GE.1300.AND.XX.LT.1500)THEN
   FF = 0.5*(2.47E0+3.25E1)
   GOTO 101
ELSEIF(XX.GE.1500.AND.XX.LT.1700)THEN
   FF = 0.5*(3.25E1+3.1E2)
   GOTO 101
ELSEIF(XX.GE.1700.AND.XX.LT.2100)THEN
   FF = 0.5*(3.1E2+1.39E4)
   GOTO 101
ELSEIF(XX.GE.2100.AND.XX.LT.2300)THEN
   FF = 0.5*(1.39E4+7.18E4)
   GOTO 101
ELSEIF(XX.GE.2300.AND.XX.LT.2500)THEN
   FF = 0.5*(7.18E4+3.22E5)
   GOTO 101
ELSEIF(XX.GE.2500.AND.XX.LT.2700)THEN
   FF = 0.5*(3.22E5+1.29E6)
   GOTO 101
ELSEIF(XX.GE.2700.AND.XX.LT.2900)THEN
   FF = 0.5*(1.29E6+4.67E6)
   GOTO 101
ELSEIF(XX.GE.2900.AND.XX.LT.3000)THEN
   FF = 0.5*(4.67E6+8.6E6)
   GOTO 101
ELSEIF(XX.GE.3000.AND.XX.LT.3500)THEN
   FF = 0.5*(8.6E6+1.38E8)
   GOTO 101
ELSEIF(XX.GE.3500.AND.XX.LT.4000)THEN
   FF = 0.5*(1.38E8+1.53E9)
   GOTO 101
ELSEIF(XX.GE.4000.AND.XX.LT.4500)THEN
   FF = 0.5*(1.53E9+1.28E10)
   GOTO 101
```

```
ELSEIF(XX.GE.4500.AND.XX.LT.5000)THEN
   FF = 0.5*(1.28E10+8.52E10)
   GOTO 101
ELSEIF(XX.GE.5000.AND.XX.LT.5500)THEN
   FF = 0.5*(8.52E10+6.7666E11)
   GOTO 101
ELSEIF(XX.GE.5500.AND.XX.LT.6000) THEN
   N = 14
   DO IROW = 1,N
      READ(10,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GE.6000.AND.XX.LE.7000) THEN
   N = 14
   DO IROW = 1,N
      READ(11,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.7000.AND.XX.LE.8000)THEN
   N = 14
   DO IROW = 1,N
      READ(12,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.8000.AND.XX.LE.9000)THEN
   N = 14
   DO IROW = 1,N
      READ(13,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.9000.AND.XX.LE.10000)THEN
   N = 14
   DO IROW = 1,N
      READ(14,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.10000.AND.XX.LE.11000)THEN
   N = 14
   DO IROW = 1,N
      READ(15,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.11000.AND.XX.LE.12000)THEN
   N = 14
   DO IROW = 1,N
      READ(16,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.12000.AND.XX.LE.13000)THEN
   N = 14
   DO IROW = 1,N
      READ(17,*) X(IROW),FX(IROW,1)
   ENDDO
ELSEIF(XX.GT.13000.AND.XX.LE.15000)THEN
```

```
         N = 14
         DO IROW = 1,N
            READ(18,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.15000.AND.XX.LE.17000)THEN
         N = 14
         DO IROW = 1,N
            READ(19,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.17000.AND.XX.LE.19000)THEN
         N = 14
         DO IROW = 1,N
            READ(20,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.19000.AND.XX.LE.21000)THEN
         N = 14
         DO IROW = 1,N
            READ(21,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.21000.AND.XX.LE.23000)THEN
         N = 14
         DO IROW = 1,N
            READ(22,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.23000.AND.XX.LE.25000)THEN
         N = 14
         DO IROW = 1,N
            READ(23,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.25000.AND.XX.LE.27000)THEN
         N = 14
         DO IROW = 1,N
            READ(24,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.27000.AND.XX.LE.28000)THEN
         N = 14
         DO IROW = 1,N
            READ(25,*) X(IROW),FX(IROW,1)
         ENDDO
      ELSEIF(XX.GT.28000) THEN
         FF = 6.24869E13
         GOTO 101
      ENDIF

C Compute divided-difference coefficients:
      M = N
      DO 20 ICOL = 2,N
```

```
         M = M - 1
         DO 30 IROW = 1,M
            FX(IROW,ICOL) =
     &      FX(IROW+1,ICOL-1)-FX(IROW,ICOL-1)
            FX(IROW,ICOL) =
     &      FX(IROW,ICOL)/(X(IROW+ICOL-1) - X(IROW))
 30      CONTINUE
 20   CONTINUE
C Compute desired F(X) at the given X value:
C      WRITE(5,*)'Input the temperature (K):'
C      READ(5,*) XX
      FF = FX(1,1)
      FAC = 1.
      DO 40 I = 2, N
         FAC = FAC*(XX-X(I-1))
         FF  = FF + FX(1,I)*FAC
 40   CONTINUE
C 101  WRITE(2,100) XX, FF,FF*1.1126535E-10
 101  ZAP = 1
C convert from (stat mho)/cm to mho/m by time 1.126535E-10
      SIG_P = FF*1.1126535E-10

