VOLUME AVERAGED MODELING OF HIGH DENSITY DISCHARGES

by

Kedar Patel

Memorandum No. UCB/ERL M98/28

12 May 1998

VOLUME AVERAGED MODELING OF HIGH DENSITY DISCHARGES

by

Kedar Patel

Memorandum No. UCB/ERL M98/28

12 May 1998

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Volume Averaged Modeling of High Density Discharges

by

Kedar Kantilal Patel

Master of Science in Engineering-Electrical Engineering

University of California at Berkeley

A self-consistent spatially averaged model of high-density oxygen and boron trifluoride discharges has been developed. We determine the positive ion, negative ion, and electron densities, the ground state and metastable densities, and the electron temperature, as a function of the control parameters: gas pressure, gas flow rate, input power and reactor geometry. A simplified surface model is used. The wall recombination coefficient for oxygen and the wall sticking coefficient for boron trifluoride (BF_3) are the single adjustable parameters used to model the surface chemistry on the aluminum walls of the RF inductive source used in the Eaton ULE2 ion implanter. Complete wall recombination of O atoms is found to give the best match to the experimental data for oxygen, whereas a sticking coefficient of 0.62 for all neutral species in a BF_3 discharge was found to best match the experimental data. Density weighted parameters are used to incorporate multiple ion species into the 1-D single ion species model developed in earlier studies. The model results are in good agreement with experimental data for the Eaton source.

To my dearest family: Mummyji, Pappaji, Nikhil, Bhabhi, Punita, and Didi

Contents

Acknowledgments

I would like to thank Professor Michael Lieberman for being my advisor and Professor Charles Birdsall for being the second reader. I am deeply indebted to Prof. Lieberman for being a pillar of support throughout my stay at Cal. His ability to understand and solve my convoluted problems and his words of encouragement made my research very straightforward and enjoyable. I have known Mike as one of the most patient and even-tempered persons I have had the pleasure of knowing. I especially would like to thank him for letting me barge into his office to ask questions at any hour of the day. I am going to miss all the coffee runs in the morning and afternoon with him. I was truly fortunate to have him as a teacher at Cal.

I would like to thank Mike Graf of Eaton Corporation for providing the experimental data from the Eaton ULE2 ion source. I would also like to thank Jon Gudmundsson, a past member of the plasma lab, for helping me settle into the laboratory and getting acquainted with the transformer coupled plasma source (TCPS). I can never forget the time he smoked the entire lab while making measurements on the TCPS. I am also grateful to Jon for diligently computing the rate constants for reactions involving fluorine in the boron trifluoride model. I thank Thomas Philip, a friend and colleague, for calculating various rate constants and help with the Matlab code. I would like to thank Igor Kouznetsov and Jon Gudmundsson for helpful discussions, Gianluca Gregori, Yuri Glukhoy, Sunder Kumar Iyer (SKI), Kaustav Banerjee, and David Baca.

My stay at Cal had its good and bad times. I still remember my first day of classes at Cal as a undergraduate in Engineering Physics. My biggest fear was whether or not I belong at Cal. It took a semester for Cal to become my new home away from home. Dorm life in Freeborn Hall, the eggplant parmesan of Unit 1 dining commons, movie nights at Wheeler, the amazing maze called Dwinelle Hall, the roller-coaster ride of Evans Hall elevators, Cafe Strada, Telegraph Avenue, midnight runs to south side food court, the weirdos at Sproul (the preacher, Rick Starr, the hate-man), the drummer, the south american folk music band, sleepless nights in Doe stacks have all contributed in making my stay at Cal memorable.

This work was supported by Eaton Corporation contract ESEO-M2708.

Finally, I would like to thank my family for their love and support. I would not be here without them, especially my late sister, Prasanna Patel (Didi).

Chapter 1

Oxygen Discharge

1.1 Introduction

In the electronics industry, plasma processing holds a central place. It has been widely used for anisotropic etching in fabrication of microelectronic chips, implantation of various dopants in semiconductors, deposition of various compounds for surface passivation and insulation, ashing of photoresist [1]. The need for larger wafer sizes, with the ever increasing semiconductor technology, has increased the demand for better, higher density plasma reactors. Inductively coupled plasma sources have emerged as a commercial plasma source in the recent years due to their simple design, good uniformity, and wide operating pressure (0.1-100 mTorr) and power (200-2000 W) regimes. The optimum regime is chosen based on the type of discharge species desired. For instance, low pressures (<50 mTorr) and high powers (>500W) yield a higher concentration of positive ion species, and high pressures (>50 mTorr) and low power (<500 W) yield a higher concentration of negative ion species.

Most modern day plasma reactors are designed empirically; we lack sufficient understanding about the plasma behavior to eliminate a costly trial-and-error equipment development approach [2]. Although there are some commercially available codes to model plasmas, they are computationally expensive and thus the need for a simple, definitive code still exists. We present a simple

volume averaged steady state model of plasma discharges that is implemented in the MATLAB programming environment. The model presented here is described by Lieberman and Gottscho [3] for the noble gas family and by Lee et al. [4, 5] for molecular gases. A global model for high pressure electronegative discharges was presented by Y. Lee et al. [6]. A global model for pulsed power discharges at low pressures was introduced by Ashida and Lieberman [7]. The model presented in this paper is formulated for cylindrical inductive discharges and is a refinement of the work done by Lee et al. [4, 5].

A generalized power balance equation is used which separately describes the volume and surface energy losses. The volume energy loss channels due to electron-neutral collisions include rotational, vibrational, and electronic excitations, ionization, and dissociation. Particle balance equations are derived from simple mass conservation laws. Rate constants for various reactions are obtained by integrating their respective collision cross-sections over an assumed maxwellian distribution. Density weighted quantities are used to determine the mean free paths in ion-neutral interactions. In light of comparisons made with experimental data from the Eaton ULE2 source, we estimate the value of the wall recombination coefficient for O atoms on the aluminum wall surface to be unity. The complete set of equations is solved self consistently using a quasi Newton-Raphson algorithm called the Levenberg-Marquardt algorithm [8]. The code was implemented in the commercially available MATLAB programming language. The effects of variations in the control parameters, such as the reactor pressure, power, flow-rate and reactor geometry, on the plasma parameters, such as various species concentrations, energy losses, and electron temperature, are studied.

1.2 Model Formulation

A description of the processes occurring within the reactor, under the steady state assumption, is obtained by relating the conservation of mass and energy to the generation and annihilation of chemical species within the reactor volume. The code input consists of the reactor geometry, the process conditions, and the starting estimate of the solution. The particle balance equations and the reaction coefficients are also required as input parameters. A set of non-linear equations is compiled incorporating the particle balance equations and the power balance equation. The Levenberg-Marquardt algorithm [8] is used initially to solve the non-linear set of equations and Gauss-Newton algorithm [8] is used to do the subsequent stepping in parameter space. As in solving any non-linear set of equations, reasonable starting estimates are required. Starting estimates for concentration of various species and the electron temperature often can only be derived from intuition and a trial-and-error approach.

1.2.1 Assumptions

The model presented in this paper is based on a cylindrical reactor geometry, with L=20 cm and R=10 cm, with multipole magnetic confinement. The details about the magnetic confinement are discussed in chapter 2. The assumptions made in the formulation of the global model are as follows.

1. A maxwellian electron energy distribution function (EEDF) is assumed.
2. The discharge gas and ion temperatures are assumed to be constant, irrespective of the discharge conditions. Both the ion and neutral temperatures are assumed to be 600 K.
3. All densities n are assumed to be volume averaged:

$$n = \frac{1}{\pi R^2 L}\left(2\pi \int_0^R r dr \int_0^L n(r,z) dz\right) \tag{1.1}$$

Thus the model does not describe the spatial distribution of various plasma species.

4. Steady state is assumed.

5. All the particles are assumed to be created uniformly throughout the volume of the discharge and are assumed to have a isotropic distribution of velocities.

6. The density profiles of all species are assumed to be uniform in the entire discharge except possibly for a sharp decrease near the walls at the plasma-sheath edge. The negative ion density is assumed to fall to zero at the plasma-sheath edge. The quasi-neutrality condition

$$n_e = n_+ - n_- \tag{1.2}$$

dictates that at the sheath edge $n_{es} = n_{+s}$.

7. The sheath is assumed to be negligibly thin compared to the size of the plasma discharge.

8. The electron temperature is assumed to be uniform in the entire discharge. This limits the model range to pressures below 100 mTorr [6].

9. We neglect the energy loss processes involving electron collisions with charged species because the charged particle densities are small compared to the neutral densities. We account for electron energy loss collisions only with ground state neutral species.

10. Three body reactions are neglected since three body collisions may be important only for pressures greater than 100 mTorr.

11. Threshold processes such as excitation, dissociation and ionization are assumed to be electron induced only.

12. The value of the wall recombination coefficient γ_{rec} is not available for O atoms on an alu-

minum surface. It is a quantity whose value can range from $\gamma_{rec} = 0$ (no surface recombination) to $\gamma_{rec} = 1$ (complete surface recombination). In our model, we found that complete surface recombination of O atoms at the aluminum surface of the reactor provided the best fit to experimental data for the Eaton ULE2 ion source.

13. Only singly ionized species are considered.

1.2.2 Plasma Chemistry

Table 1.1 summarizes the oxygen reaction set. Only the most dominant reactions and species are considered in this model. Rate coefficients for the electron impact collisions were obtained by integration of the collision cross-sections over an assumed maxwellian distribution

$$k_i = \langle \sigma v \rangle = 4\pi \int_0^\infty \sigma v^3 f(v) dv \tag{1.3}$$

where

$$f(v) = \left(\frac{m_e}{2\pi k_B T_e}\right)^{3/2} exp\left(-\frac{m_e v^2}{2k_B T_e}\right) \tag{1.4}$$

is the maxwellian speed distribution. The rate constants were then fit to the Arrhenius form,

$$k_i = A T_e^B exp\left(-\frac{C}{T_e}\right) \tag{1.5}$$

in the range of 1 to 8 eV. The diffusional losses of O and O* to the reactor walls are estimated by [12]

$$k_g = \left[\frac{\Lambda_o^2}{D_g} + \frac{2V(2-\gamma)}{A v_g \gamma}\right]^{-1} \tag{1.6}$$

where, D_g is the neutral diffusion coefficient given by,

$$D_g = \frac{eT_g\lambda_g}{v_g m_g} \tag{1.7}$$

v_g is the mean neutral speed given by,

$$v_g = \left(\frac{8eT_g}{\pi m_g}\right)^{1/2} \tag{1.8}$$

γ is the sticking coefficient, and V and A are the volume and wall surface area of the reactor respectively. and λ_g is the mean free path. The effective diffusion length of each of the neutral species is given by [5]

$$\Lambda_0 = \left[\left(\frac{\pi}{L}\right)^2 + \left(\frac{2.405}{R}\right)^2\right]^{-1/2} \tag{1.9}$$

The wall recombination coefficient (γ_{rec}) has not been studied extensively. However, we know that it is strongly dependent on the wall material. Booth and Sadeghi [12] report a experimental value of ~ 0.5 on stainless steel. In a study done by Greaves and Linnett [13], the recombination coefficient γ for oxygen atoms on silica was determined for the temperature range 20°C to 600°C. They showed that recombination coefficient increased from $\gamma_{rec} = 1.6\times10^{-4}$ at 20°C to $\gamma_{rec} = 1.4\times10^{-2}$ at 600°C. If we assume that the interior surface of the reactor is passivated with a monolayer of oxygen atoms, the wall recombination would be further reduced due to the low physisorption surface coverage [5]. In most ICP sources, however, the reactor surface is made of aluminum. With no available scientific data for aluminum. we find that $\gamma_{rec} = 1$ is a best fit of our model to data from the Eaton ULE2 source.

In addition to recombination of O atoms at the wall, other surface reaction take place such as the neutralization of positive ions as they impinge upon the surface; the neutralized species is then recycled back into the bulk discharge. Reactions 19 and 20 represent such process. The Bohm velocity of ions at the sheath edge is given by [14]

$$U_B = \left[\frac{eT_i}{m_i}\left(\frac{1+\alpha}{1/\gamma+\alpha}\right)\right]^{1/2} \tag{1.10}$$

where T_e and m_i are the electron temperature and ion mass respectively, $\alpha = n_-/n_e$ is a measure of the electronegativity of the discharge and

$$\gamma \equiv \left(\sum_i n_i \frac{T_e}{T_i}\right) / \left(\sum_i n_i\right) \tag{1.11}$$

is an ion concentration weighted quantity. For small α, (1.10) reduces to the more familiar form

$$U_B = \sqrt{\frac{eT_e}{m_i}}. \tag{1.12}$$

1.2.3 Particle Balance

The rate of generation and annihilation of each species is determined by the plasma chemistry. This includes electron impact, ion-ion, and ion-neutral collisions. For each neutral and charged species, a particle balance equation is generated which accounts for all the dominant generation and destruction processes and diffusion losses to the wall. Steady state particle balance equations are written for O_2, O, O*, O_2^+, O^+, and O^-. The general form of the particle balance equations is

Flow In + Rate of Generation = Rate of Annihilation + Pumping Loss

The particle balance equations for neutral species are listed below.

$$\frac{dn_{O_2}}{dt} = \frac{Q}{V} + k_5 n_{O_2^+} n_{O^-} + k_9 n_O n_{O^-} + k_{20} n_{O_2^+} + \frac{1}{2} k_{21} n_O - k_{12} n_{O_2} n_{O^+} - \cdots$$
$$(k_1 + k_3 + k_8 + k_{10} + k_{11} + k_{13}) n_{O_2} n_e - k_p n_{O_2} \tag{1.13}$$

$$\frac{dn_O}{dt} = 2k_2 n_{O_2^+} n_e + (k_3 + 2k_8 + k_{11} + k_{13}) n_{O_2} n_e + k_5 n_{O_2^+} n_{O^-} + 2k_6 n_{O^-} n_{O^+} + \cdots$$
$$k_7 n_e n_{O^-} + k_{12} n_{O_2} n_{O^+} + k_{15} n_{O_2} n_{O^*} + k_{16} n_O n_{O^*} + k_{18} n_{O^*} + k_{19} n_{O^+} - \cdots \tag{1.14}$$
$$(k_4 + k_{14}) n_O n_e - k_9 n_O n_{O^-} - k_{21} n_O - k_p n_O$$

$$\frac{dn_{O^*}}{dt} = k_{13} n_{O_2} n_e + k_{14} n_O n_e - k_{15} n_{O_2} n_{O^*} - k_{16} n_O n_{O^*} - k_{17} n_e n_{O^*} - k_{18} n_{O^*} - k_p n_{O^*} \tag{1.15}$$

The flow through the reactor has been characterized by a nominal residence time τ. The residence time, and hence the pumping rate $k_p = 1/\tau$, has been deduced from the flow-rate, pressure, and the reactor volume.

$$\tau = \frac{PV}{Q} \tag{1.16}$$

Thus the pumping rate is determined from the two control parameters, pressure and flow-rate, for a given reactor volume. Experimentally the pumping rate can be manipulated by adjusting the exhaust throttle valve. Varying the residence time helps us investigate purely chemical effects such as a changing gas composition. At high gas flow values (i.e short residence times), neutral species created inside the reactor are flushed out of the system and the plasma chemistry can drastically be altered. In an oxygen discharge, for example, at high flow rates neutral species like O are flushed out of the system. This causes the O/O_2 and O^+/O_2^+ concentration ratios to decrease.

The reactor pressure is determined by the total neutral concentration (n_g) of the plasma. The reac-

tor pressure is given by

$$P = n_g k_B T_g \tag{1.17}$$

where $n_g = n_{O_2} + n_O + n_{O^*}$ and $T_g = 600\text{K}$ is the neutral temperature. The reactor pressure can be different for no-plasma and plasma conditions. The difference is considerable in case of a highly dissociated plasma where the concentration of neutral particles is increased as a result of dissociation. Although the degree of ionization is typically less that 1%, the degree of dissociation can be close to 100%. However, the gas temperature typically increases along with the degree of dissociation, yielding a reduced neutral gas density.

For our model, we consider the production of ions only through electron-neutral collisions. So O_2^+ is created from O_2 and O^+ is created from O, O*, and O_2(small). The particle balance equations for the positive ions are written as

$$\frac{dn_{O_2^+}}{dt} = k_1 n_{O_2} n_e + k_{12} n_{O_2} n_{O^+} - k_2 n_{O_2^+} n_e - k_5 n_{O_2^+} n_{O^-} - k_{20} n_{O_2^+} \tag{1.18}$$

$$\frac{dn_{O^+}}{dt} = k_4 n_O n_e + k_{10} n_{O_2} n_e + k_{11} n_{O_2} n_e + k_{17} n_{O^*} n_e - k_6 n_{O^-} n_{O^+} - k_{19} n_{O^+} - k_{12} n_{O_2} n_{O^+} \tag{1.19}$$

Stoffels et al. [15] have shown that roughly 90% of the negative ions present in the discharge are O^- ions. Thus in our model formulation we consider O^- as the only negative ion species in the plasma and further we assume that no negative ions are lost as reactor exhaust. The justification for this assumption is that the negative ions are essentially trapped in the bulk plasma because of the high positive potential of the plasma with respect to the chamber walls. The particle balance for O^- is written as

$$\frac{dn_{O^-}}{dt} = k_3 n_{O_2} n_e + k_{10} n_{O_2} n_e - k_5 n_{O_2^+} n_{O^-} - k_6 n_{O^-} n_{O^+} - k_7 n_e n_{O^-} - k_9 n_O n_{O^-} \tag{1.20}$$

Since the bulk of the plasma is essentially neutral, we have

$$\frac{dn_e}{dt} = \frac{dn_{O^+}}{dt} + \frac{dn_{O_2^+}}{dt} - \frac{dn_{O^-}}{dt} \tag{1.21}$$

In each of the aforementioned particle balance equations, we make the steady state assumption so that the time derivatives for all species are zero, i.e.

$$\frac{d}{dt} = 0 \tag{1.22}$$

1.2.4 Power Balance

The power balance equation is obtained by applying conservation of energy to the plasma under the assumption that all input power is absorbed by the plasma. The energy is dissipated in various collisional processes including excitation processes. The RF power is deposited into the plasma primarily via ohmic heating. The total power balance has the general form

$$P_{abs} = P_v + P_s \tag{1.23}$$

where P_{abs} is the power absorbed by the plasma, P_v is the power lost to due to electronic collisions in the volume, and P_s is the power lost due to electron and ion flux to the reactor walls. The electron collisional energy loss per electron-ion pair produced by ionization of the i-th target neutral species is given by

$$E_{CL,i} = \frac{1}{K_{i,iz}} \sum_{j=1}^{N_{CL}} K_{i,j} E_{i,j} \tag{1.24}$$

where $K_{i,iz}$ is the ionization rate constant of i-th species, $K_{i,j}$ is the rate coefficient of j-th pro-

cess of i-th species, $E_{i,j}$ is the activation energy of j-th process of i-th species, and N_{CL} is the total number of volume energy loss channels due to electron-neutral collisions which includes rotational, vibrational, and electronic excitations, ionization, dissociation, and elastic collisions. The electron targets include O_2, O, O*, O_2^+, O^+, and O^-. Due to unavailability of cross-sectional data for all processes, we only consider O_2 and O neutral species. Table 1.2 and Table 1.3 summarize the rate coefficients for various processes. The rate coefficients in Table 1.2 were calculated using the cross sections provided by Phelps [16, 17] and Eliasson and Kogelschatz [11]. Thus the volume power loss for an electron colliding with i-th electron target in j-th process is given by

$$P_v = en_e V \sum_i n_i \sum_{j=1}^{N_{CL}} K_{i,j} E_{i,j} \tag{1.25}$$

where V is the reactor volume, n_e is the electron density and n_i is the concentration of i-th electron target.

As ions accelerate through the sheath, they acquire an energy

$$E_i = \phi + \frac{1}{2} T_e \tag{1.26}$$

where ϕ is the potential of the plasma-sheath edge with respect to the wall and T_e is the electron temperature in volts. This energy is derived from the bulk plasma and therefore it appears as a energy loss mechanism for the plasma, because it is lost to the reactor walls upon collision. The typical ion energy lost to the wall is about 5 to 8 T_e. The electrons also lose energy to the wall under a similar mechanism. The average energy lost per electron to the walls is given by

$$E_e \approx 2T_e \tag{1.27}$$

Assuming no net current flows to the reactor wall, the plasma potential can be estimated by balancing of the total positive and negative flux to the wall, i.e.

$$\Gamma_e + \Gamma_- = \Gamma_+ \tag{1.28}$$

where Γ_e is the electron flux, Γ_- is the total negative ion flux, and Γ_+ is the total positive ion flux and are given by

$$\Gamma_e = \frac{1}{4} n_{es} v_e \exp\left(\frac{\phi}{T_e}\right) \tag{1.29}$$

$$\Gamma_- = \frac{1}{4} n_{-s} v_- \exp\left(\frac{\phi}{T_e}\right) \tag{1.30}$$

$$\Gamma_+ = n_{+s} U_B \tag{1.31}$$

The quantities n_{es}, n_{+s}, and n_{-s} refer to the sheath edge concentration of electrons, positive ions and negative ions. U_B is the Bohm velocity given by (1.10) or (1.12), and $v_e = (8eT_e/\pi m_e)^{1/2}$ and $v_- = (8eT_-/\pi m_-)^{1/2}$ are the electron and negative ion mean kinetic velocities respectively. In order to determine the plasma potential, we neglect the negative ion flux to the wall. This assumption is justified because (i) in the regime of interest the negative ion concentration is low, and (ii) the negative ions are essentially trapped in the bulk plasma because of the high positive potential of the plasma with respect to the chamber walls. Thus solving (1.29) and (1.31) for ϕ, using (1.12) and $n_{es} = n_{+s}$ at the sheath edge, we get,

$$\phi = \frac{T_e}{2} \ln\left(\frac{m_i}{2\pi m_e}\right). \tag{1.32}$$

The surface power loss is then given by

$$P_s = eA_{eff}\sum_i U_{B,i} n_i E_T \tag{1.33}$$

where

$$A_{eff} = 2\pi R^2 h_L + 2\pi R L h_R \tag{1.34}$$

is the effective interior area of the reactor, $U_{B,i}$ is the Bohm velocity of i-th ion given by (1.10) or (1.12), n_i is the i-th ion concentration, and $E_T = E_i + E_e$ is the total energy lost to the wall per electron-ion pair lost to the wall. The factors h_L and h_R scale the bulk and sheath edge densities in axial and radial directions respectively. Tsendin [19] and Lichtenberg et al. [20] have showed that it is possible to model plasmas as consisting of an electronegative core, with electrons and negative ions in Boltzmann equilibrium, which is matched to the electropositive edge regions. Lee et al. [4, 5] showed that in a low pressure-high power regime, the negative ion concentration is low and thus a slightly modified electropositive model still can be used. We have

$$h_L = \frac{0.86}{1+\alpha}\left(3 + \frac{L}{2\lambda_i}\right)^{-1/2} \tag{1.35}$$

and

$$h_R = \frac{0.8}{1+\alpha}\left(4 + \frac{R}{\lambda_i}\right)^{-1/2}. \tag{1.36}$$

This factors are very similar to those derived by Godyak [21] for an electropositive discharge. However, the scaling factors h_L and h_R factors are modified such that the ion-neutral mean free path λ_i is a density weighted quantity and a pre-factor of $1/(1+\alpha)$ is added to account for the effect of negative ions. The ion-neutral mean free path is given by

$$\frac{1}{\lambda_i} = \sum_{i=1}^{N_i} n_{g,i}\sigma_{g,i} \tag{1.37}$$

where $n_{g,i}$ refers to the parent neutral species of the i-th ion, $\sigma_{g,i}$ is the scattering cross-section of the electron-neutral collision that produces the i-th ion, and N_i is the total number of positive ion species.

The modified effective area and the sheath density scaling factors that are used for magnetic confinement are discussed and presented in chapter 2.

1.3 Results and Discussion

The reactor pumping was held fixed to give 1 mTorr reactor pressure at a gas flow rate of 1.5 sccm in the absence of a plasma. Thus an increase in the gas flow rate corresponds to an increase in reactor pressure.