 100  FORMAT('Temp =',E10.4,'is', E16.7,'or',E16.7)

      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(25)
      RETURN
      END


C     Function to place contour labels
      FUNCTION ZCONT(R)
      INCLUDE 'common.f'
      ZCONT = ZBOT + (ZTOP-ZBOT)*( (R-RLEFT)/RRIGHT )**2
      RETURN
      END
C---------------------------------------------------------------
C
C     Model of cathode sheath
C     Provides sheath voltage and heat flux
C     as functions of cathode temperature and current density
C
C     Inputs:
C     T (K): Cathode surface temperature
C     J (A/cm^2): Current density
C
C     Outputs:
C     VC (V): Cathode sheath voltage
C     Q (W/cm^2): Heat flux to cathode surface
C
      SUBROUTINE CMODEL( T, J, VC, QQ )
C      INCLUDE 'common_declare.f'
C      IMPLICIT REAL( J )

      SAVE

      LOGICAL FIRST /.TRUE./, DONE

C     Note: Following three lines must be consistent with each other!
      PARAMETER( NT_SH=301, NJ=2001 )
      PARAMETER( TLOW=2500, THIGH=4000, TSTEP=5 )
      PARAMETER( JLOW=0, JHIGH=10000, JSTEP=5 )

      DIMENSION VCDAT( NT_SH,NJ ), QDAT( NT_SH,NJ ), TVALS( NT_SH ), JVALS( NJ )
      DIMENSION IJMIN( NT_SH ), IJMAX( NT_SH )

C      write(*,*) ' CMODEL called with T=',T,' J=',J

      IF( FIRST )  THEN
C     First time called: Read the data files
         OPEN( 8, FILE='Vc.dat', STATUS='UNKNOWN' )
         OPEN( 9, FILE='q.dat', STATUS='UNKNOWN' )
         READ(8,*) NR, NC
C         WRITE(*,*) 'In file Vc.dat, got ',NR, ' rows -- should be ', NT_SH
         READ(9,*) NR, NC
C         WRITE(*,*) 'In file q.dat, got ',NR, ' rows -- should be ', NT_SH

         DO IT=1, NT_SH
C            write(*,*)'Reading Vc, line ', IT
```

```
            READ(8,*) ( VCDAT(IT,IJ), IJ=1, NJ )
C            write(*,*)'Reading q, line ', IT
            READ(9,*) ( QDAT(IT,IJ), IJ=1, NJ )
         ENDDO

C         WRITE(*,*) 'Finished reading Vc, Q values from data files'

C     Set the temperature and current density values
         DO IT=1, NT_SH
            TVALS(IT) = TLOW + (IT-1) * TSTEP
         ENDDO
         DO IJ=1, NJ
            JVALS(IJ) = JLOW + (IJ-1) * JSTEP
         ENDDO
C
C     For each temperature value, find region of convergence of model
C
C     Outside region of convergence: Extrapolate
C     Sheath voltage: Increase rapidly for too-high values of J
C     Heat flux:
C       Increase rapidly for too-high values of J
C       Decrease rapidly for too-low values of J
C
C     IJMIN, IJMAX: For each T value, the smallest and largest values
C     of IJ for which QDAT is nonzero
C
         DO IT=1, NT_SH
            DO IJ=1, NJ
               IF( QDAT(IT,IJ) .NE. 0 )  THEN
                  IJMAX(IT) = IJ
               ENDIF
            ENDDO
            DO IJ=NJ,1,-1
               IF( QDAT(IT,IJ) .NE. 0 )  THEN
                  IJMIN(IT) = IJ
               ENDIF
            ENDDO
         ENDDO
C
C         write(*,*) 'Finished finding limits of convergence'
         VCSLOPE = 10
         QSLOPE = 100
C
         FIRST = .FALSE.
      ENDIF
C
C     Check values -- for debugging
```

```
         DONE = .FALSE.
c         DO WHILE( .NOT. DONE )
         DO WHILE( .false. )
            WRITE(*,200)
 200        FORMAT( 'Enter IT, IJ (0 0 to quit): ', $ )
            READ(*,*) IT, IJ
            IF( IT .EQ. 0 ) THEN
               DONE = .TRUE.
            ELSE
               WRITE(*,*) 'Vc(',IT,IJ,') = ', VCDAT(IT,IJ)
               WRITE(*,*) 'q(',IT,IJ,') = ', QDAT(IT,IJ)
            ENDIF
         ENDDO