1.3.1 Dependence on Input Power

The effect of varying the input power to the plasma can be observed in figures 1-14. Figure 1 shows the neutral concentration as a function of power absorbed by the plasma. Since the reactor pressure (1.17) is determined by the total neutral concentration, figure 1 shows the variation in reactor pressure as a function of input power. It is clear that as power is increased the total neutral concentration increases due dissociation of O_2 molecules into O and O*. Thus as input power is increased, O_2 concentration decreases (see figures 2 and 4), and an increase in O and O* concentrations is observed (see figures 3 and 5). The neutral concentration at zero power gives the no-

plasma reactor pressure as determined by the flow rate and pumping.

Figure 6 shows a linear variation in electron concentration as a function of input power. Figure 7 shows that the electron temperature weakly depends on input power; it decreases slightly with increasing power. The negative ion density is insignificant in the low pressure regime as shown in Figure 8. It is almost four orders of magnitude less than the O_2 and O concentrations. The plasma is very electropositive in the low pressure regime. Figure 9 shows the decrease in electronegativity of the plasma as the input power is increased. Since the bulk of the plasma is quasi neutral (1.2) and the negative ion density is negligible, increasing the input power yields higher concentration of positive ion species in the plasma as shown in Figure 10.

The fractional ionization of the plasma is given by

$$\chi_{iz} = \frac{n_+}{n_+ + n_g} \tag{1.38}$$

where n_+ is the total positive ion concentration and n_g is the total neutral concentration of the plasma. Figure 11 shows the variation in fractional ionization as a function of input power. For an input power of 2000 watts, the degree of ionization is only 0.1%. As mentioned earlier, the plasma can be highly dissociated (see figure 5) even though the degree of ionization is very small. Figures 12 and 13 show the concentrations of O_2^+ and O^+ respectively. It can be seen that O^+ is clearly the dominant positive ion species. Figure 14 shows the ratio of concentration of O^+ to O_2^+. As the reactor pressure is increased, O_2^+ is seen to become the dominant ion species. As shown in figure 14, the threshold power for co-dominance of both species increases as the reactor

pressure is increased. Since atomic oxygen is the dominant neutral species, it not surprising that the dominant ion species is O^+.

1.3.2 Dependence on Reactor Pressure

The effect of varying the reactor pressure can be observed in figures 15-28. Figure 15 shows the total neutral concentration as a function of pressure. Since the pumping is kept fixed at all times, the graph is a straight line with a slope 1.5 sccm/mTorr. The atomic oxygen concentration appears to increase linearly with pressure over a wider range at high input power than at low input power values (see figure 16). Figure 17 is a re-statement of figure 5. It shows that the degree of dissociation is reduced as the reactor pressure is increased. Figures 18 and 19 show that the metastable species is insignificant in the low pressure regime. The metastable concentration is proportional to electron concentration and since the electron density increases with power so does the O^* concentration. Figure 20 shows the decrease in electron temperature with increasing pressure. The variation of electron temperature with pressure is more pronounced than its variation with input power.

Figure 21 shows that the electron density shows only a slight variation with increasing pressure. It was shown earlier that the electron density was proportional to the input power. For low pressure, high density oxygen discharges the concentration of negative ions is negligible. However, as shown in figure 22, in low power and high pressure range, the negative ions begin to become more important. From the particle balance equation for O^-, we can see that the dominant loss mechanism for negative ions is neutralization with the positive ion species. Also the primary production mechanism of negative ions is the dissociative attachment reaction involving electrons

and O_2 molecules. Since negative ions are trapped in the bulk plasma due to the high potential barrier, their concentration is directly affected by the concentration of O_2. Due to decreased dissociation at lower power, the O_2 concentration is higher. Hence the concentration of O^- increases due to the higher dissociative attachment rate. Also the total positive ion concentration is lower at lower power (see figure 10) and hence there are fewer ions present to participate in the neutralization of negative ions. A combination of these effects results in net increase in the negative ion density, and thus the electronegativity α of the plasma steadily increases with pressure as shown in figure 23. Even though the value of α increases with pressure, the negative ions still make an insignificant contribution in the plasma chemistry.

In figure 24, we see that for a fixed power, the total positive ion density decreases as the reactor pressure is increased. In figure 25, the fractional ionization (1.38) is shown to decrease linearly with reactor pressure. As seen in figure 26, the O_2^+ concentration does not vary significantly and it increases slightly with pressure. However, figure 27 shows that the O^+ concentration decreases with pressure. The net result shown in figure 28 is that the ratio O^+/O_2^+ also decreases with pressure with O^+ still being the dominant species.

1.3.3 Comparison with Experimental Results

Figure 29 shows the comparison of model predictions with the experimental data for ion concentration ratios from the Eaton ULE2 RF driven inductively coupled plasma source. The results are in good agreement with the experimental data for both variations in input power and the reactor pressure for the choice of $\gamma_{rec} = 1$ i.e, complete recombination at the reactor wall surface. Fig-

ure 30 shows the comparison between the model and experimental data for O^+. The experimental O^+ ion concentration is deduced from the measured O^+ ion current by normalizing the current to a factor of 10^{17} in both cases of figure 30. Again there is good agreement between the experimental values and the model for variations in the input power. The experimental data shows that the O^+ ion concentration has a steeper fall off than that predicted by the model. However, the discrepancy between the two results is only about 15%.

1.3.4 Dependence on Wall Recombination

Figures 31-41 show the effects of the wall recombination coefficient (γ_{rec}) for O atoms. As mentioned previously, this quantity can range from $\gamma_{rec} = 0$ (no surface recombination) to $\gamma_{rec} = 1$ (complete surface recombination). Figure 31 shows that in absence of wall recombination the O atom density increases linearly with pressure, but as wall recombination is increased, the density increases less steeply. Figure 32 shows that the steep fall off in the O* metastable density occurs at a lower pressure as the wall recombination is increased. There is no significant change in the electron temperature for different values of recombination coefficient as can be seen in figure 33. The electron density has no strong dependence on γ_{rec}. The negative ion density seems to be strongly dependent on the wall recombination of O atoms. As seen in figure 35, the O^- concentration is extremely sensitive to the value of γ_{rec}. A plausible explanation for this can be (i) the increase in molecular oxygen concentration due to destruction of O atoms leads to a higher dissociative attachment rate, (ii) due to destruction of O atoms, the concentration of O^+ ions is reduced, and thus the destruction of O^- ions by positive-negative ion mutual neutralization is reduced. The value of α also increases due to the increase in O^- density as seen in figure 36. The total positive

ion concentration, shown in figure 37, is not greatly affected by the changes in wall recombination coefficient and the fractional ionization remains practically unchanged in figure 38. The O_2^+ ion concentration shows a significant variation with γ_{rec}. Again, this can be attributed to an increase in O_2 concentration due to recombination of O atoms at the reactor surface. The O^+ concentration, shown in figure 40, clearly falls much more rapidly for γ_{rec}= 1 than for smaller values of γ_{rec}. This is due to a reduction in O atom concentration, which is the primary source of O^+ ions. The combined effect of changes in O^+ and O_2^+ concentrations causes their ratio to fall for higher values of recombination coefficient as shown in figure 41.

Table 1.1 Oxygen Reaction Set[a]

Reaction	Rate Coefficient [m^3/s]	Ref.
1. $e + O_2 \rightarrow O_2^+ + 2e$	$k_1 = 9\times10^{-16} T_e^2 \exp(-12.6/T_e)$	[4]
2. $e + O_2^+ \rightarrow 2O$	$k_2 = 5.2\times10^{-15}/T_e$	[9]
3. $e + O_2 \rightarrow O + O^-$	$k_3 = 8.8\times10^{-17} \exp(-4.4/T_e)$	[10]
4. $e + O \rightarrow O^+ + 2e$	$k_4 = 9\times10^{-15} T_e^{0.7} \exp(-13.6/T_e)$	[4]
5. $O^- + O_2^+ \rightarrow O_2 + O$	$k_5 = 1.5\times10^{-13} (300/T_g)^{0.5}$	[4]
6. $O^- + O^+ \rightarrow 2O$	$k_6 = 2.5\times10^{-13} (300/T_g)^{0.5}$	[4]
7. $e + O^- \rightarrow O + 2e$	$k_7 = 2\times10^{-13} \exp(-5.5/T_e)$	[10]
8. $e + O_2 \rightarrow 2O + e$	$k_8 = 4.2\times10^{-15} \exp(-5.6/T_e)$	[4]
9. $O^- + O \rightarrow O_2 + e$	$k_9 = 3\times10^{-16}$	[10]
10. $e + O_2 \rightarrow O^- + O^+ + e$	$k_{10} = 7.1\times10^{-17} T_e^{0.5} \exp(-17/T_e)$	[10]
11. $e + O_2 \rightarrow O + O^+ + 2e$	$k_{11} = 5.3\times10^{-16} T_e^{0.9} \exp(-20/T_e)$	[10]
12. $O^+ + O_2 \rightarrow O + O_2^+$	$k_{12} = 2\times10^{-17} (300/T_g)^{0.5}$	[11]
13. $e + O_2 \rightarrow O + O^* + e$	$k_{13} = 5\times10^{-14} \exp(-8.4/T_e)$	[4]
14. $e + O \rightarrow O^* + e$	$k_{14} = 4.5\times10^{-15} \exp(-2.29/T_e)$	[4]
15. $O^* + O_2 \rightarrow O + O_2$	$k_{15} = 4.11\times10^{-17}$	[4]
16. $O^* + O \rightarrow 2O$	$k_{16} = 8.1\times10^{-18}$	[4]
17. $e + O^* \rightarrow O^+ + 2e$	$k_{17} = 9\times10^{-15} T_e^{0.7} \exp(-11.6/T_e)$	[4]
18. Wall Reaction $O^* \rightarrow \frac{1}{2}O_2$	$k_{18} = \left[\frac{\Lambda_o^2}{D_{O^*}} + \frac{2V(2-\gamma_{O^*})}{A v_{O^*} \gamma_{O^*}}\right]^{-1}$ $[s^{-1}]$	[4]
19. Wall Reaction $O^+ \rightarrow O$	$k_{19} = U_{B,O^+} \frac{A_{eff}}{V}$ $[s^{-1}]$	[5]
20. Wall Reaction $O_2^+ \rightarrow 2O$	$k_{20} = U_{B,O_2^+} \frac{A_{eff}}{V}$ $[s^{-1}]$	[5]
21. Wall Reaction $O \rightarrow \frac{1}{2}O_2$	$k_{21} = \left[\frac{\Lambda_o^2}{D_O} + \frac{2V(2-\gamma_O)}{A v_O \gamma_O}\right]^{-1}$ $[s^{-1}]$	[5]

a. T_e in Volts, T_g in Kelvins and O^* denotes the $O(^1D)$ state.

Table 1.2 Excitation Energy Loss Reaction Set for O_2

Reaction	Process	Activation Energy [V]	Rate Coefficient [m^3/s]	Ref.
1. $e + O_2 \rightarrow O_2(^1\Delta_g) + e$	Metastable Excitation	0.977	$k_1 = 1.37\times10^{-15}\exp(-2.14/T_e)$	[10][a]
2. $e + O_2 \rightarrow O_2(^1\Sigma_g) + e$	Metastable Excitation	1.627	$k_2 = 3.24\times10^{-16}\exp(-2.218/T_e)$	[10][a]
3. $e + O_2 \rightarrow O_2 + e$	Momentum Transfer	$3\frac{m_e}{m_i}T_e$	$k_3 = 1.64\times10^{-13}\exp(-4.749/T_e)$	[10][a]
4. $e + O_2 \rightarrow O_2^+ + e$	Ionization	12.06	$k_4 = 9\times10^{-16}T_e^2\exp(-12.6/T_e)$	[4]
5. $e + O_2 \rightarrow O_2(r) + e$	Rotational Excitation	0.02	$k_5 = 0$	-
6. $e + O_2 \rightarrow O_2(v{=}1) + e$	Vibrational Excitation	0.19	$k_6 = 2.8\times10^{-15}\exp(-3.72/T_e)$	[10][a]
7. $e + O_2 \rightarrow O_2(v{=}2) + e$	Vibrational Excitation	0.38	$k_7 = 1.28\times10^{-15}\exp(-3.67/T_e)$	[10][a]
8. $e + O_2 \rightarrow O_2(v{=}3) + e$	Vibrational Excitation	0.57	$k_8 = 7.81\times10^{-16}\exp(-3.83/T_e)$	[10][a]
9. $e + O_2 \rightarrow O_2(v{=}4) + e$	Vibrational Excitation	0.75	$k_9 = 4.8\times10^{-16}\exp(-4.33/T_e)$	[10][a]
10. $e + O_2 \rightarrow e + O_2^*$	Excitation[b]	4.5	$k_{10} = 1.07\times10^{-15}\exp(-3.43/T_e)$	[10][a]
11. $e + O_2 \rightarrow e + O_2^*$	Excitation[b]	6.0	$k_{11} = 3.73\times10^{-15}\exp(-4.9/T_e)$	[10][a]
12. $e + O_2 \rightarrow e + O_2^*$	Excitation[b]	8.4	$k_{12} = 3.91\times10^{-14}\exp(-8.29/T_e)$	[10][a]
13. $e + O_2 \rightarrow e + O_2^*$	Excitation[b]	10	$k_{13} = 3.92\times10^{-16}\exp(-11.48/T_e)$	[10][a]

a. Based on data of Phelps [16, 17], Eliasson and Kogelschatz [11]

b. O_2^* denotes excited ground state O_2 molecule.

Table 1.3 Excitation Energy Loss Reaction Set for O

Reaction	Process	Activation Energy [V]	Rate Coefficient [m^3/s]	Ref.
1. $e + O(^1P) \rightarrow O(^1D) + e$	Metastable Excitation	1.96	$k_1 = 4.47\times10^{-15}\exp(-2.29/T_e)$	[18]
2. $e + O(^1P) \rightarrow O(^1S) + e$	Metastable Excitation	4.18	$k_2 = 4.54\times10^{-15}\exp(-4.49/T_e)$	[18]
3. $e + O(^1P) \rightarrow O(^3P^0) + e$	Excitation	15.65	$k_3 = 4.54\times10^{-15}\exp(-17.34/T_e)$	[18]
4. $e + O(^1P) \rightarrow O(^5S^0) + e$	Excitation	9.14	$k_4 = 9.67\times10^{-16}\exp(-9.97/T_e)$	[18]
5. $e + O(^1P) \rightarrow O(^3S^0) + e$	Excitation	9.51	$k_5 = 9.67\times10^{-16}\exp(-9.75/T_e)$	[18]
6. $e + O(^1P) \rightarrow O^{h^*} + e$	Excitation	12.0	$k_6 = 4.31\times10^{-14}\exp(-18.6/T_e)$	[18]
7. $e + O \rightarrow O^+ + 2e$	Ionization	13.61	$k_7 = 9\times10^{-15}T_e^{0.7}\exp(-13.6/T_e)$	[4]

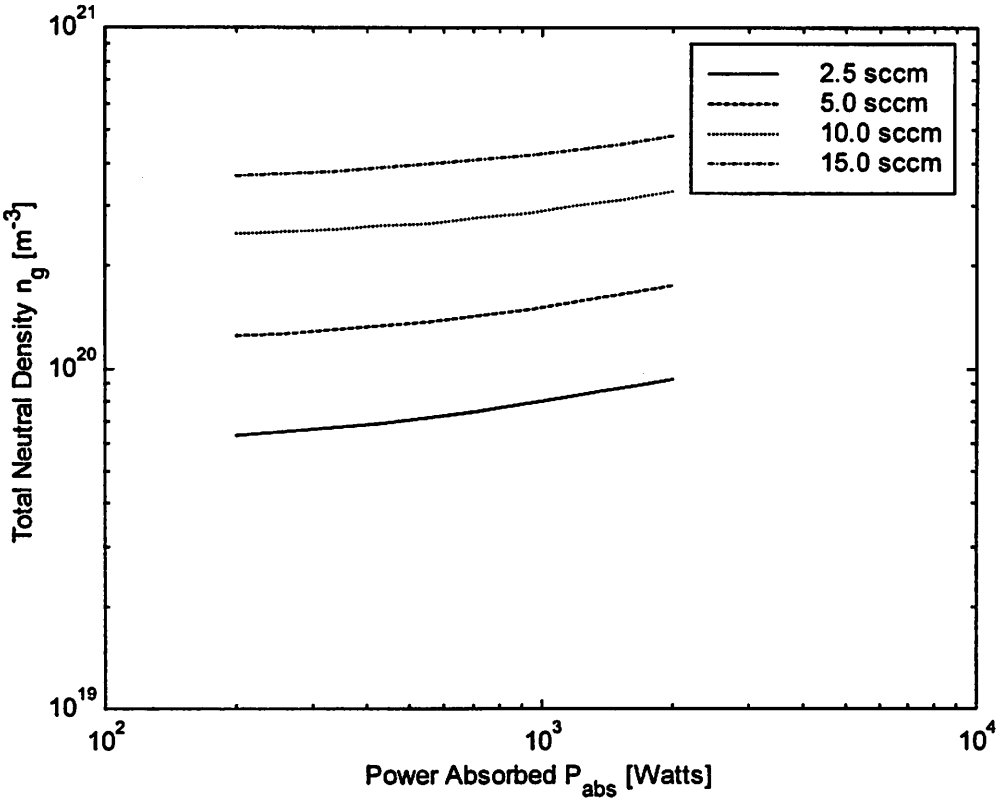

Figure 1.1: Total neutral concentration as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

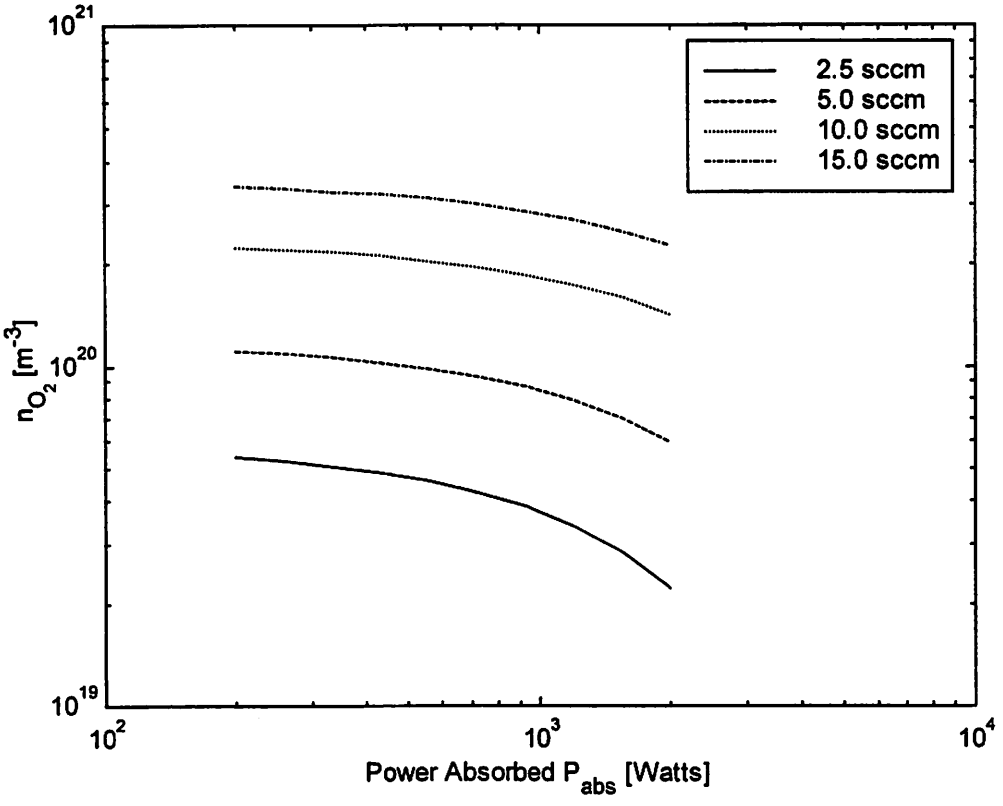

Figure 1.2: O_2 concentration as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

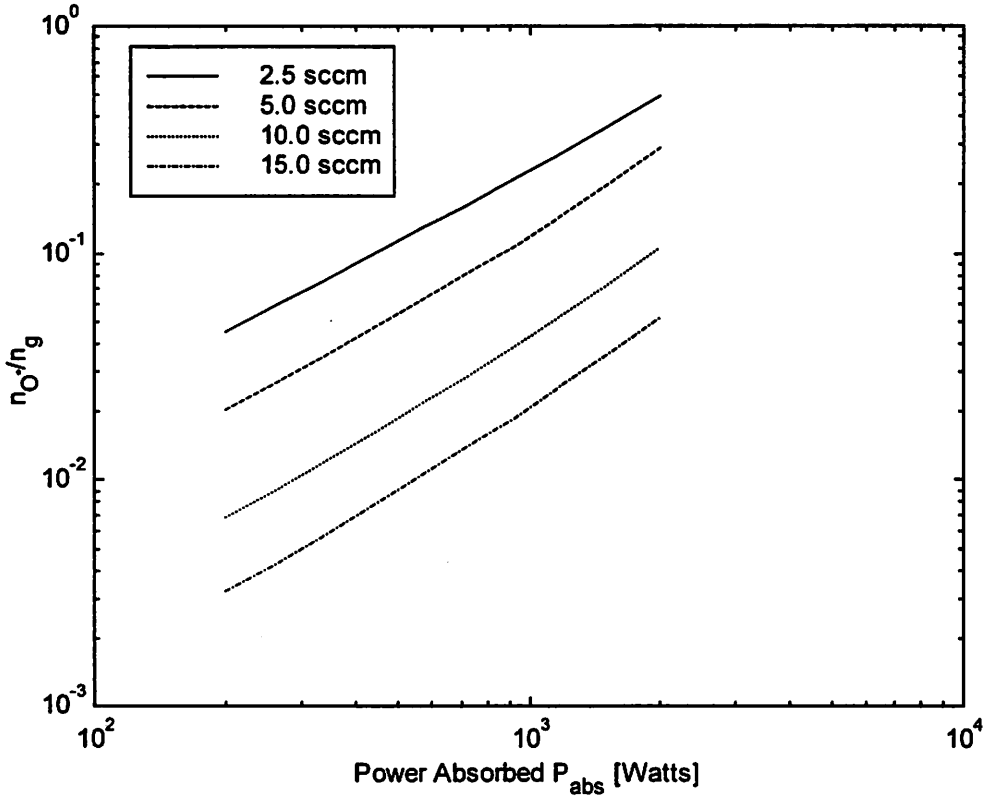

Figure 1.3: O (^{1}D) metastable concentration as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

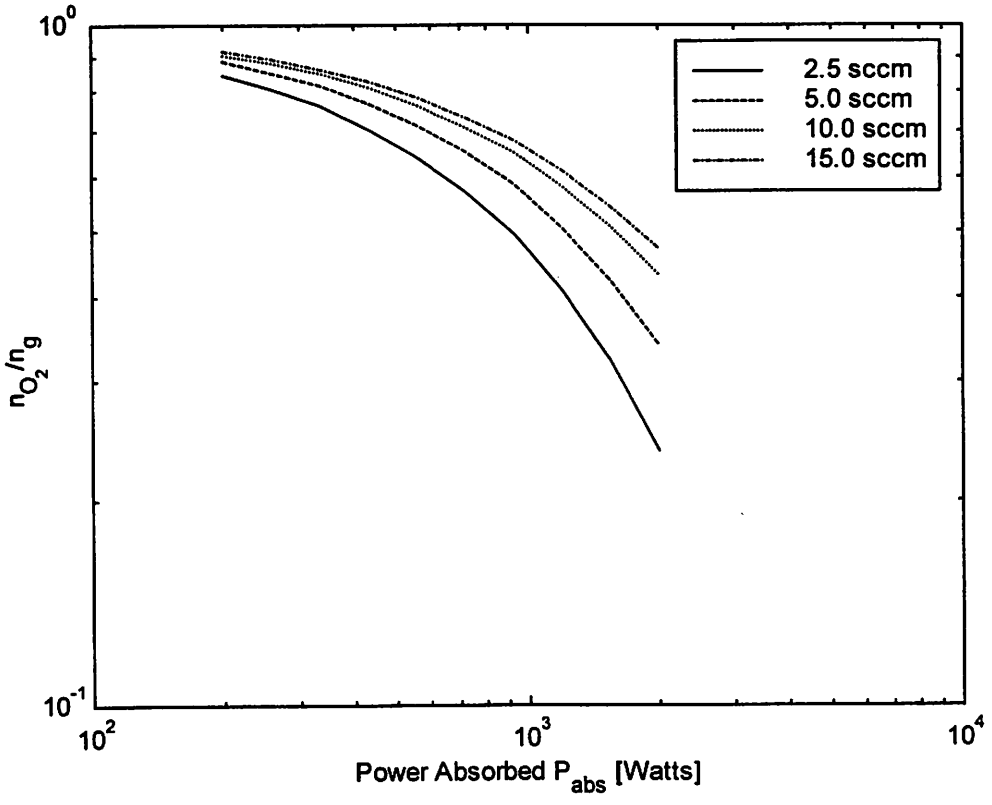

Figure 1.4: Fractional O_2 concentration as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

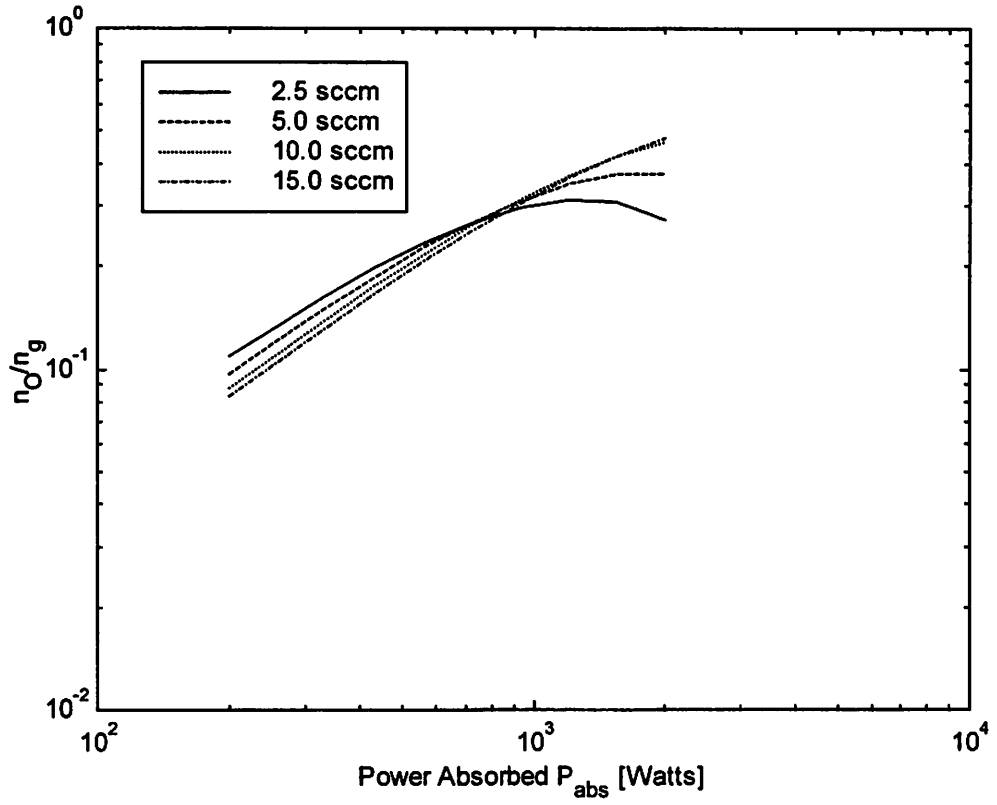

Figure 1.5: Fractional O concentration as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

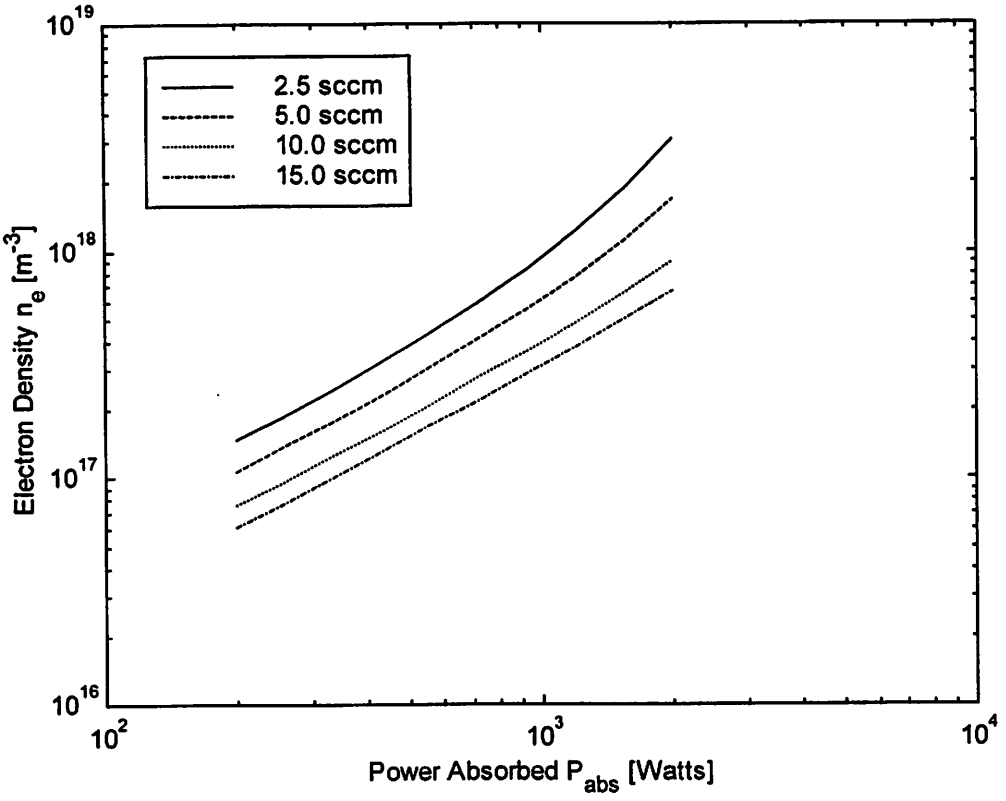

Figure 1.6: Electron density as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

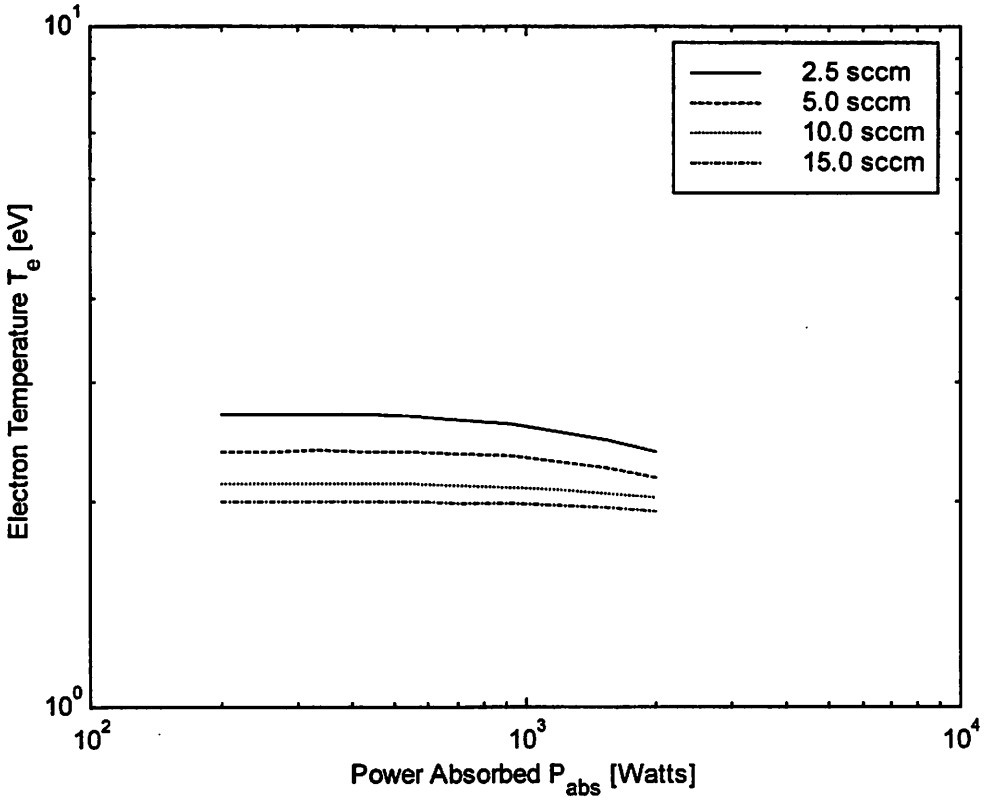

Figure 1.7: Electron temperature as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

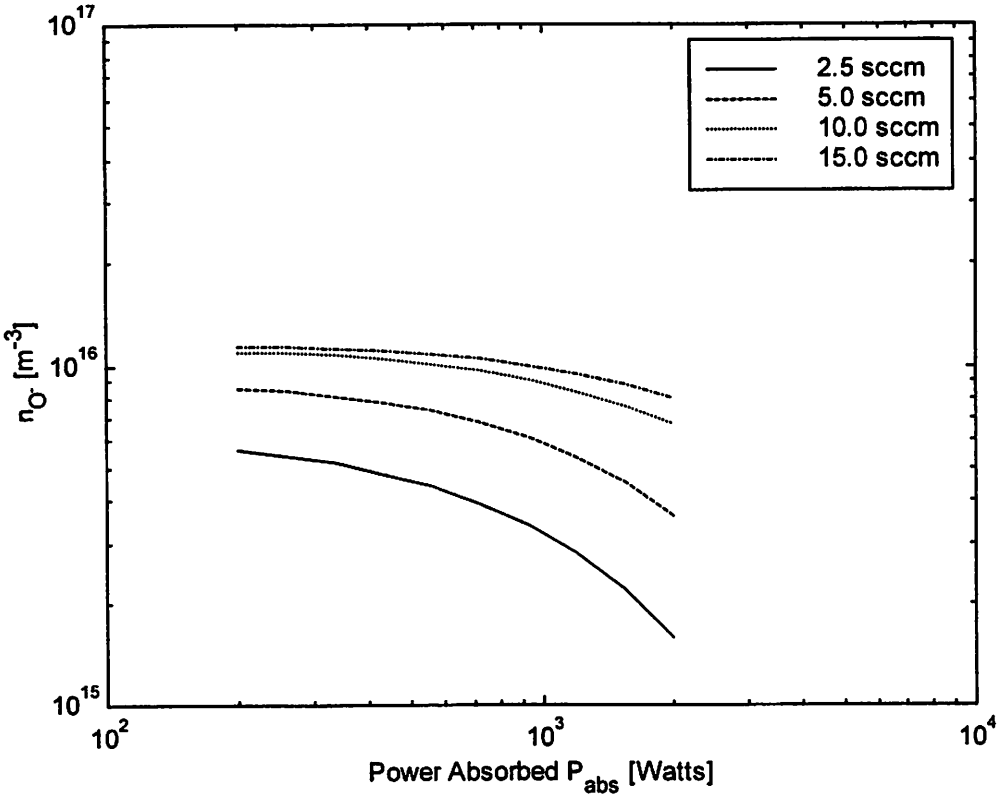

Figure 1.8: Negative ion density as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

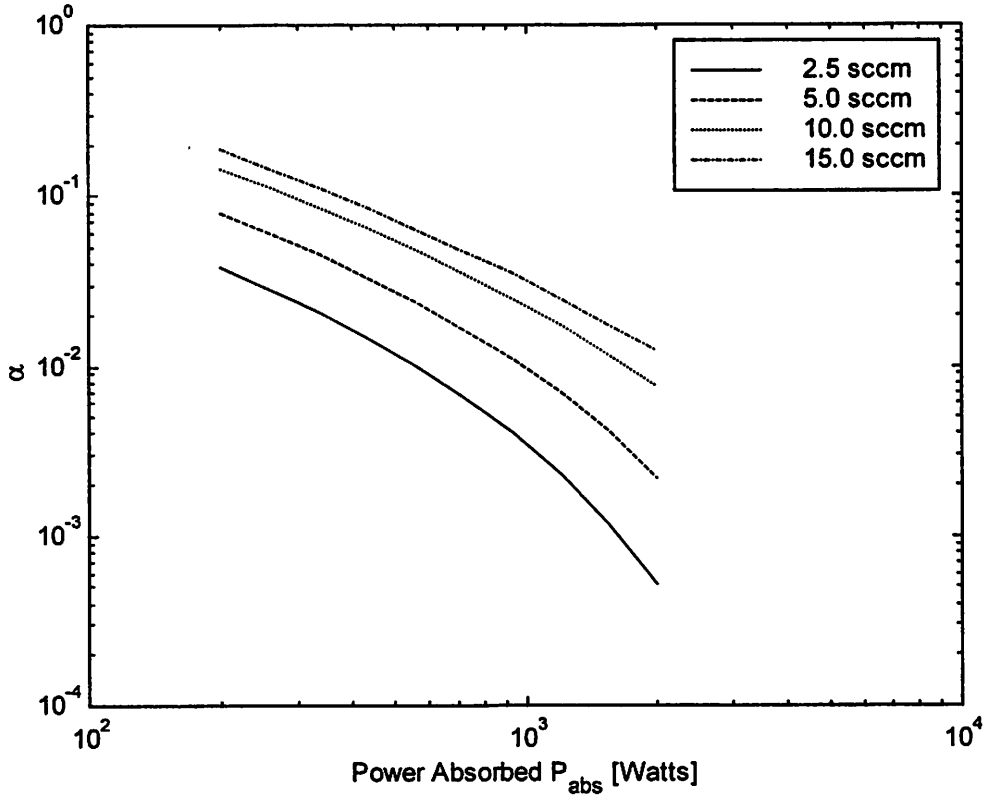

Figure 1.9: α as a function of input power. The value of $\alpha \equiv n_-/n_e$ reflects the electronegative nature of the bulk plasma. The pumping is fixed to give 1mTorr at 1.5 SCCM.

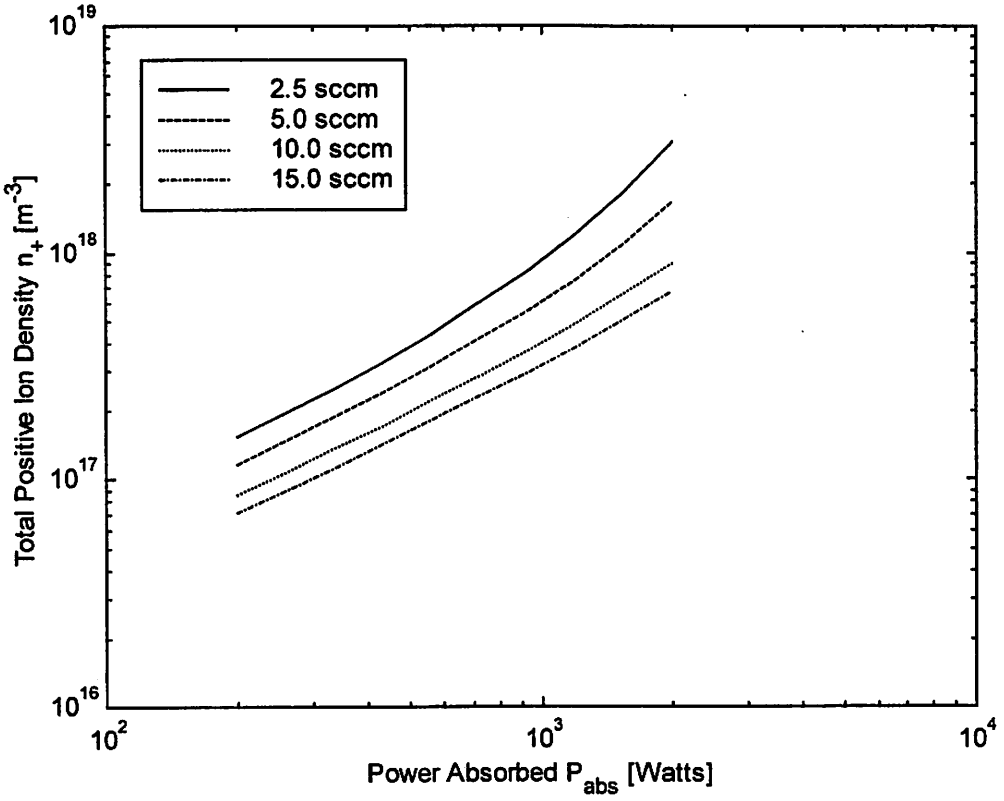

Figure 1.10: Total positive ion density as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

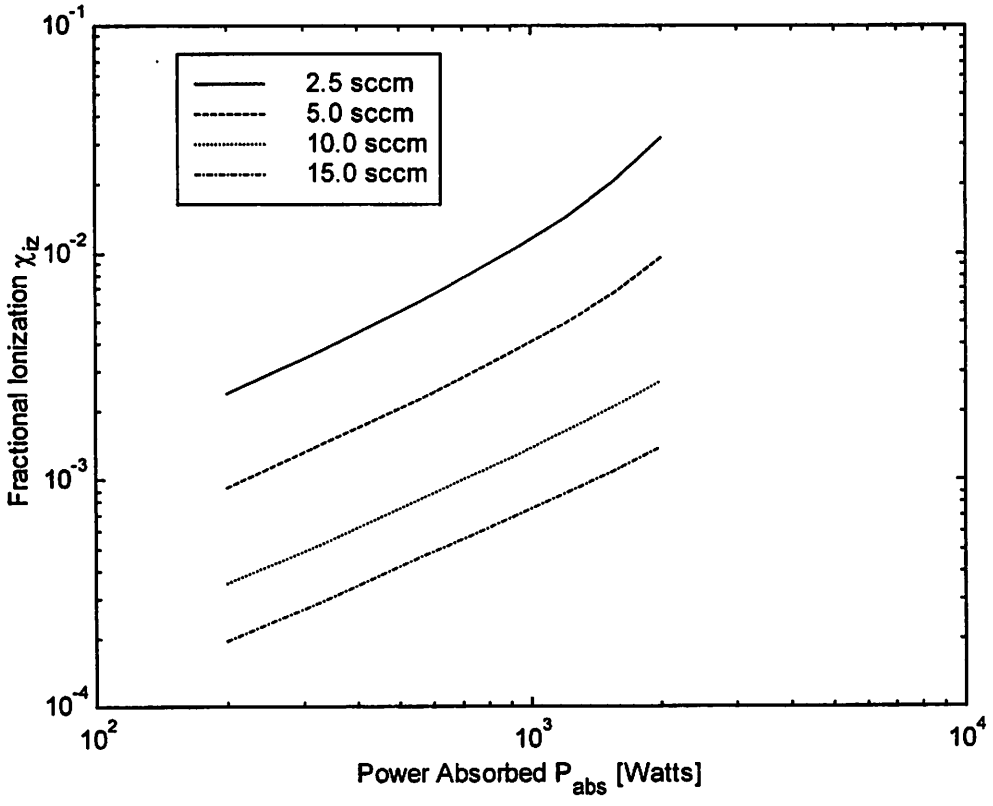

Figure 1.11: Fractional ionization as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

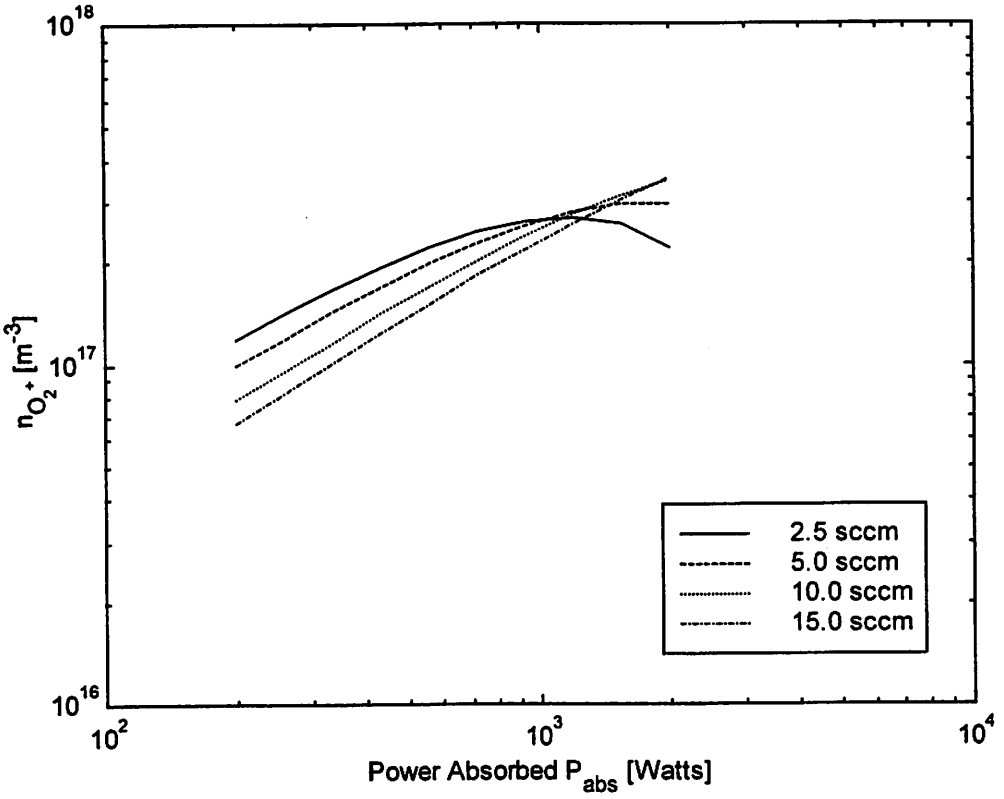

Figure 1.12: O_2^+ ion concentration as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

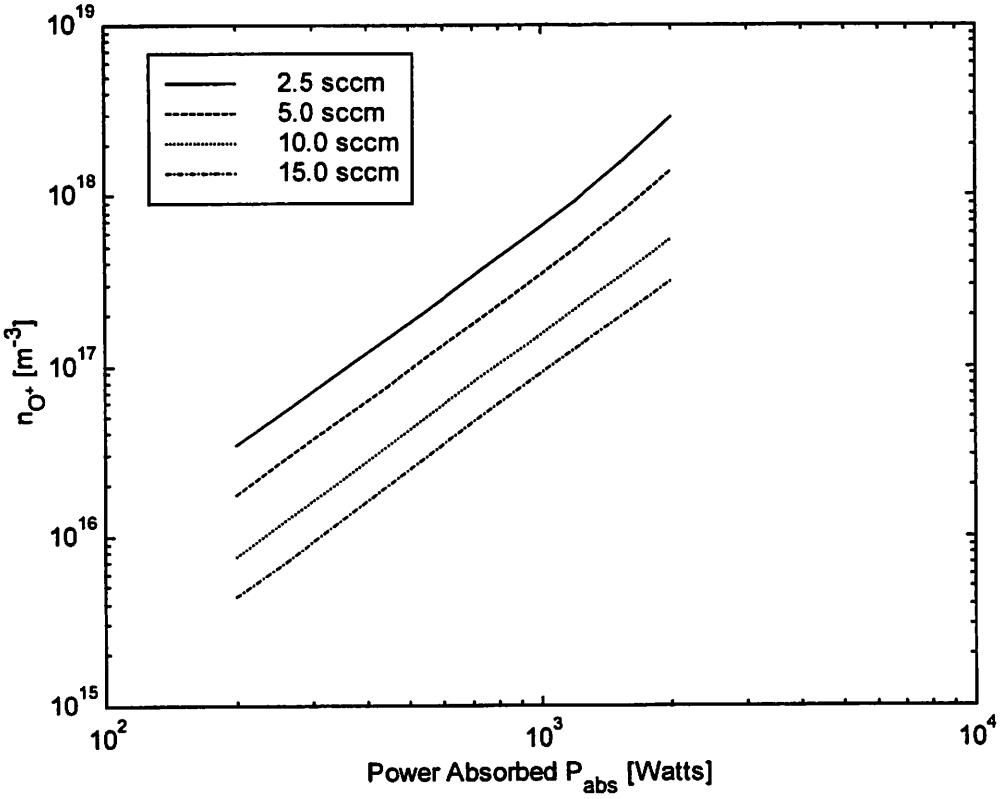

Figure 1.13: O^+ ion concentration as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

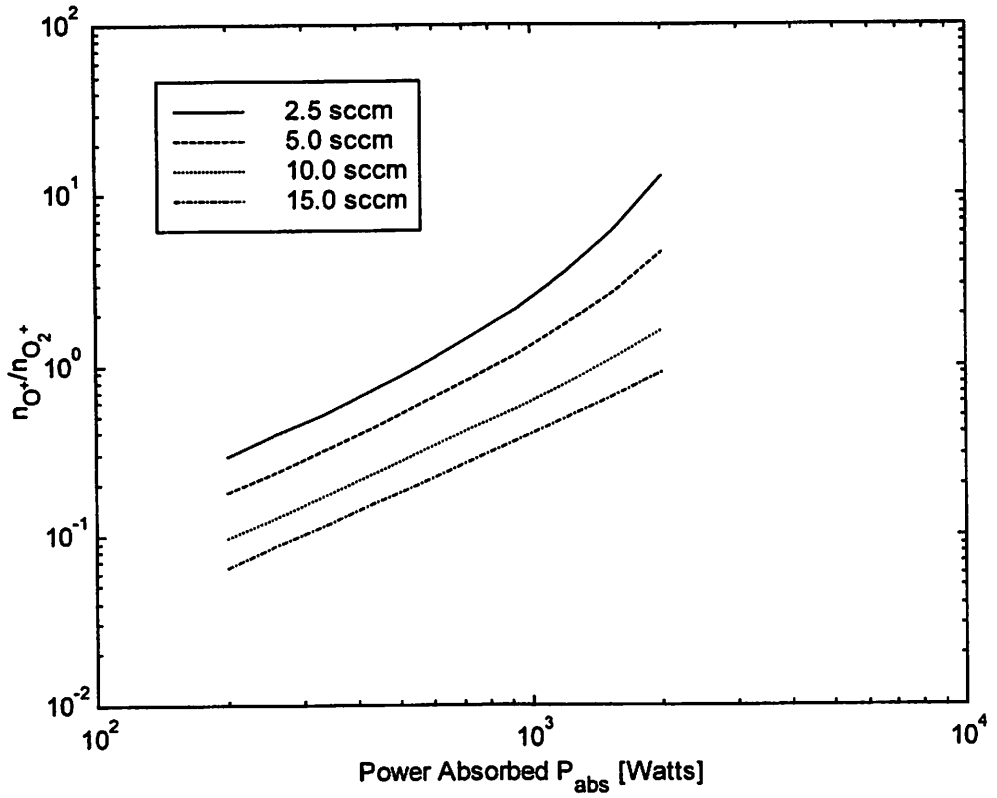

Figure 1.14: Ion concentration O^+/O_2^+ as a function of input power. The pumping is fixed to give 1mTorr at 1.5 SCCM.