C     Check values -- for debugging
         DONE = .FALSE.
c         DO WHILE( .NOT. DONE )
         DO WHILE( .false. )
            WRITE(*,300)
 300        FORMAT( 'Enter IT (0 to quit): ', $ )
            READ(*,*) IT
            IF( IT .EQ. 0 ) THEN
               DONE = .TRUE.
            ELSE
               WRITE(*,*) 'IJMIN(',IT,') = ', IJMIN(IT)
               WRITE(*,*) 'IJMAX(',IT,') = ', IJMAX(IT)
            ENDIF
         ENDDO
C
C     Find location of input T, J in data files
C
C     XT: continuous, ranges from 0 to just below NT_SH-1
C     IT1, IT2: Integer values below and above XT
C     PT: continuous, ranges from 0 to 1: how far across T step we are
      XT = MAX( 0.0, MIN( ( T - TLOW ) / TSTEP, NT_SH-1.01 ) )
      IT1 = INT(XT) + 1
      IT2 = IT1 + 1
      PT = XT - INT(XT)
c      WRITE(*,*) 'Temperature ', T, ' is between IT = ', IT1, IT2
c      WRITE(*,*) 'Fraction ',PT, ' across temperature increment'

      XJ = MAX( 0.0, MIN( ( REAL(J) - JLOW ) / JSTEP, NJ-1.01 ) )
      IJ1 = INT(XJ) + 1
      IJ2 = IJ1 + 1
      PJ = XJ - INT(XJ)
c      WRITE(*,*) 'Current density ', J, ' is between IJ = ', IJ1, IJ2
c      WRITE(*,*) 'Fraction ',PJ, ' across current density increment'
```

```
c      write(*,*) 'IJMAX of ',IT1,' is ', IJMAX(IT1)

      IF( IJ2 .GT. IJMAX(IT1) )  THEN
C     Current too high for model -- extrapolate up
c         write(*,*) 'IT1: Current too high -- ijmax is ',IJMAX(IT1)
         IF( IJMAX(IT1) .EQ. 0 )  THEN
C     Model not converged at all for this T
c            WRITE(*,*) 'Model not converged at all for this T'
c            write(*,*) 'vcslope ',VCSLOPE,' qslope ',QSLOPE
            VC1 = VCSLOPE*J
            Q1 = QSLOPE*J
         ELSE
            JMAX = JVALS(IJMAX(IT1))
c            write(*,*) 'JMAX is ',JMAX
            VC1 = VCDAT(IT1,IJMAX(IT1)) + VCSLOPE*(J-JMAX)
            Q1 = QDAT(IT1,IJMAX(IT1)) + QSLOPE*(J-JMAX)
         ENDIF
      ELSEIF( IJ1 .LT. IJMIN(IT1) )  THEN
C     Current too low for model -- extrapolate down
c         write(*,*) 'IT1: Current too low -- ijmin is ',IJMIN(IT1)
         JMIN = JVALS(IJMIN(IT1))
c         write(*,*) 'JMIN is ',JMIN
         VC1 = 0
         Q1 = QDAT(IT1,IJMIN(IT1)) + QSLOPE*(J-JMIN)
      ELSE
C     Model converged here -- interpolate model data
c         write(*,*) 'IT1: Model is converged'
         VC1 = VCDAT(IT1,IJ1)*(1-PJ) + VCDAT(IT1,IJ2)*PJ
         Q1 = QDAT(IT1,IJ1)*(1-PJ) + QDAT(IT1,IJ2)*PJ
      ENDIF
c      write(*,*) 'Got VC1 = ',VC1,' Q1 = ',Q1

      IF( IJ2 .GT. IJMAX(IT2) )  THEN
C     Current too high for model -- extrapolate up
         IF( IJMAX(IT2) .EQ. 0 )  THEN
C     Model not converged at all for this T
c            WRITE(*,*) 'Model not converged at all for this T'
            VC2 = VCSLOPE*J
            Q1 = QSLOPE*J
         ELSE
            JMAX = JVALS(IJMAX(IT2))
            VC2 = VCDAT(IT2,IJMAX(IT2)) + VCSLOPE*(J-JMAX)
            Q2 = QDAT(IT2,IJMAX(IT2)) + QSLOPE*(J-JMAX)
         ENDIF
      ELSEIF( IJ1 .LT. IJMIN(IT2) )  THEN
C     Current too low for model -- extrapolate down
```

```
         JMIN = JVALS(IJMIN(IT2))
         VC2 = 0
         Q2 = QDAT(IT2,IJMIN(IT2)) + QSLOPE*(J-JMIN)
      ELSE
C     Model converged here -- interpolate model data
         VC2 = VCDAT(IT2,IJ1)*(1-PJ) + VCDAT(IT2,IJ2)*PJ
         Q2 = QDAT(IT2,IJ1)*(1-PJ) + QDAT(IT2,IJ2)*PJ
      ENDIF

C     Now interpolate in the T direction
      VC = VC1*(1-PT) + VC2*PT
      QQ = Q1*(1-PT) + Q2*PT

      RETURN

      END
```