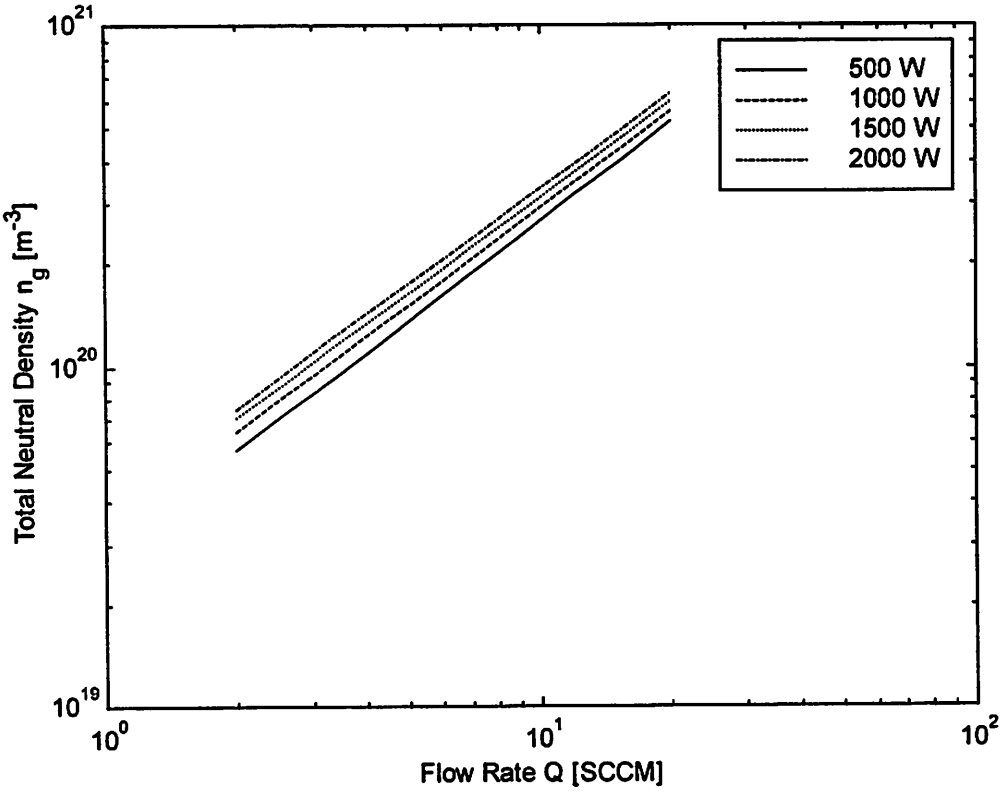

Figure 1.15: Total neutral concentration as a function of flow rate. This graph essentially shows the pressure variance as a function of gas flow. Since the pumping is fixed to give 1mTorr at 1.5 SCCM, the plot shows a straight line.

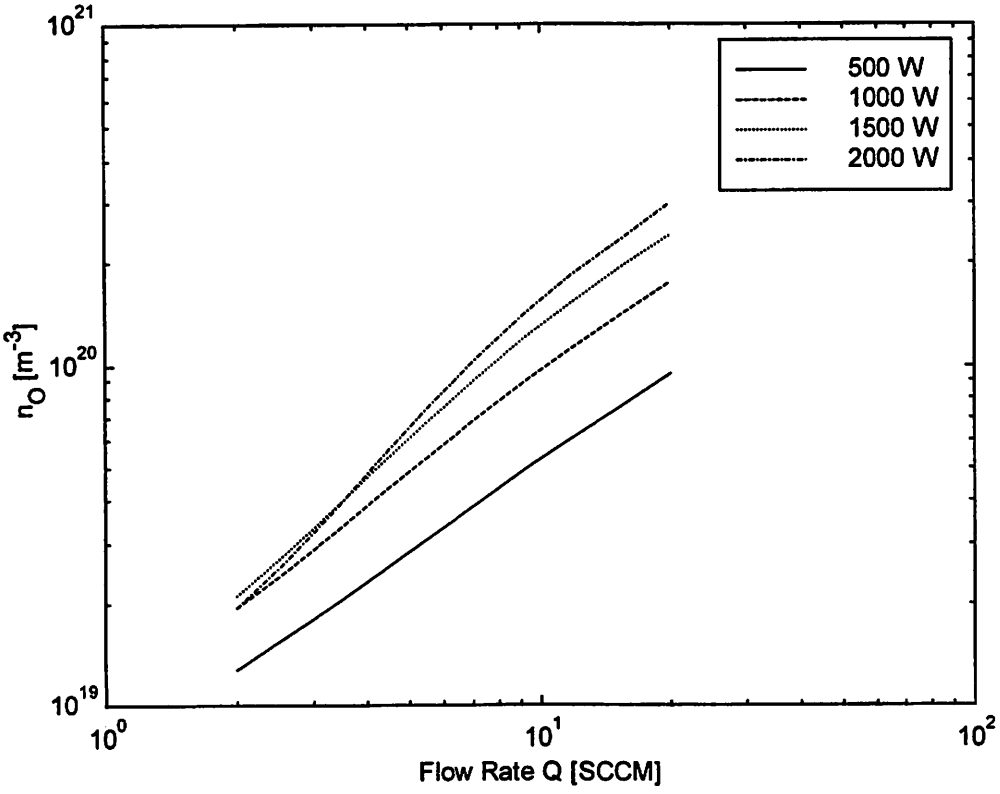

Figure 1.16: Atomic oxygen concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

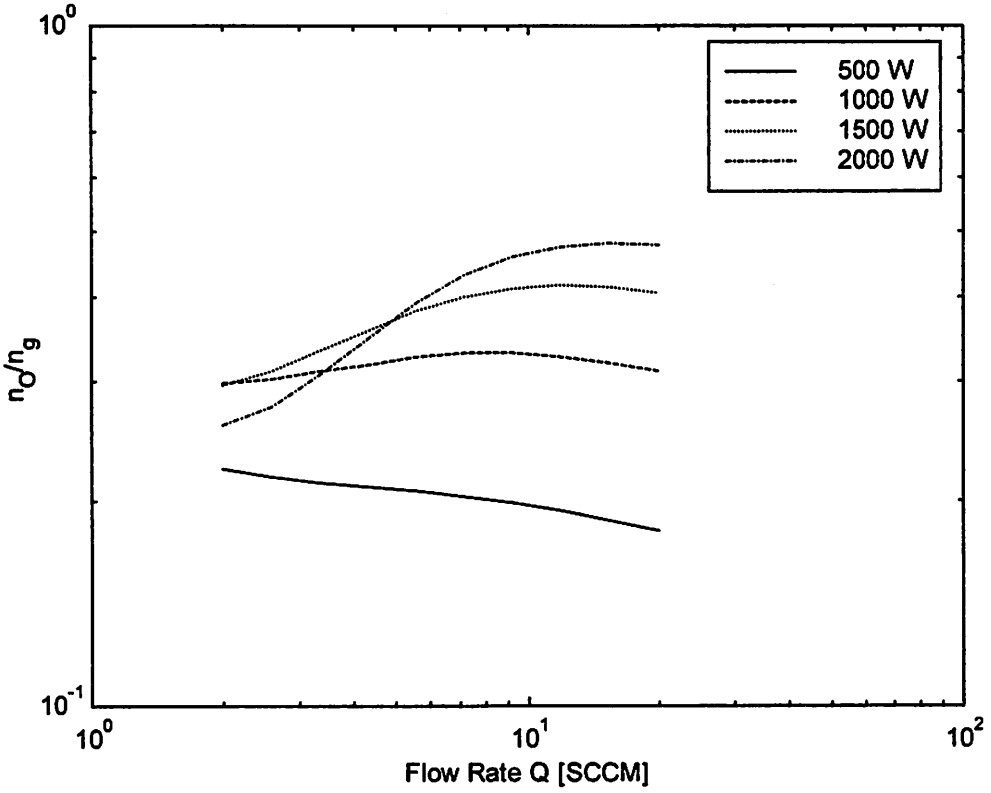

Figure 1.17: Fractional atomic oxygen concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

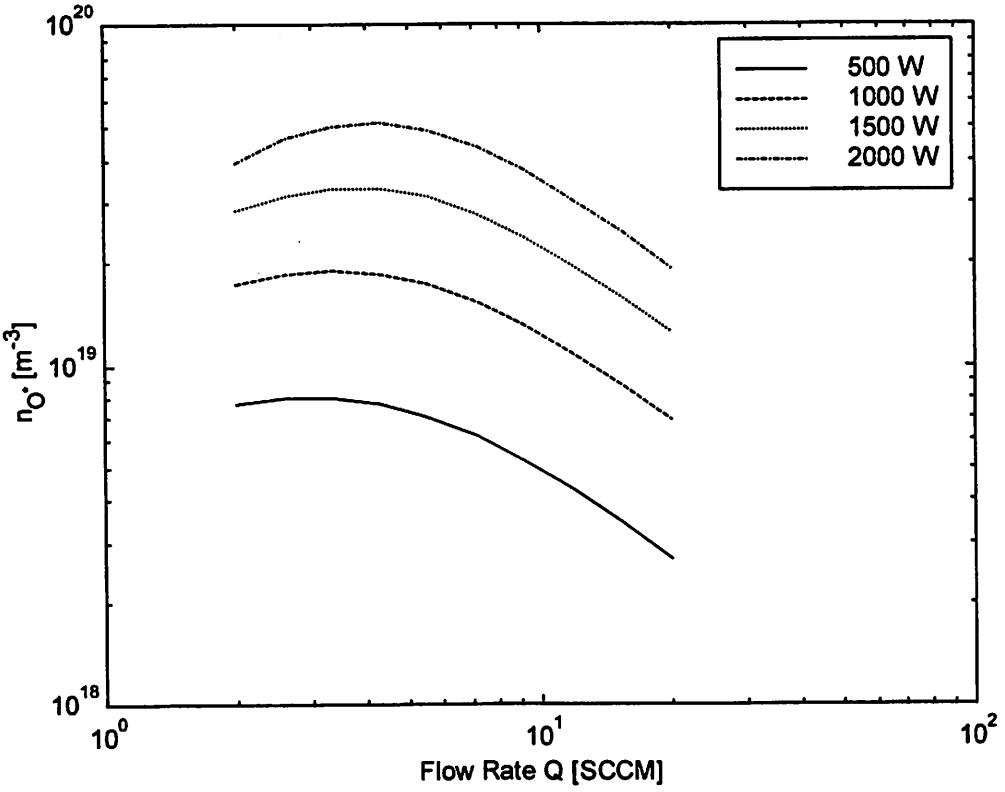

Figure 1.18: O (1D) metastable concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

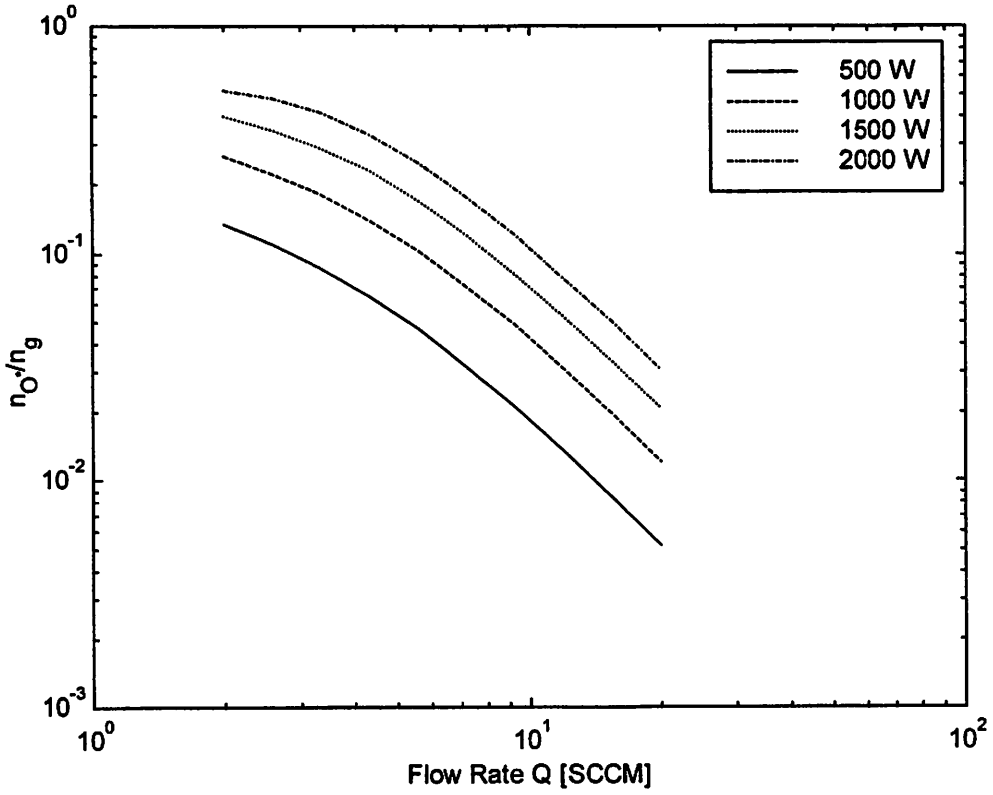

Figure 1.19: Fractional O (1D) metastable concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

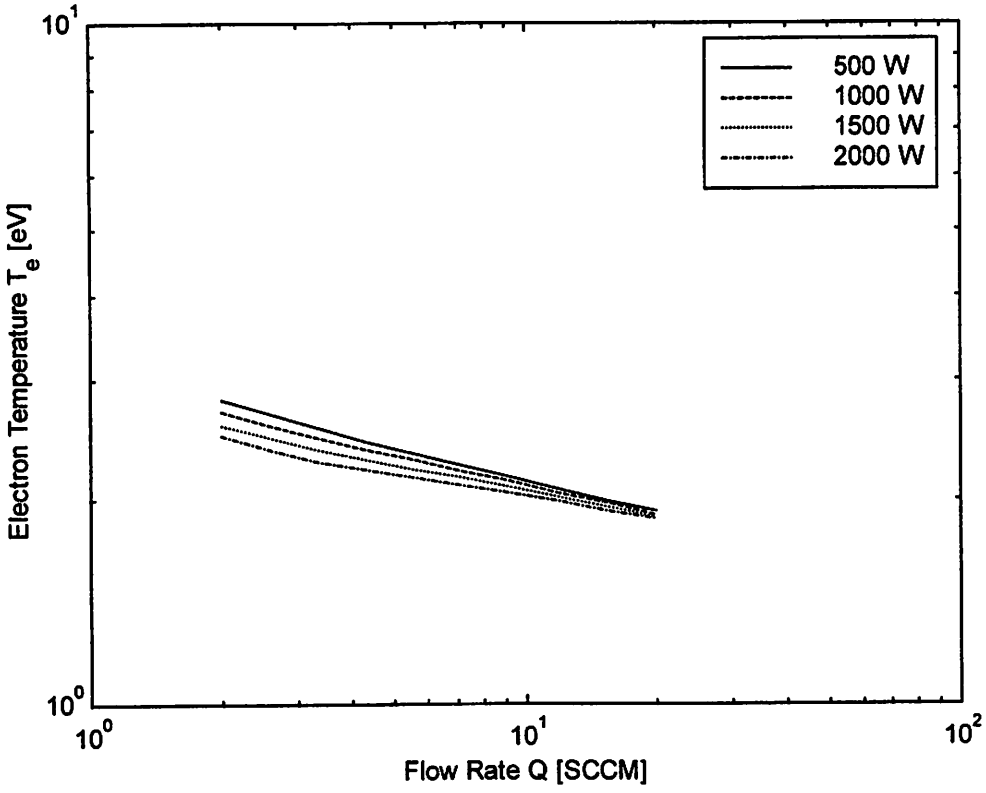

Figure 1.20: Electron temperature as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

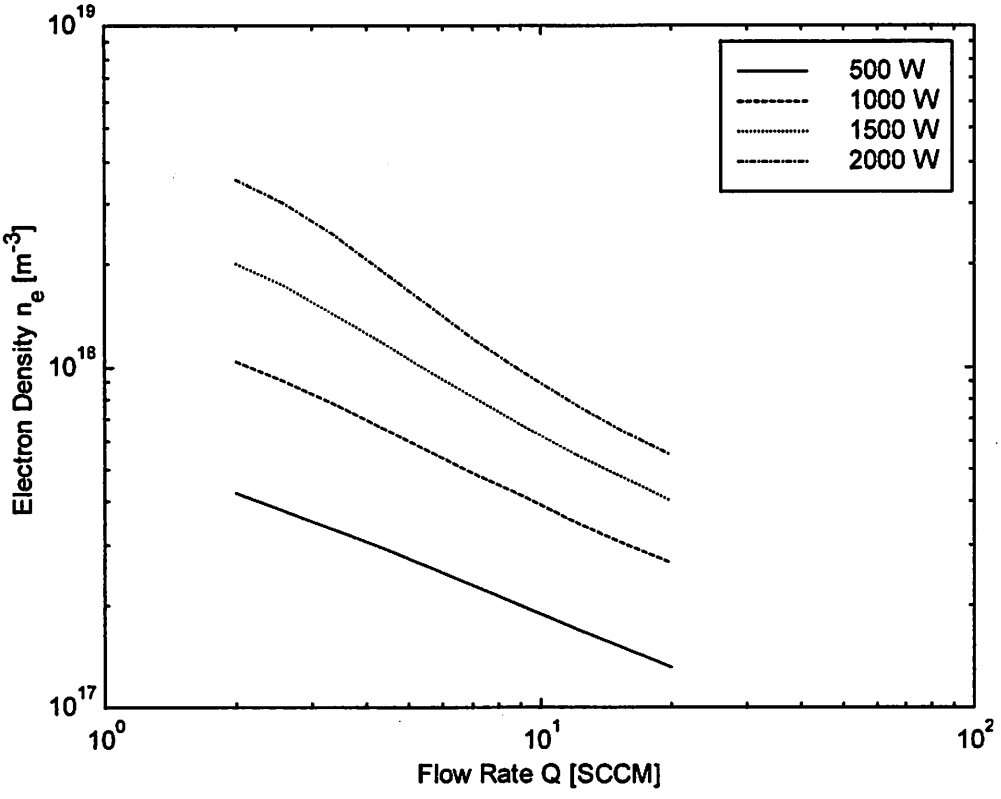

Figure 1.21: Electron concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

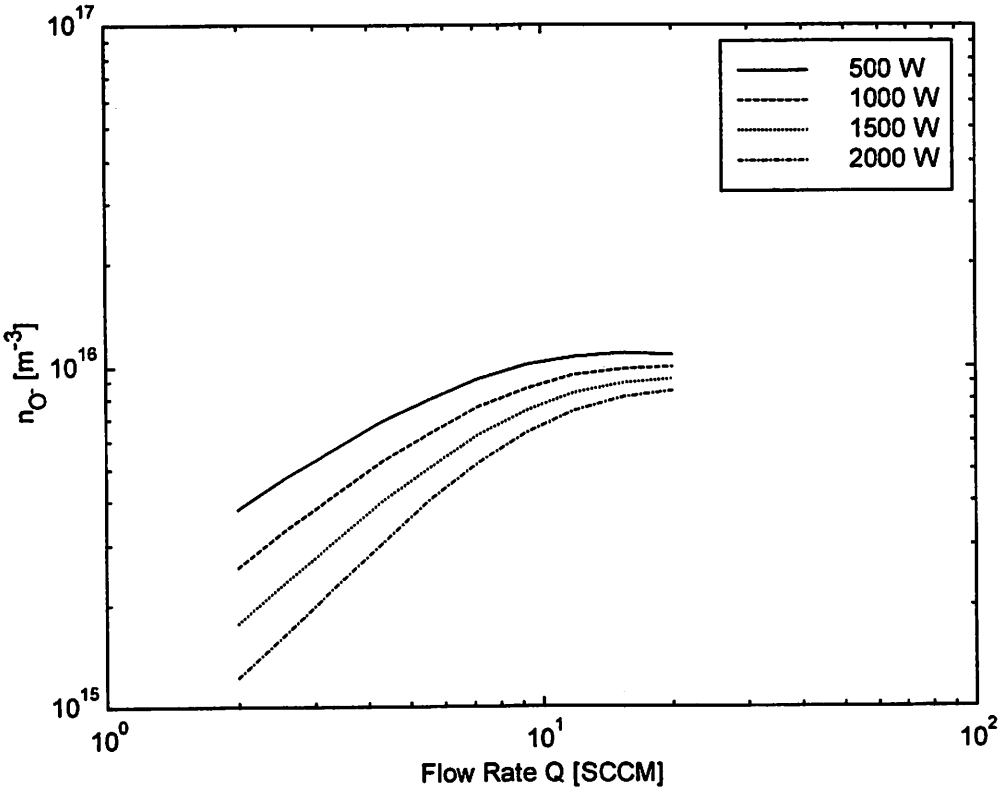

Figure 1.22: Negative ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

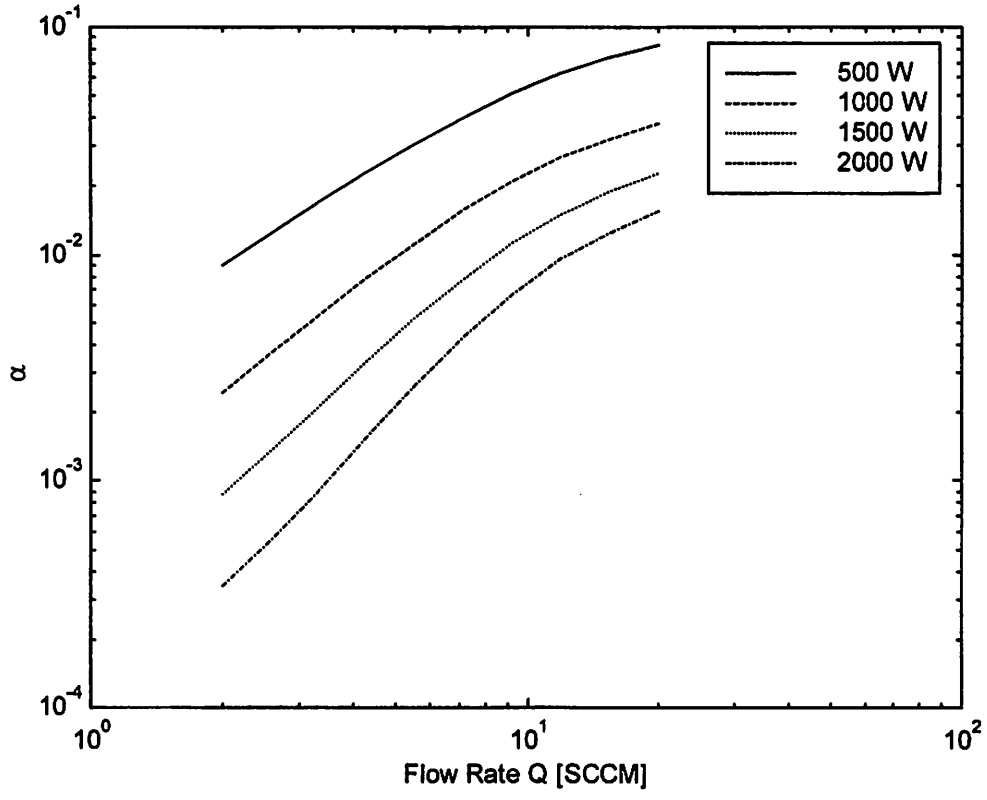

Figure 1.23: α as a function of input power. The value of $\alpha \equiv n_{-}/n_{e}$ reflects the electronegative nature of the bulk plasma. The pumping is fixed to give 1mTorr at 1.5 SCCM.

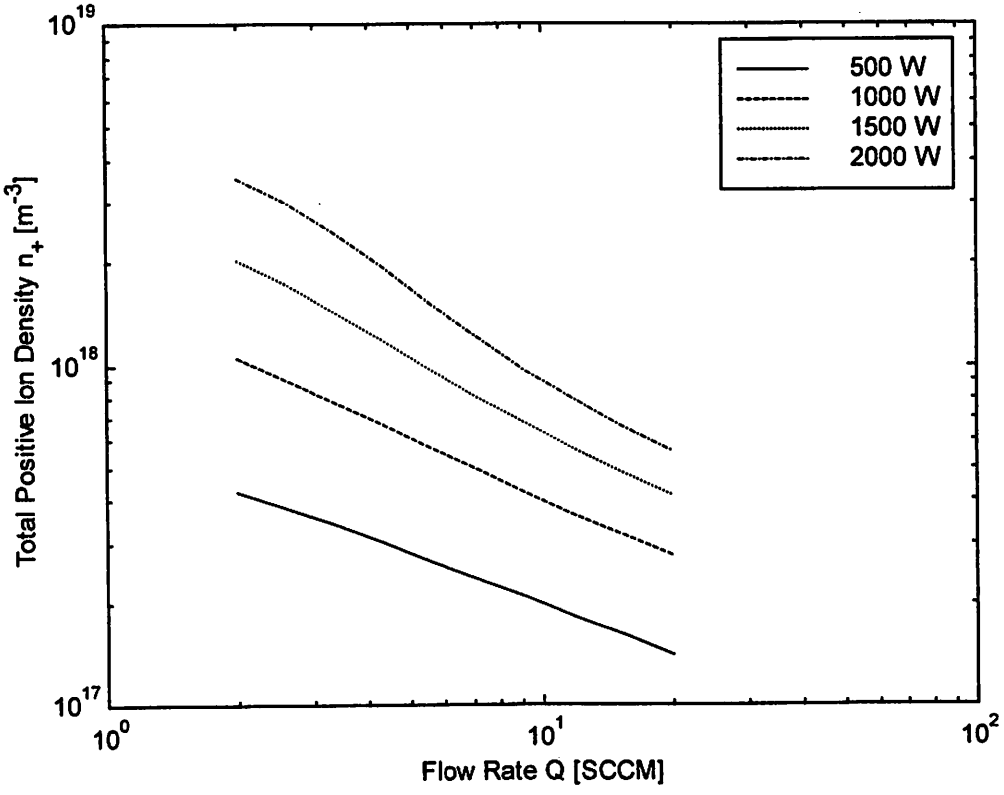

Figure 1.24: Total positive ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

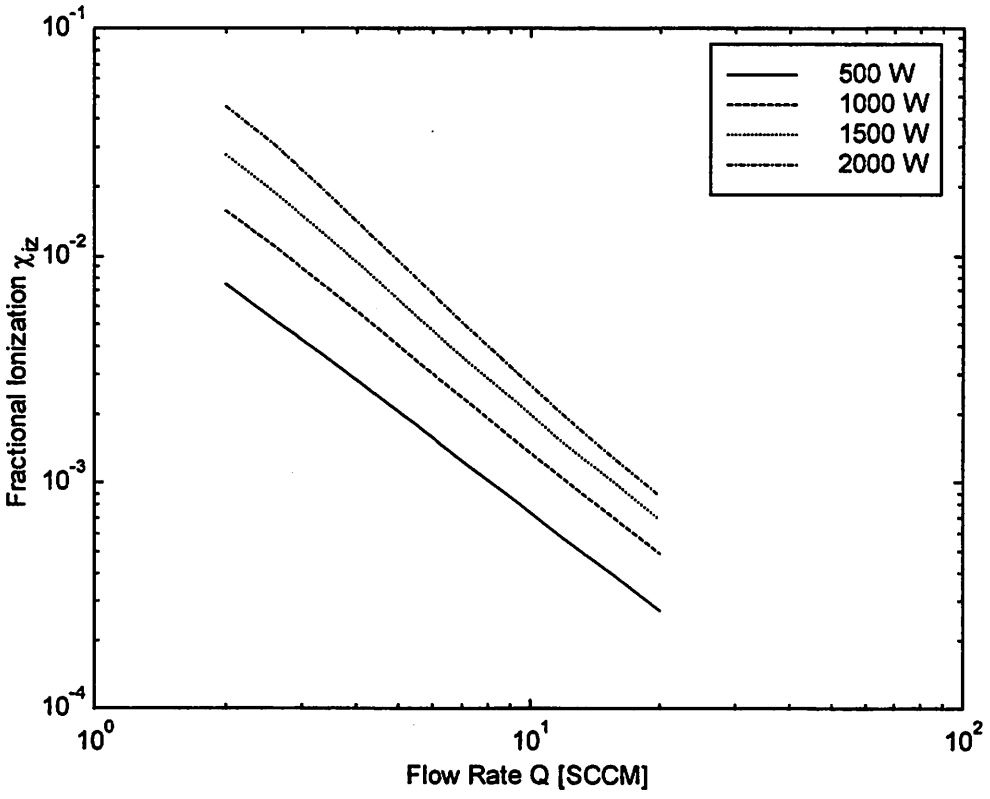

Figure 1.25: Fractional ionization as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

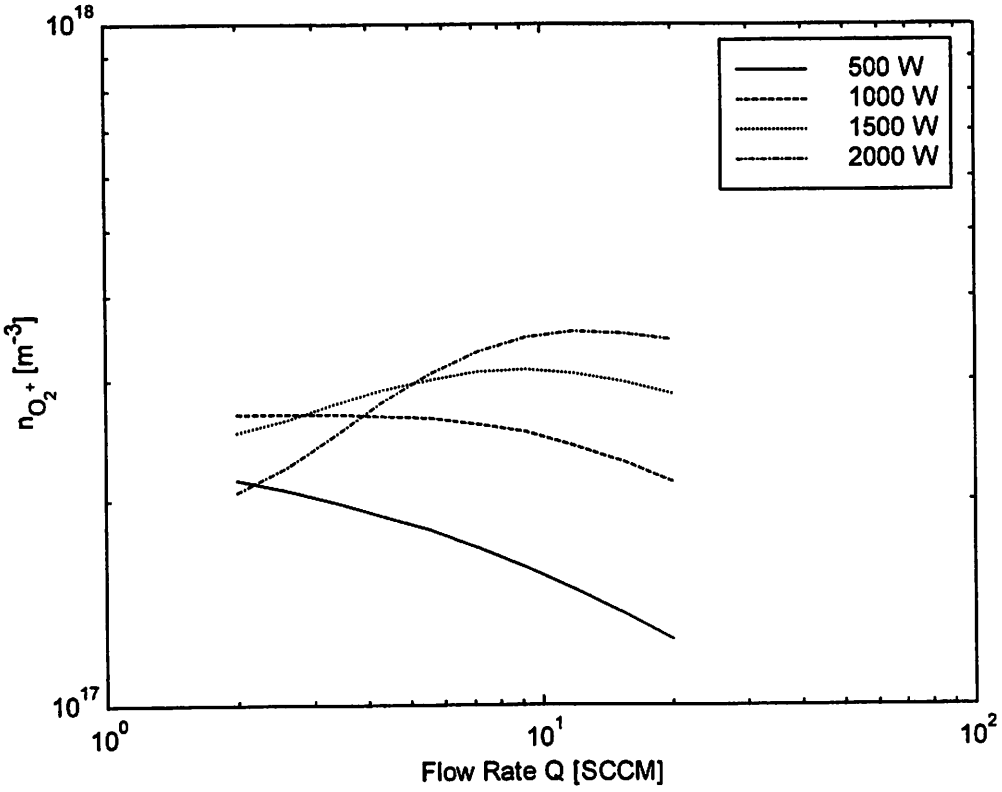

Figure 1.26: O_2^+ ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

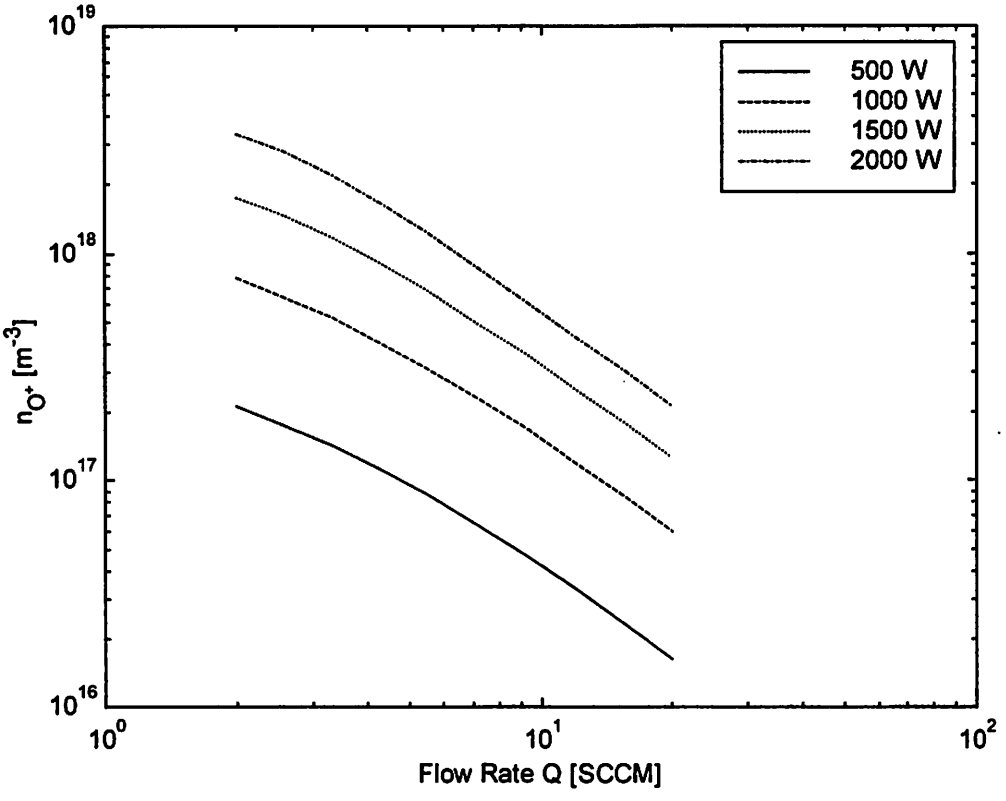

Figure 1.27: O^+ ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

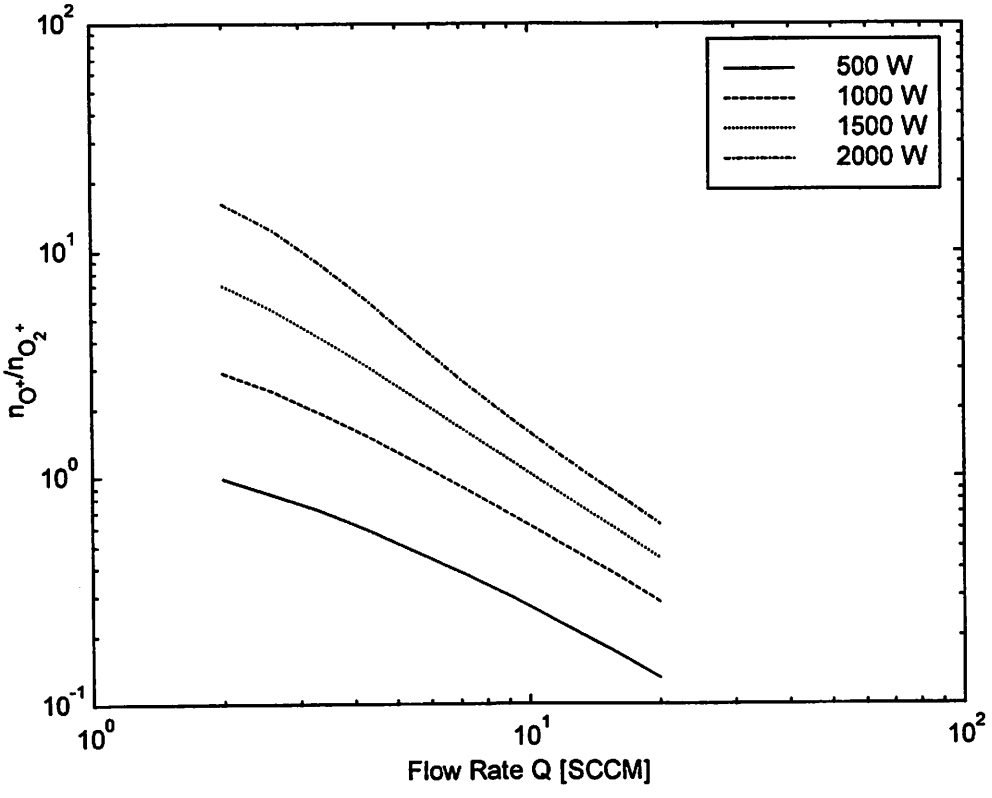

Figure 1.28: Ion concentration O^+/O_2^+ as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

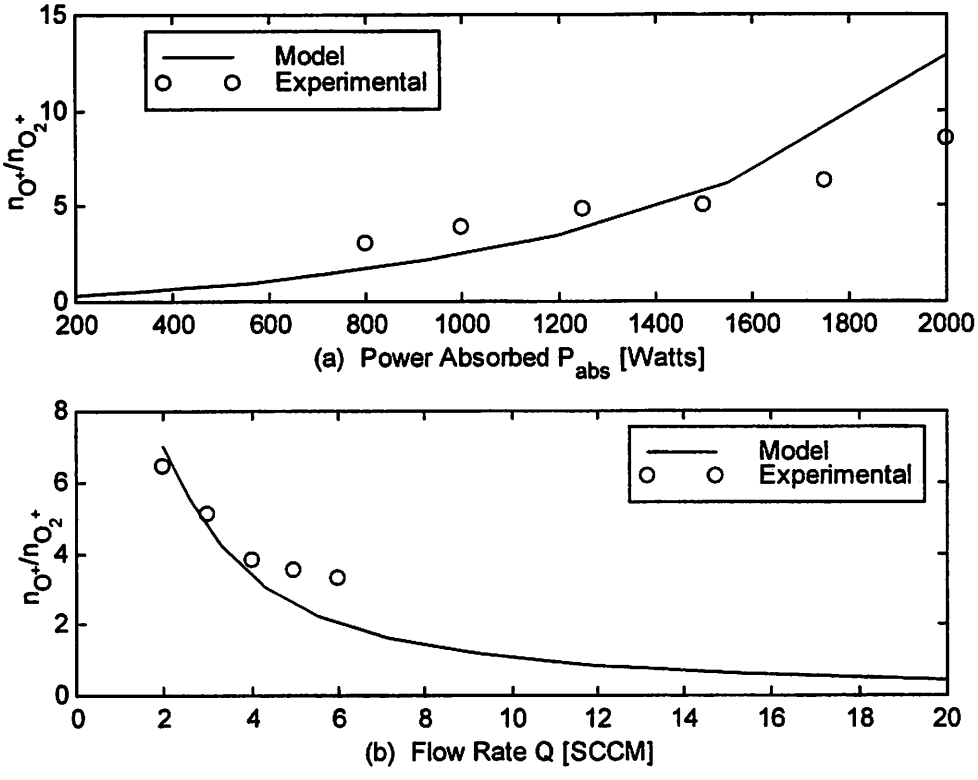

Figure 1.29: Comparison of ion concentration O^+/O_2^+ with experimental data. (a) Flow rate is 2.5 SCCM and reactor pressure is 1.67 mTorr. (b) Input power is 1500 Watts and pumping is fixed to give 1mTorr at 1.5 SCCM.

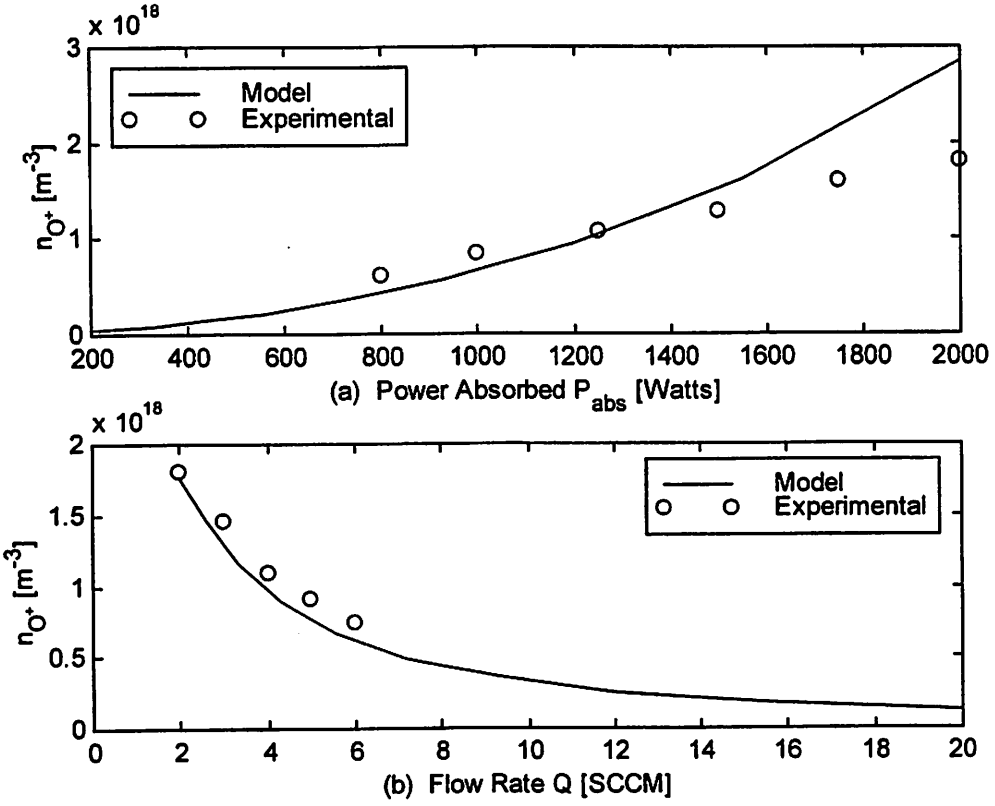

Figure 1.30: Comparison of O and O_2 ion concentration with experimental data. (a) Flow rate is 2.5 SCCM and reactor pressure is 1.67 mTorr. (b) Input power is 1500 Watts and pumping is fixed to give 1mTorr at 1.5 SCCM.

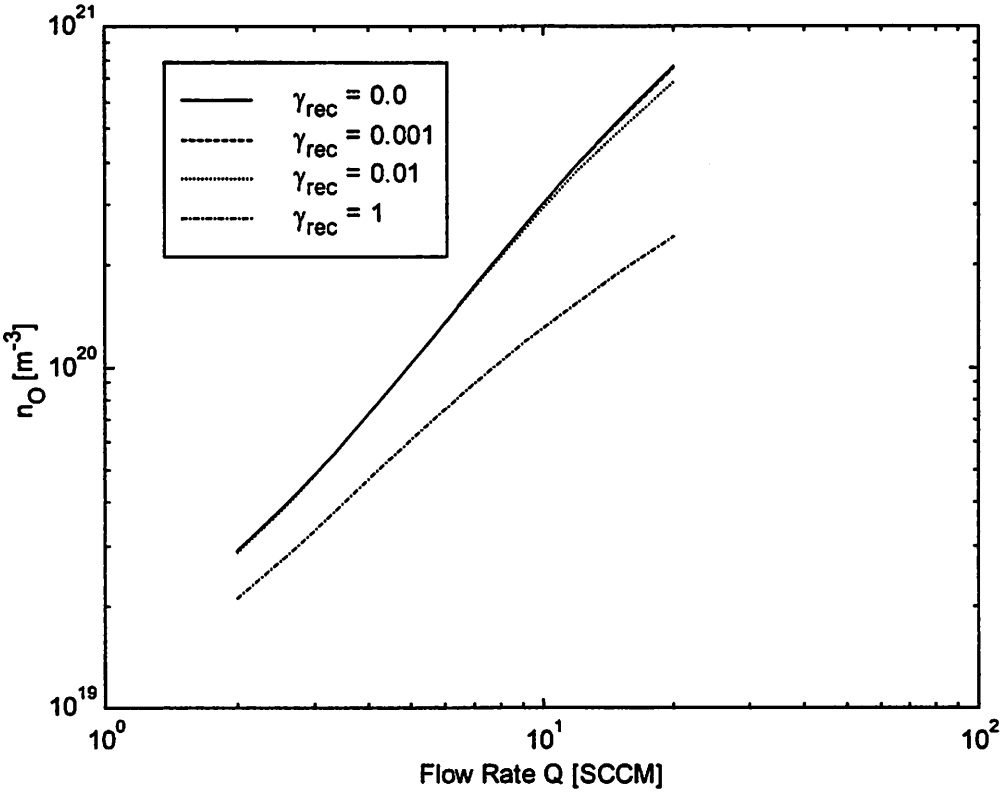

Figure 1.31: Atomic oxygen concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

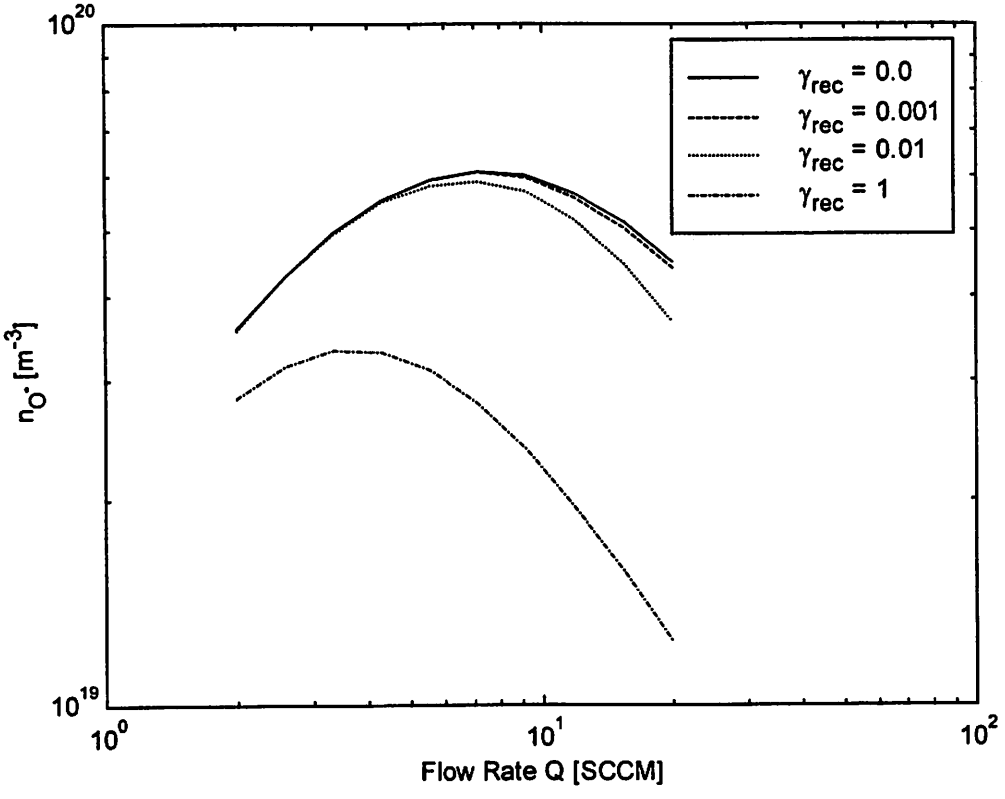

Figure 1.32: O (1D) metastable concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

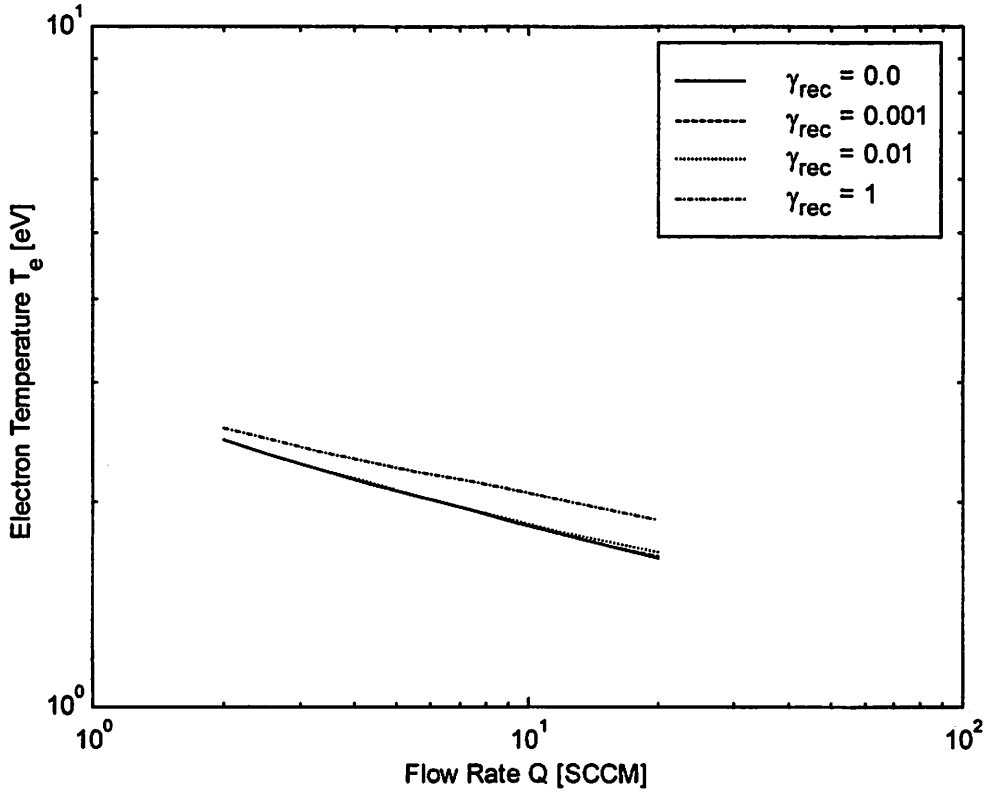

Figure 1.33: Electron temperature as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

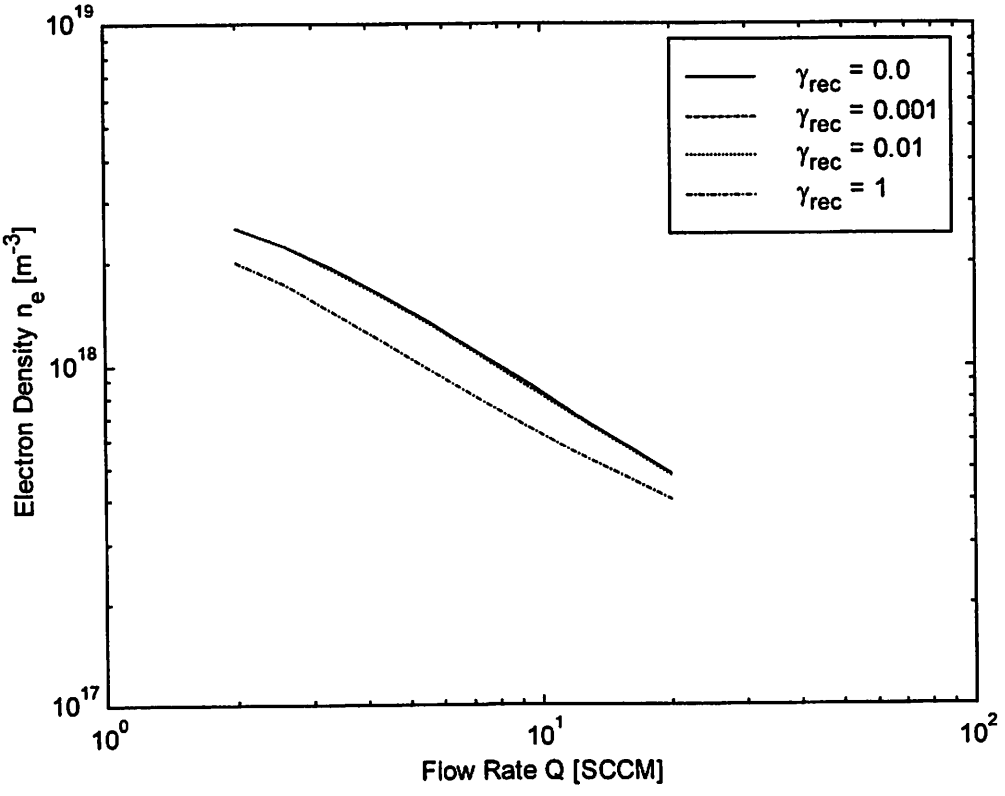

Figure 1.34: Electron concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

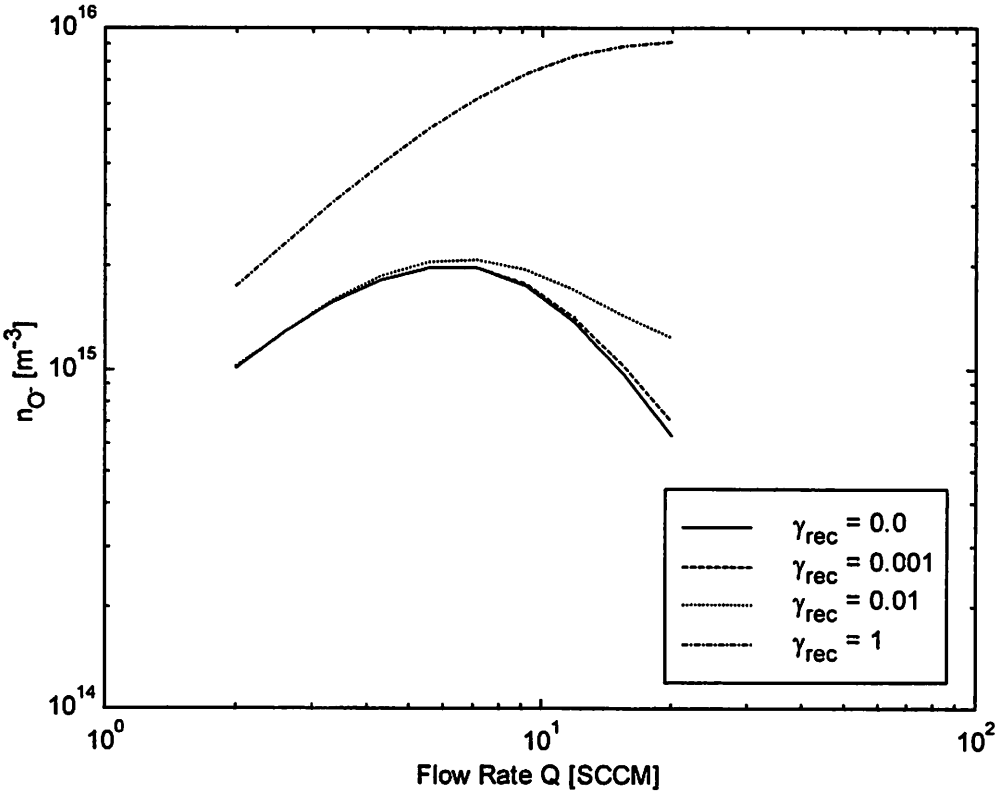

Figure 1.35: Negative ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

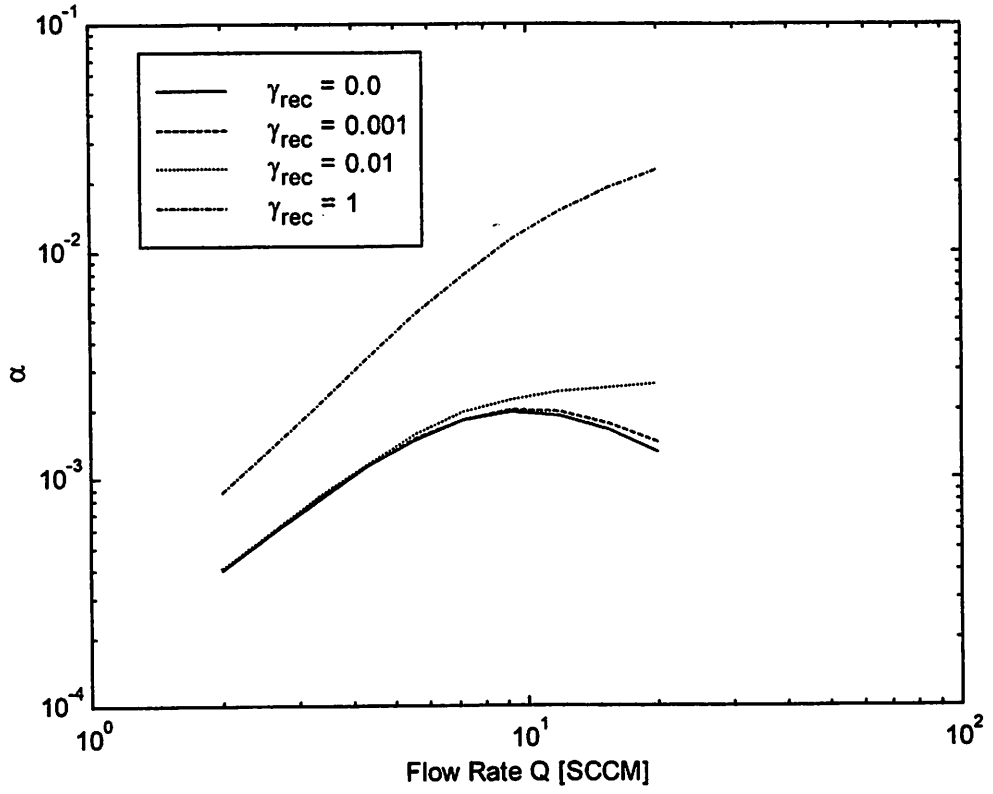

Figure 1.36: α as a function of input power. The value of $\alpha \equiv n_- / n_e$ reflects the electronegative nature of the bulk plasma. The pumping is fixed to give 1mTorr at 1.5 SCCM.

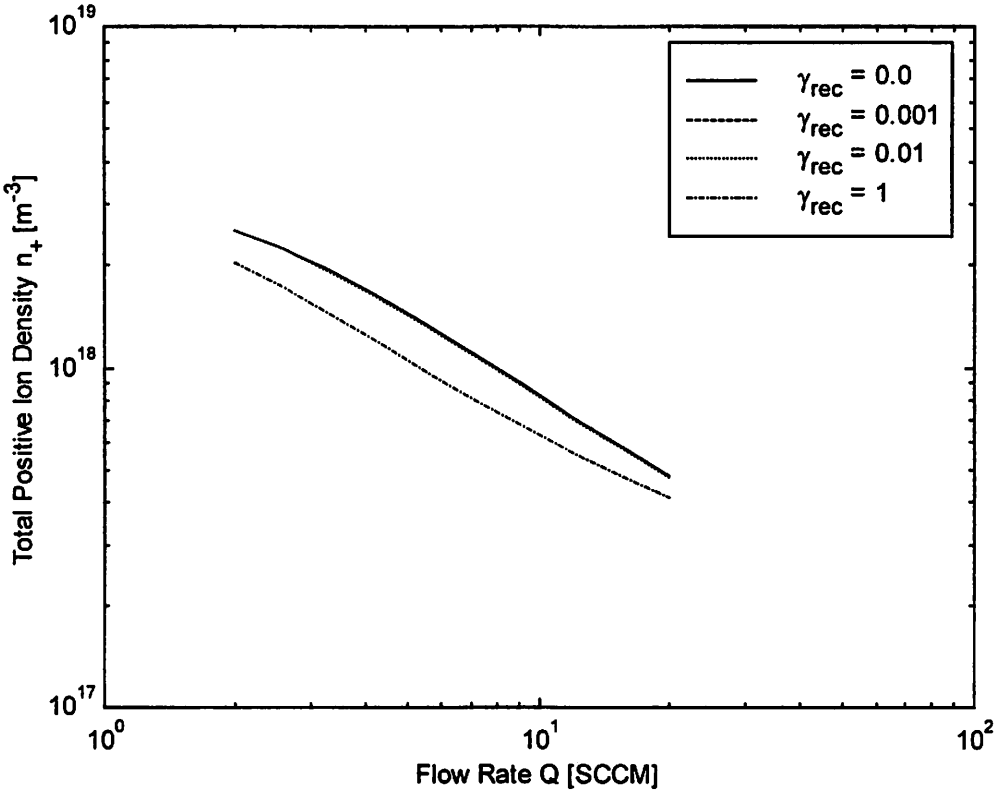

Figure 1.37: Total positive ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

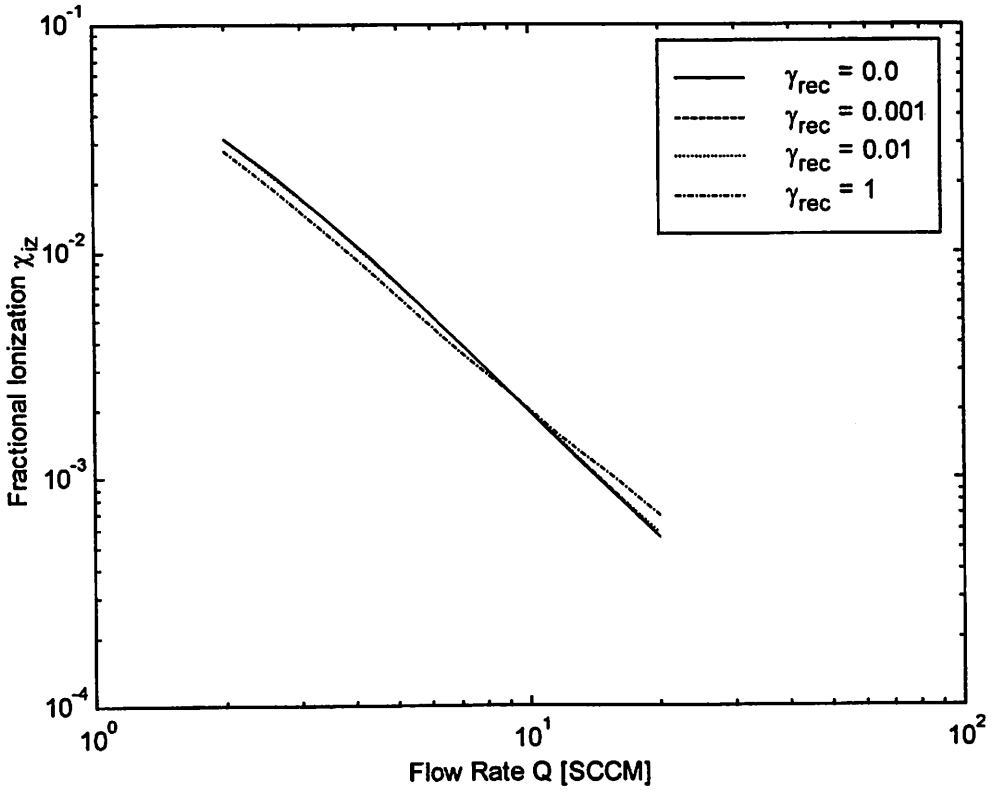

Figure 1.38: Fractional ionization as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

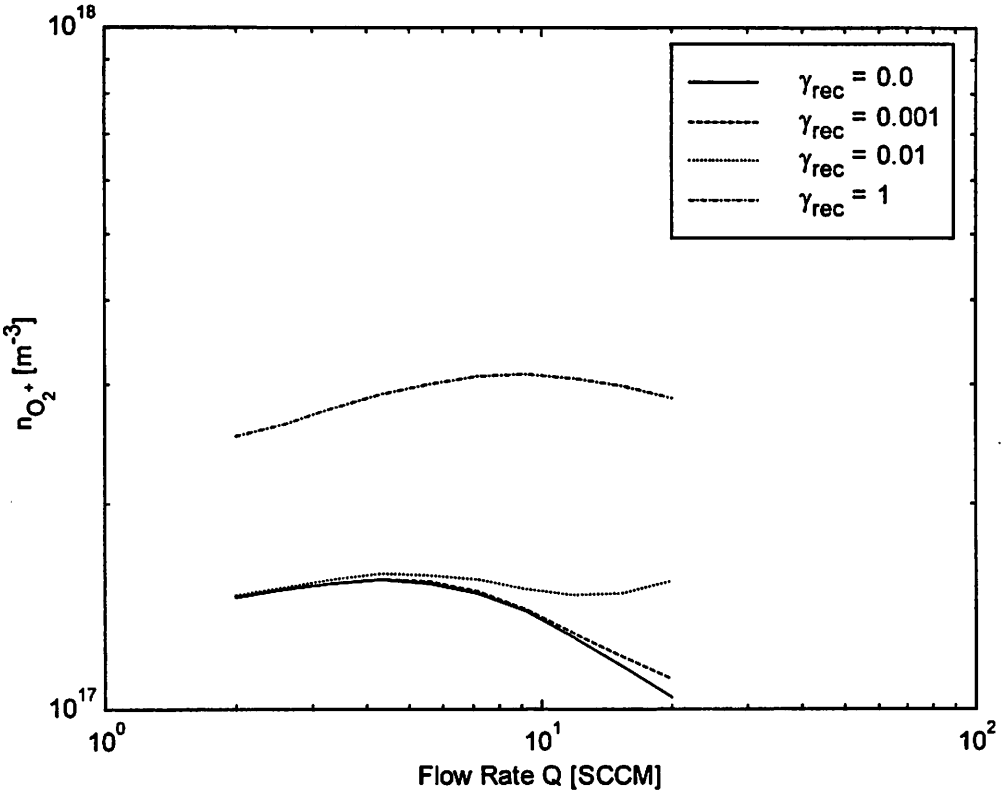

Figure 1.39: O_2^+ ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

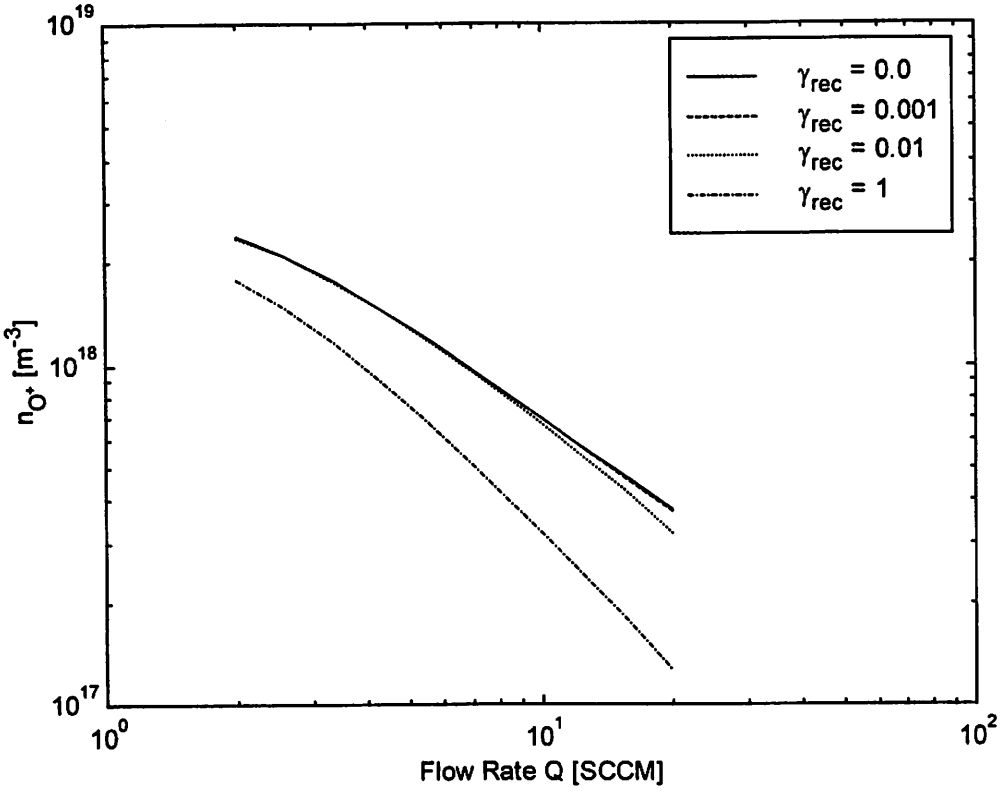

Figure 1.40: O^+ ion concentration as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

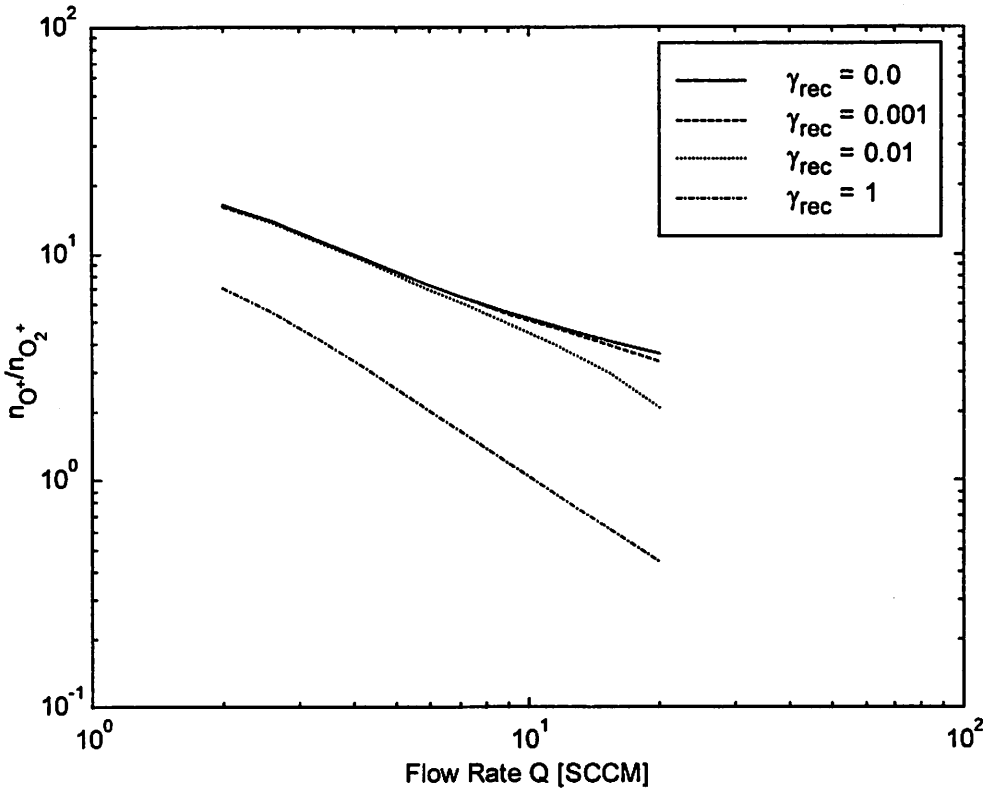

Figure 1.41: Ion concentration O^+/O_2^+ as a function of gas flow. The pumping is fixed to give 1mTorr at 1.5 SCCM.

Chapter 2

Boron Trifluoride Discharge

2.1 Introduction

The need to fabricate shallow junctions has increased in order to avoid short channel effects in CMOS devices [22]. Doping of silicon with boron in such processes requires very low energy implantation. Two approaches for low energy boron implantation are plasma ion immersion implantation (PIII) and conventional beamline implanters. Operating at ultra-low energy, Matyi et al. [23] have demonstrated the advantages of PIII. PIII has been shown to successfully create ultra-shallow P^+/N junctions using boron as a dopant [22]. In this technique, boron trifluoride (BF_3) and diborane (B_2H_6) have been conventionally used as process gases. Although BF_3 is corrosive, it less toxic and hence safer than B_2H_6. Efforts have been made to study the plasma chemistry in PIII [24].

At the heart of an implanter is the source of plasma. Ion species are created in the source and may be separated using a mass-analyzer to discard the contaminants. An understanding of the composition of multi-species BF_3 discharges can be indispensible for optimizing process conditions, dosage and gas recipes. The surface chemistry, which involves adsorption, desorption, and reactions between resident surface species, is crucial to this understanding. In this work, the surface is assumed to have boron coverage at active sites. A simple gas and surface

model, incorporating multipole magnetic confinement of the plasma, is developed to match the experimental data obtained from the Eaton ULE2 ion source.

2.2 Neutral Volumetric Losses

An effective diffusion length (Λ_0) can be used for analysis of volumetric neutral particle loss due to diffusion provided (i) the mean free path (λ_m) is small compared to the container dimensions, and (ii) particles are annihilated on impact with the walls by adsorption, recombination, or quenching surface processes. For a right circular cylinder, Chantry [25] found the effective diffusion length to be

$$\Lambda_0 = \left[\left(\frac{\pi}{L}\right)^2 + \left(\frac{2.405}{R}\right)^2\right]^{-1/2} \tag{2.1}$$

where R is the radius and L is the length of the cylindrical reactor. A good approximation for the diffusion loss frequency (rate coefficient) for neutral species is [12]

$$k_{wg} = \left[\frac{\Lambda_0^2}{D_g} + \frac{2V(2-\gamma)}{A v_g \gamma}\right]^{-1} \tag{2.2}$$

where, D_g is the neutral diffusion coefficient (1.7), v_g is the mean neutral speed (1.8), γ is the sticking coefficient, and V and A are the volume and wall surface area of the reactor respectively. In (2.2) both terms are found to be of comparable magnitude and therefore neither can be neglected. For $\gamma < 0.1$ the diffusion-limited term (first term) in (2.2) becomes insignificant and the neutral atom density is found to be nearly uniform throughout the discharge, slightly lower near the walls than in the bulk.

2.3 Surface Chemistry Model

In a BF_3 discharge, the surface chemistry is complex and not well understood. The reactor wall serves as a site for complex processes such as physical sputtering, ion impact fragmentation, neutral adsorption-desorption, neutral adsorption-reaction-desorption, etc. In a BF_3 discharge, boron is adsorbed on the reactor walls. A significant amount of boron containing species (mostly BF_2^+ ions) has been observed in a NF_3 discharge after having run a BF_3 discharge in the same reactor. This suggests that boron can be rapidly removed from the wall by fluorine-containing discharges. A similar phenomenon has been experimentally studied by Toyoda et al. [26]. They showed that rapid cleaning of boron thin films was possible with use of CF_4-H_2 discharge. Further, their mass spectroscopic measurements revealed that the film was removed as BF_3. Toyoda et al. [26] also showed that the addition of 2-3% oxygen enhanced the removal rate by a factor of two.

In the low pressure plasmas (<20mTorr) studied here, gas-phase recombination reactions of neutral atoms and radicals are very slow. As a result, diffusion to the chamber walls, where recombination takes place, is the dominant loss process for neutral species. The wall composition in a typical BF_3 discharge consists of boron and fluorine atoms adsorbed at the active sites on the reactor walls. Active sites are usually imperfections, dislocations etc. in the wall surface. Kota et al. [27] have shown that an incident flux of neutrals can either reflect without adsorption or physisorb into a weakly bound state. Physisorbed species can diffuse along the reactor surface and chemisorb at an active site. Physisorbed species can either desorb without recombining or react with chemisorbed species and then desorb. Because the neutral to ion flux ratio is small and

the surface temperature is low, the thermal desorption component is assumed to be negligible.

2.3.1 Assumptions

The surface chemistry model assumptions for BF_3 are:

1. All neutral species physisorb on the wall with the same sticking coefficient. A value γ=0.62 was found to best fit the model to experimental data. Also it is assumed that all ions fragment on impact with the reactor wall and do not stick to the wall, i.e. γ=0 for all ions.
2. The surface acts as a sink for each neutral species diffusing to the wall at a rate given by (2.2). Thus the volumetric rate of loss of boron atoms at the wall is given by

$$F_{in,B} = \sum_g k_{wg} n_g B_g \tag{2.3}$$

where, n_g is the neutral species density, B_g is the number of boron atoms in the diffusing neutral species, and k_{wg} is given by (2.2). Similarly for fluorine,

$$F_{in,F} = \sum_g k_{wg} n_g F_g . \tag{2.4}$$

3. The surface acts as a source, producing 100% BF_3. In other words, all boron-containing neutrals diffusing to the wall are recycled back into the plasma as BF_3. The volumetric rate of production of BF_3 is given by

$$F_{out,BF_3} = F_{in,B} \tag{2.5}$$

and by conservation of particles,

$$F_{out,F} = F_{in,F} - 3F_{in,B} . \tag{2.6}$$

It should be noted that $F_{out,F}$ can be negative. This only implies that the surface acts as a net sink for F atoms.

4. Due to lack of data for various ion species, we assume that all ions hitting the surface are fragmented into neutral atomic components and recycled back into the discharge; i.e, BF^+ ions hitting the reactor wall are recycled back as B and F atoms. The bond dissociation energies (i.e, the energy to free one F atom) for BF, BF_2, and BF_3 are 8.1 V, 5.9 V, and 5.8 V respectively [28] which are considerably smaller than the ion impact energy of ~30 eV. A study done by Lau and Hildenbrand [29] on BF_2 reveals that the FB-F bond is weaker than both B-F and F_2B-F bonds.

The model described above is consistent with a Langmuir-Hinshelwood site model assuming that the surface active sites are saturated with chemisorbed B and F atoms, and that ion bombardment dominates the production and desorption of neutral BF_3. For such a model, the fractional boron coverage is given by

$$\theta_B = \frac{F_{in,B}}{F_{in,B} + Y_i \sum_i k_i n_i B_i} \tag{2.7}$$

where Y_i is a yield factor (BF_3 neutrals produced per ion) [30], k_i is the rate constant [s^{-1}] for ion bombardment at the surface, n_i is the concentration of the i-th kind of ion and B_i is the boron composition of i-th kind of ion.

2.4 Surface Magnetic Confinement

The source geometry used in this model is that of the Eaton ULE2 ion source. The source is a cylinder of length L=20 cm and radius R=10 cm. Magnetic confinement is provided by a 20 cusp multipole field at the circumferential perimeter and at one axial end of the cylinder, with a field

strength of 2.5 kG at the cusps.

Figure 2.1 shows the magnetic confinement geometry. Ion and electron trajectories are altered due to the influence of the surface magnetic field, which reflects some charged particles back into the discharge. Hence surface magnetic confinement reduces the effective wall area for charged species.

Leung et al. [31, 32] found that hot electrons can be efficiently trapped at low pressures. These electrons can be the main source of ionization for species such as fluorine atoms whose ionization potential is 17.42 eV. The fraction f_{loss} of diffusing electron-ion pairs lost in the effective leak width of a line cusp can be written as [10]

$$f_{loss} = \frac{Nw}{2\pi R} \tag{2.8}$$

where N is the number of cusps, w is the effective leak width, and $Nw < 2\pi R$. The size of the leak width is not well understood. A heuristic formula for the effective leak width at the circumferential wall is

$$w = 4\sqrt{r_{ce} r_{ci}}\left(1 + \frac{R}{N\sqrt{\lambda_e \lambda_i}}\right). \tag{2.9}$$

Here r_{ce} and r_{ci} are the electron and ion gyro radii, and λ_e and λ_i are the electron and ion mean free paths, respectively. The effective leak width is observed to increase with pressure. The form of leak width in (2.9) is valid for low and intermediate pressures and it goes to the correct limits presented by Hershkowitz et al. [33] and Mattthieussent and Pelletier [34].

The diffusion loss of electron-ion pair requires a modified scaling of density ratios at the circumferential sheath edge:

$$h_R^w = \frac{n_{sR}}{n_0} = \frac{0.8 f_{loss}}{\left(0.64 + \left(3.36 + \frac{R}{\lambda_i}\right) f_{loss}\right)^{1/2}} \tag{2.10}$$

We use a similar expression at the axial wall of the source

$$h_L^w = \frac{n_{sL}}{n_0} = \frac{0.86 f_{loss}}{\left(0.74 + \left(2.26 + \frac{L}{2\lambda_i}\right) f_{loss}\right)^{1/2}}. \tag{2.11}$$

Here n_{sR} and n_{sL} are the sheath edge densities at the radial and axial edges of the source respectively. For $f_{loss} \to 1$ (no magnetic confinement), (2.10) and (2.11) reduce to form presented by Godyak et al. [21]. For $f_{loss} \to 0$ (no diffusional losses to the wall), both (2.10) and (2.11) reduce to zero. The effective area for ions striking the chamber walls is then given by

$$A_{eff} = \pi R(Rh_L + Rh_L^w + 2Lh_R^w). \tag{2.12}$$

A comparison of (2.12) with (1.34) shows that (2.12) is smaller, and hence the surface loss component (1.33) of the generalized power balance equation is also reduced.

2.5 Plasma Chemistry

The model incorporates BF_3, BF_2, BF, B, F, BF_3^+, BF_2^+, BF^+, B^+, B^{++}, and F^+. For each neutral and charged species, a particle balance equation is developed that accounts for creation and destruction processes for each species. The reaction set used for BF_3 is listed in Table 2.1 and the excitation energy loss processes are listed in Table 2.2. The types of electron-neutral reactions included are direct ionization, dissociative ionization, dissociation, momentum transfer

and excitation. Negative ions and metastables were not included, as their effects are negligible in the pressure and power regime of interest, except possibly for metastable F atoms. Two step ionization, i.e. excitation from ground state to a metastable state followed by ionization from metastable state, was not considered for any of the species.

The appearance potential of BF_2^+ in reaction 2 is 15.76 V which is only 0.2 V higher than BF_3^+ in reaction 1. Farber and Srivastava [35] have shown in their study of BF_3 that reaction 2 dominates reaction 1 at electron impact energies above 18 eV [35]. Farber and Srivastava [35] have also shown that the reactions

$$e + BF_3 \rightarrow BF_2 + F^+ + 2e$$

and

$$e + BF_3 \rightarrow BF^+ + 2F + 2e$$

both have a very high threshold (30 eV and 24 eV respectively) and thus they have been neglected.

Since no experimental data was available for the electron impact dissociation of BF_2 and BF (reactions 5 and 7 respectively), the reaction rate was estimated by integrating a classical cross-section over a maxwellian distribution over an electron temperature range of 1-8 eV. A classical estimate of the dissociation cross-section is given by [10]

$$\sigma_{diss}(E) = \begin{cases} 0, & E < E_1 \\ \sigma_0 \dfrac{E - E_1}{E_1}, & E_1 < E < E_2 \\ \sigma_0 \dfrac{E_2 - E_1}{E}, & E > E_2 \end{cases} \tag{2.13}$$

where,

$$\sigma_0 = \pi \left(\frac{e}{4\pi\varepsilon_0 E_1} \right)^2 . \tag{2.14}$$

Here E_1 is the dissociation energy and E_2 is the ionization energy of the molecule. It should be noted that the cross-section rises linearly for $E_1 < E < E_2$ and then falls off as $1/E$ for $E > E_2$. A similar rate constant was computed for the dissociation of BF_3 (reaction 3), for which data is available. Comparing the classical estimate from (2.13) with the data, a normalization factor was obtained. The rate constants calculated for reactions 5 and 7 were then normalized using this factor.

The double ionization of atomic boron (reaction 9) was estimated using the classical Thomson cross-section given by [10]

$$\sigma_{iz}(E) = \begin{cases} 0, & E < E_{iz} \\ \pi \left(\dfrac{e}{4\pi\varepsilon_0} \right)^2 \dfrac{1}{E} \left(\dfrac{1}{E_{iz}} - \dfrac{1}{E} \right), & E > E_{iz} \end{cases} \tag{2.15}$$

The B^{++} production (reaction 9) rate constant was calculated by integrating (2.15) over a maxwellian distribution. A similar computation was performed for production of B^+ (reaction 8). The resulting rate constant was then compared to its experimental counterpart [36], and a normalization factor was obtained. The rate constant for reaction 9 was then normalized using

this factor.

Various sources were compiled in order to determine the momentum transfer rate constant of atomic fluorine. Robb and Henry [37] calculated the elastic momentum transfer cross-section for very low electron impact energy in the range of 2.7-27.2 meV. In the energy range 0.07-12.2 eV, the cross-sectional data produced by Robinson and Geltman [38] was used. Above 12 eV, the scattering cross-section was assumed to decrease inversely with the electron energy. The final rate constant was obtained by integrating the cross-section over an assumed maxwellian distribution.

Only BF_3 feed-gas is introduced into the reaction chamber, all neutral and charged ion species are assumed to be pumped away at the same pumping rate (which is fixed for a given flow-rate and pressure). The neutral species diffusing towards the reactor wall are being pumped out of the discharge by a "surface sink" and they are recycled back into the discharge as BF_3 by a "surface source". The "surface sink" for neutrals is given by (2.2) and the "surface source" for BF_3 is given by (2.5).

The primary BF_3 reaction set (Table 2.1) is limited to the smallest reasonable set of reactions. It is, therefore, very interesting to study the production sequence of various species. In the operating regime of interest (high power, low pressure), the discharge tends to be highly dissociated. BF_3 is fed into the reactor as a feed-gas. The present surface model recycles all the surface bound neutral species back into the discharge as BF_3. The main production mechanism

for BF_2^+ ions was found to be reaction 2. Ionization is the only process that produces boron ions (singly and doubly ionized). Therefore, the B^+ and B^{++} ion concentrations are directly linked to the neutral boron density in the discharge. Atomic boron is produced by ions as they strike the wall and fragment into their neutral components. Production of fluorine atoms is by reactions 2, 3, 5, and 7. Ionization of atomic fluorine (reaction 10) is the sole process by which fluorine ions are produced. The BF^+ ion is produced exclusively by ionization of BF (reaction 6). Both F^+ and BF^+ ions are destroyed exclusively at the surface.

In the excitation energy loss reactions, only electronic interactions with neutral species were considered. The loss processes included in the model are excitation, momentum transfer, direct ionization, dissociative ionization, and vibrational excitation. The reaction $e + F \rightarrow F^* + 2e$ is the total excitation rate for transitions to the 10 lowest excitation states in the energy range from 16.3 to 41 eV. The threshold value of 12.7 eV is the energy for the transition from ground state to the next highest energy state. The rates were calculated using theoretical estimations given by Baliyan and Bhatia [39].

A more complete reaction set is listed in Table 2.3. It includes electron-ion recombination reactions and charge transfer reactions between various ion species. However, no data is available for these reactions, and only simple classical estimates were used [10], in un-normalized for, to determine the rate constants. Therefore we do not present any results for this reaction set.

2.6 Results and Discussion

There are a total of six ion species included in the model. Figures 2.2-2.5 show a comparison of the model with experimental data at various powers. All pressures in these figures are those obtained in the absence of plasma. The experimental data was obtained from the Eaton ULE2 ion source. The present surface model is the simplest model that agrees well with the experimental data. Using a common sticking coefficient for all neutral species γ=0.62 gives us a best fit to experimental data. In the lower range of input powers investigated, the model is in excellent agreement with experimental data (see figures 2.2 and 2.3) and the essential trends in fractional ion composition are well captured.

At very low pressures (<1 mTorr) and high powers (>900 Watts), the model does not accurately predict the behavior of B^+ and F^+ species (see figures 2.4 and 2.5). The F^+ concentration does not appear to increase at the same rate as the experimental data. One possible explanation for this is that atomic fluorine may have an important two step ionization process involving a metastable species. Ashida et al. [7] have shown that in case of argon roughly 13% of the argon ions are produced from the argon metastable at a pressure 5 mTorr and an electron density of ~2 x 10^{17}. At very low pressures, it is possible that the fluorine metastable makes a significant contribution to the total fluorine ion density. Further, experimental data shows that the B^+ concentration peaks at roughly 1mTorr and decreases at lower pressures. Figure 2.6 shows the variation in electron temperature at very low pressures. The lowest pressure investigated was 0.1 mTorr. The electron temperature increases significantly in the pressure range 0.1-0.5 mTorr. It was observed that increasing the input power at such low pressure further increased the electron temperature. A

higher electron temperature should also increase the F^+ concentration as more of higher energy electrons are now available to participate in the ionization process. An increase in F^+ concentration would correspondingly reduce the fractional B^+ concentration. Figure 2.7 shows the fractional ionization of various species. The input power was 1500 watts and the lowest reactor pressure investigated was 0.1 mTorr. Under these conditions, it should be noted that almost all the boron (roughly 80%) is ionized. At this point, the F^+ concentration is also increasing significantly with decreasing pressure.

The model predicts that the BF_3^+ and B^{++} concentrations are very low. This agrees with the experimental evidence which shows BF_2^+, BF^+, B^+, and F^+ as the only ions present in the plasma in significant concentrations. The model does predict a drop in B^+ fractional ion flux, but at 0.1 mTorr, rather than experimentally observed at ~1 mTorr. Perhaps using different values of sticking coefficients for each neutral species would result in a better agreement with the experimental data.

Figure 2.8 compares the effect of magnetic confinement with an unconfined source geometry. Since no negative ions are considered in the model, the electron density is identically equal to the total positive ion density. Magnetic confinement reduces the effective area (2.12) and this leads to lower losses at the walls. Thus under the identical conditions, magnetic confinement increases the total positive ion density in the discharge especially at low pressures.

In reaction 3, the actual threshold value (10.1 eV) was much higher than the dissociation energy of F_2B-F bond (5.8 eV) but it was lower than the ionization energy (~15.6 eV). Thus two different threshold energies were investigated for both reactions 5 and 7. The higher energy values were arbitrarily chosen to be 2 eV higher than the respective dissociation energies but lower than the ionization threshold. Since no data were available for electron impact dissociation of BF_2 and BF, the reaction rates were estimated using a classical (Thomson) cross-section, as described previously. Figures 2.9 and 2.10 show the effect of using two different threshold values for dissociation reactions 5 and 7. It should be noted that for all future discussion of the model, the lower threshold energy values have been used.

Figure 2.11 shows that the electron density decreases slightly with pressure for various input powers. Figure 2.12 shows that the plasma tends to be slightly more dissociated as power is increased at a fixed low pressure. Fractional ionization of the discharge was nearly 1% at 0.5 mTorr and higher fractional ionization was observed for higher powers (see figure 2.13). Figures 2.14-2.24 show the variations in the concentration of various species with pressure at different input powers. The BF concentration in figure 2.16 shows a peculiar behavior at higher pressures. The BF density appears to peak at high pressures (~2 mTorr).

Figure 2.17 shows that the boron density falls off slightly with increasing pressures for all powers investigated. B^{++} concentration is very low for a wide range of pressure and power. The pressure variation of B^{++} concentration is shown in figure 2.23. The B^{++} concentration varies steeply with inverse pressure, with $n_{B^{++}} \propto P^{-3}$. The negligible fraction of B^{++} can be attributed to

the high second ionization potential (33.7 volts) of boron. The electron temperature decreases with increasing pressure as shown in figure 2.25. It drops from ~3 V to ~2 V over 0.5-5 mTorr. But as shown in figure 2.6, the electron temperature variation at very low pressure (0.1 mTorr) is quite drastic. It jumps from ~3 V at 0.5 mTorr to ~5 V at 0.1 mTorr. Figures 2.26-2.36 show the fractional composition of each species.

Figure 2.37 shows the increase in electron density with power, we find approximately that $n_e \propto$ power. As mentioned earlier, using sticking coefficient γ=0.5 gave us the best fit to experimental data. An investigation of different values of the sticking coefficient (γ) showed that increasing γ had the same effect as decreasing power. In other words, increasing the value of γ led to an increase in the neutral flux to the surface. Since the surface recycles all species back into the discharge as BF_3, an increase in neutral flux leads to increase the BF_3 concentration and thus the plasma becomes less dissociated. Figure 2.38 shows that decreasing the power at low pressures also makes the plasma less dissociated. The fractional ionization of the plasma also increases with an increase in input power (see figure 2.39).

Figures 2.40-2.50 show the variation in the concentration of various species with input power. The electron temperature varies little over a wide range of power as shown in figure 2.51. Figures 2.52-2.62 show the fractional composition of various species as a function of input power.

Table 2.1 BF_3 Primary Reaction Set

Reaction	Rate Coefficient[a] [m^3/s]	Ref.
1. $e + BF_3 \rightarrow BF_3^+ + 2e$	$k_1 = 1.03\times10^{-15} T_e^{0.44} \exp(-15.37/T_e)$	[36]
2. $e + BF_3 \rightarrow BF_2^+ + F + 2e$	$k_2 = 6.7\times10^{-15} T_e^{1.06} \exp(-15.96/T_e)$	[36]
3. $e + BF_3 \rightarrow BF_2 + F + e$	$k_3 = 2.68\times10^{-14} T_e^{0.35} \exp(-10.46/T_e)$	[36]
4. $e + BF_2 \rightarrow BF_2^+ + 2e$	$k_4 = 2.23\times10^{-15} T_e^{1.37} \exp(-8.37/T_e)$	[40]
5. $e + BF_2 \rightarrow BF + F + e$	$k_5 = 1.33\times10^{-13} T_e^{-0.41} \exp(-6.768/T_e)$ $k_5 = 5.23\times10^{-14} T_e^{-0.48} \exp(-8.488/T_e)$	[10][b]
6. $e + BF \rightarrow BF^+ + 2e$	$k_6 = 9.58\times10^{-15} T_e^{0.82} \exp(-9.62/T_e)$	[40]
7. $e + BF \rightarrow B + F + e$	$k_7 = 3.73\times10^{-14} T_e^{-0.42} \exp(-8.969/T_e)$ $k_7 = 1.21\times10^{-14} T_e^{-0.49} \exp(-10.55/T_e)$	[10][b]
8. $e + B \rightarrow B^+ + 2e$	$k_8 = 2.63\times10^{-15} T_e^{1.41} \exp(-6.94/T_e)$	[41]
9. $e + B \rightarrow B^{++} + 3e$	$k_9 = 4.84\times10^{-15} T_e^{0.25} \exp(-33.7/T_e)$	(c)
10. $e + F \rightarrow F^+ + 2e$	$k_{10} = 1.3\times10^{-14} \exp(-16.5/T_e)$	[42]
11. $e + B^+ \rightarrow B^{++} + 2e$	$k_{11} = 9.41\times10^{-16} T_e \exp(-25.19/T_e)$	[41]

a. T_e in Volts.
b. Estimated value

Table 2.2 Excitation Energy Loss Reaction Set for BF_3

Reaction	Process	Energy [eV]	Rate Coefficient[a] [m^3/s]	Ref.
1. $e + BF_3 \rightarrow BF_3^+ + 2e$	Ionization	15.56	$k_1 = 1.03\times10^{-15} T_e^{0.44} \exp(-15.37/T_e)$	[36]
2. $e + BF_3 \rightarrow BF_2^+ + F + 2e$	Dissociative Ionization	15.76	$k_2 = 6.7\times10^{-15} T_e^{1.06} \exp(-15.96/T_e)$	[36]
3. $e + BF_3 \rightarrow BF_2 + F + e$	Dissociation	10.1	$k_3 = 2.68\times10^{-14} T_e^{0.35} \exp(-10.46/T_e)$	[36]
4. $e + BF_2 \rightarrow BF_2^+ + 2e$	Ionization	9.4	$k_4 = 2.23\times10^{-15} T_e^{1.37} \exp(-8.37/T_e)$	[40]
5. $e + BF_2 \rightarrow BF + F + e$	Dissociation	5.9 7.9	$k_5 = 1.33\times10^{-13} T_e^{-0.41} \exp(-6.768/T_e)$ $k_5 = 5.23\times10^{-14} T_e^{-0.48} \exp(-8.488/T_e)$	[10][b]
6. $e + BF \rightarrow BF^+ + 2e$	Ionization	11.12	$k_6 = 9.58\times10^{-15} T_e^{0.82} \exp(-9.62/T_e)$	[40]
7. $e + BF \rightarrow B + F + e$	Dissociation	8.1 10.1	$k_7 = 3.73\times10^{-14} T_e^{-0.42} \exp(-8.969/T_e)$ $k_7 = 1.21\times10^{-14} T_e^{-0.49} \exp(-10.55/T_e)$	[10][b]
8. $e + B \rightarrow B^+ + 2e$	Ionization	8.30	$k_8 = 2.63\times10^{-15} T_e^{1.41} \exp(-6.94/T_e)$	[41]
9. $e + B \rightarrow B^{++} + 3e$	Ionization	33.45	$k_9 = 4.84\times10^{-15} T_e^{0.25} \exp(-33.7/T_e)$	[10][b]
10. $e + F \rightarrow F^+ + 2e$	Ionization	17.42	$k_{10} = 1.3\times10^{-14} \exp(-16.5/T_e)$	[42]
11. $e + BF_3 \rightarrow BF_3(v=1) + e$	Vibrational Excitation	0.059	$k_{11} = 1.57\times10^{-15} T_e^{-0.41} \exp(-0.5662/T_e)$	[36]
12. $e + BF_3 \rightarrow BF_3(v=2) + e$	Vibrational Excitation	0.086	$k_{12} = 3.56\times10^{-15} T_e^{-0.35} \exp(-0.5559/T_e)$	[36]
13. $e + BF_3 \rightarrow BF_3(v=3) + e$	Vibrational Excitation	0.18	$k_{13} = 3.33\times10^{-14} T_e^{-0.21} \exp(-0.4195/T_e)$	[36]
14. $e + BF_3 \rightarrow BF_3 + e$	Momentum Transfer	$3\frac{m_e}{m_i}T_e$	$k_{14} = 6.87\times10^{-14} T_e^{0.44} \exp(-0.3893/T_e)$	[36]
15. $e + F \rightarrow F + e$	Momentum Transfer	$3\frac{m_e}{m_i}T_e$	$k_{15} = 1.15\times10^{-13} \exp(-1.93/T_e)$	[37]
16. $e + F \rightarrow F^* + 2e$	Excitation[c]	12.7	$k_{16} = 6.71\times10^{-15} T_e^{0.11} \exp(-14.02/T_e)$	[39]

a. T_e in Volts

b. Estimated value

c. F^* denotes the excited final state for the 10 lowest excited states of atomic fluorine.

Table 2.3 BF_3 Secondary Reaction Set

Reaction	Rate Coefficient[a] [m^3/s]
1. $e + BF_3^+ \rightarrow BF_2 + F$	$k_1 = 4.35\times10^{-12} T_e^{-0.5}$
2. $e + BF_2^+ \rightarrow BF + F$	$k_2 = 4.40\times10^{-15} T_e^{-1.44} \exp(-6.16/T_e)$
3. $e + BF^+ \rightarrow B + F$	$k_3 = 1.30\times10^{-14} T_e^{-1.47} \exp(-4.26/T_e)$
4. $B^{++} + BF_3 \rightarrow B^+ + BF_3^+$	$k_4 = 3.23\times10^{-16} (T_g/300)^{0.5}$
5. $B^{++} + BF_2 \rightarrow B^+ + BF_2^+$	$k_5 = 9.10\times10^{-16} (T_g/300)^{0.5}$
6. $B^{++} + BF \rightarrow B^+ + BF^+$	$k_6 = 6.87\times10^{-16} (T_g/300)^{0.5}$
7. $B^{++} + B \rightarrow B^+ + B^+$	$k_7 = 1.49\times10^{-16} (T_g/300)^{0.5}$
8. $B^{++} + F \rightarrow B^+ + F^+$	$k_8 = 3.00\times10^{-16} (T_g/300)^{0.5}$
9. $F^+ + BF_3 \rightarrow F + BF_3^+$	$k_9 = 1.57\times10^{-16} (T_g/300)^{0.5}$
10. $F^+ + BF_2 \rightarrow F + BF_2^+$	$k_{10} = 4.49\times10^{-16} (T_g/300)^{0.5}$
11. $F^+ + BF \rightarrow F + BF^+$	$k_{11} = 3.49\times10^{-16} (T_g/300)^{0.5}$
12. $F^+ + B \rightarrow F + B^+$	$k_{12} = 8.13\times10^{-16} (T_g/300)^{0.5}$
13. $BF_3^+ + BF_2 \rightarrow BF_3 + BF_2^+$	$k_{13} = 3.12\times10^{-16} (T_g/300)^{0.5}$
14. $BF_3^+ + BF \rightarrow BF_3 + BF^+$	$k_{14} = 2.61\times10^{-16} (T_g/300)^{0.5}$
15. $BF_3^+ + B \rightarrow BF_3 + B^+$	$k_{15} = 6.99\times10^{-16} (T_g/300)^{0.5}$
16. $BF_2^+ + B \rightarrow BF_2 + B^+$	$k_{16} = 7.17\times10^{-16} (T_g/300)^{0.5}$
17. $BF^+ + BF_2 \rightarrow BF + BF_2^+$	$k_{17} = 3.86\times10^{-16} (T_g/300)^{0.5}$
18. $BF^+ + B \rightarrow BF + B^+$	$k_{18} = 7.57\times10^{-16} (T_g/300)^{0.5}$

a. T_e in Volts and T_g in Kelvins.

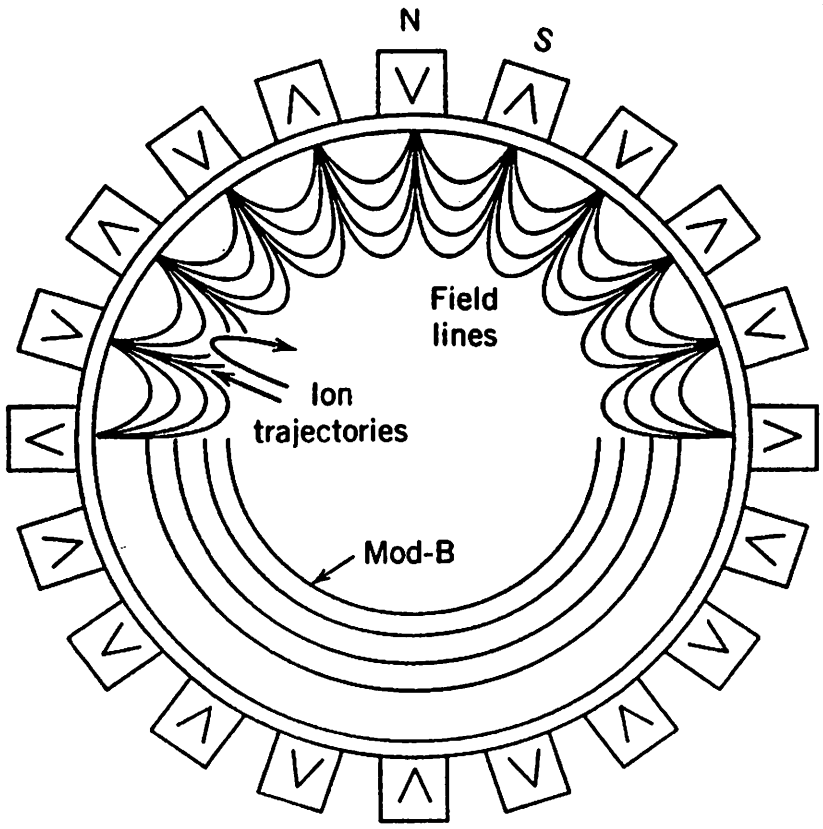

Figure 2.1: Illustration of magnetic field lines and the |**B**| surfaces near the circumferential walls.

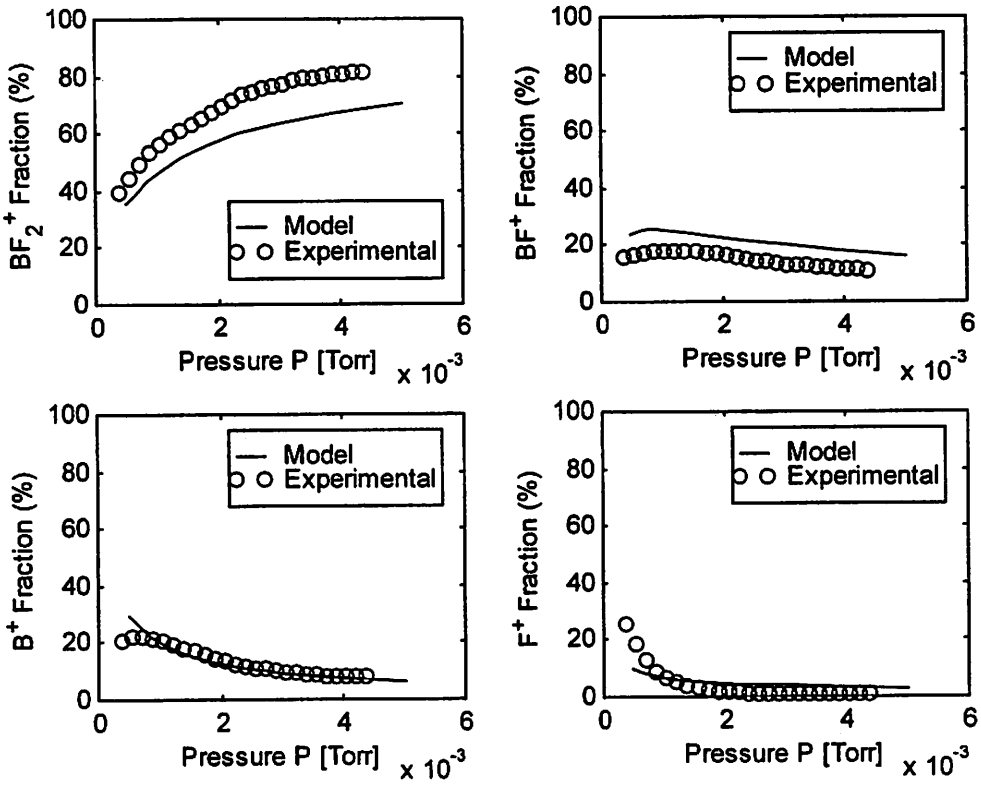

Figure 2.2: Ion flux fractions as a function of reactor pressure. Comparison is made at input power of 700 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

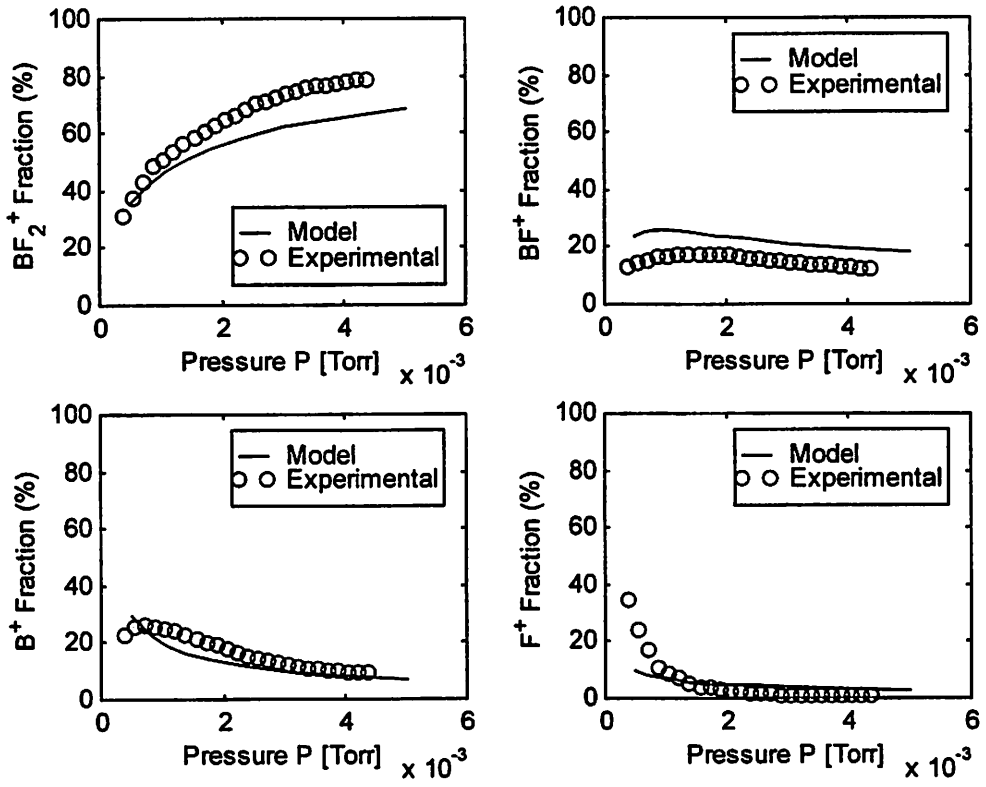

Figure 2.3: Ion flux fractions as a function of reactor pressure. Comparison is made at input power of 900 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

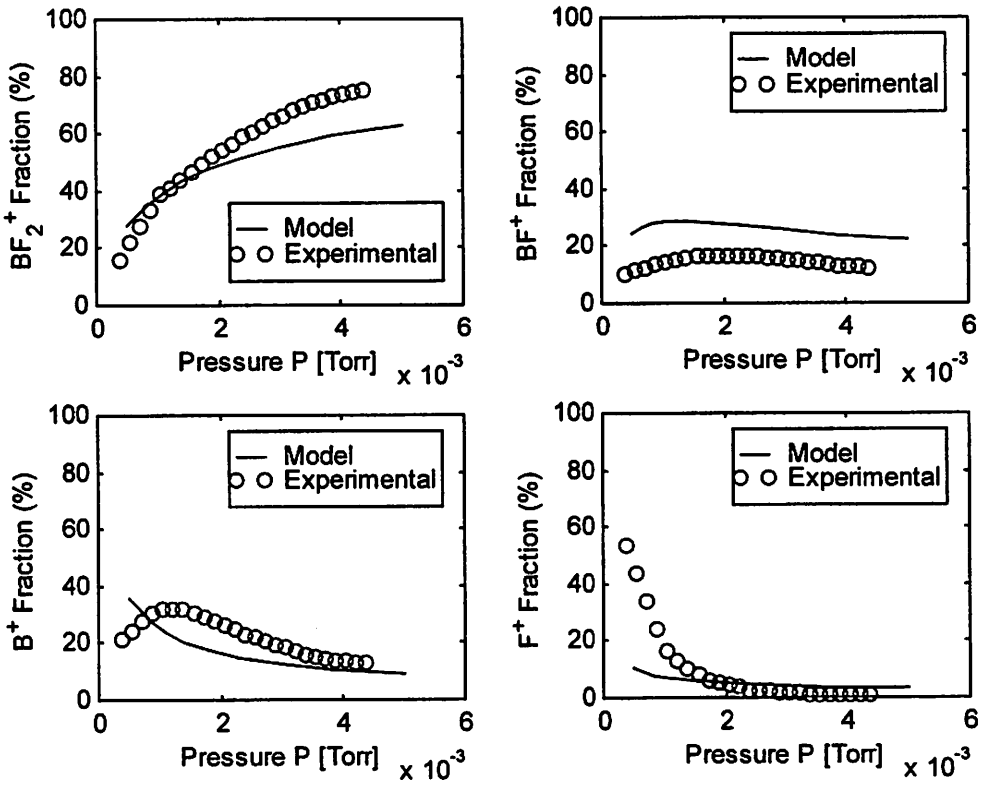

Figure 2.4: Ion flux fractions as a function of reactor pressure. Comparison is made at input power of 1200 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

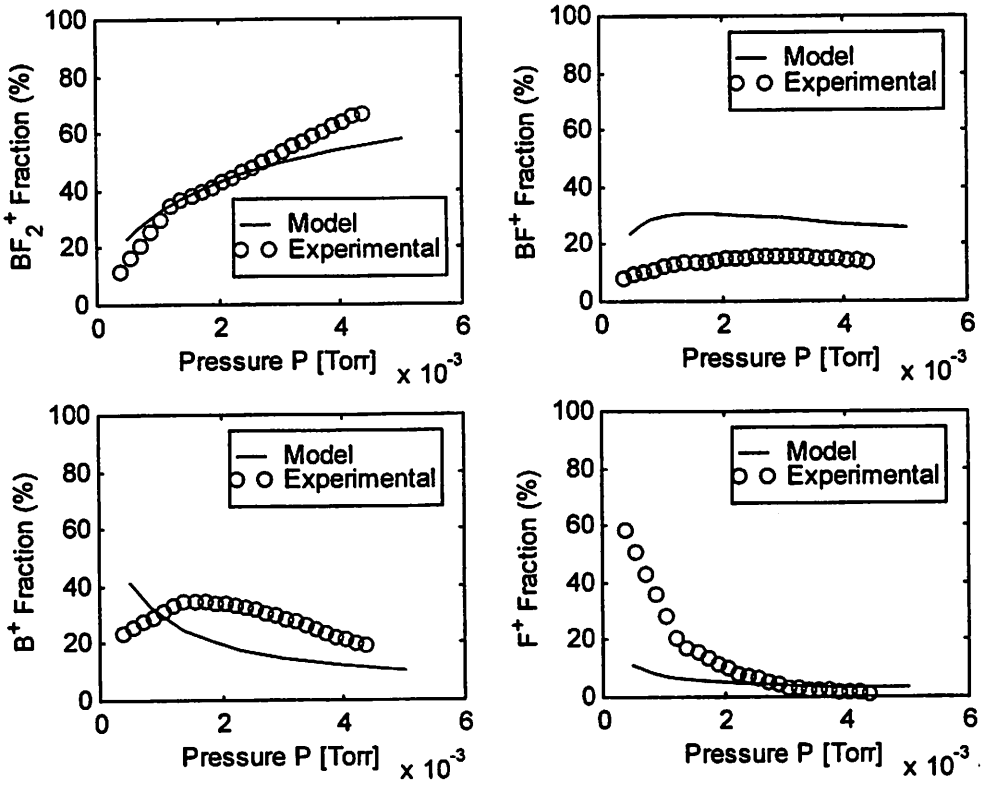

Figure 2.5: Ion flux fractions as a function of reactor pressure. Comparison is made at input power of 1500 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

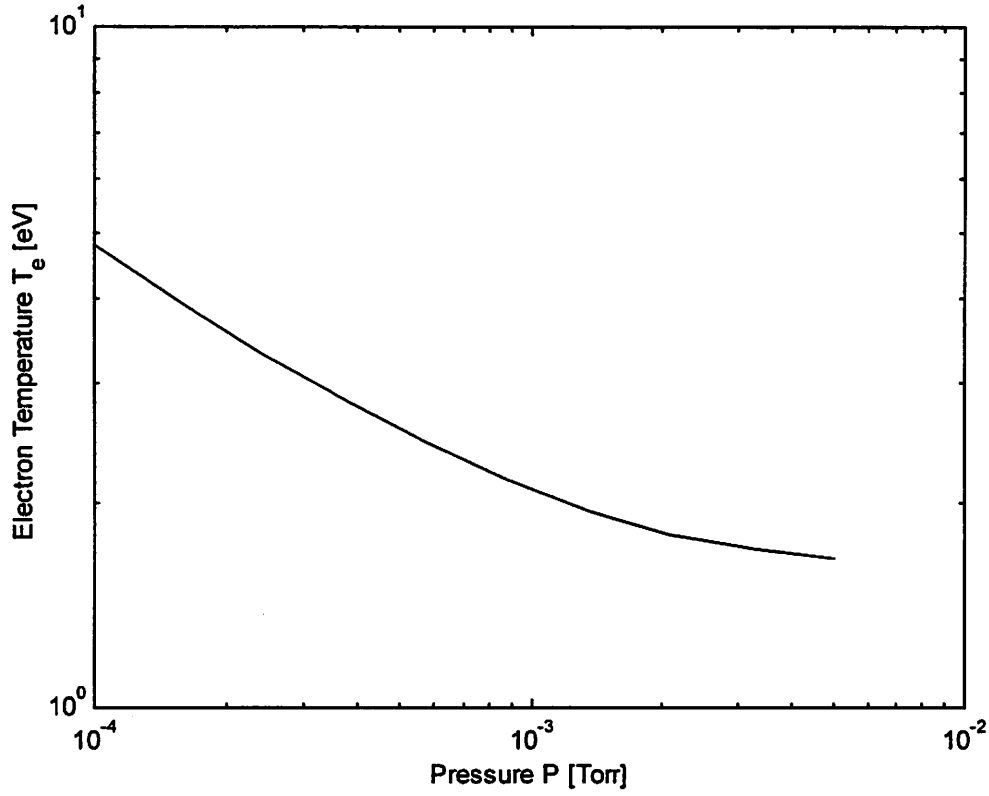

Figure 2.6: Electron temperature as a function of reactor pressure. Input power is 1500 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

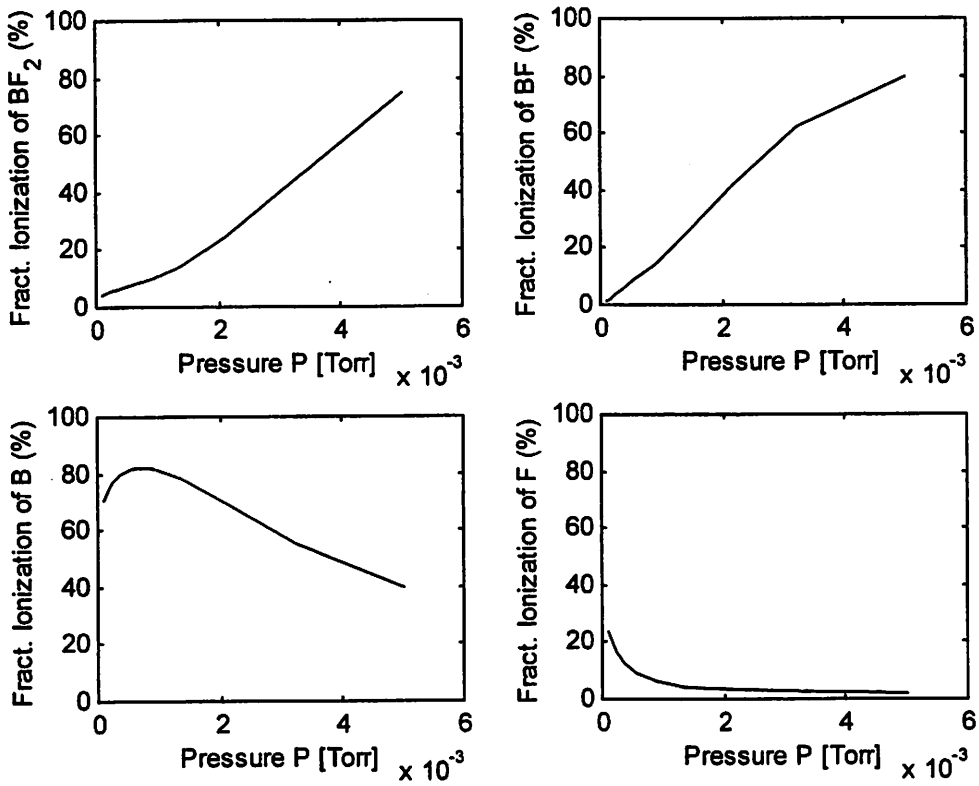

Figure 2.7: Fractional ionizations as a function of reactor pressure. Input power is 1500 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

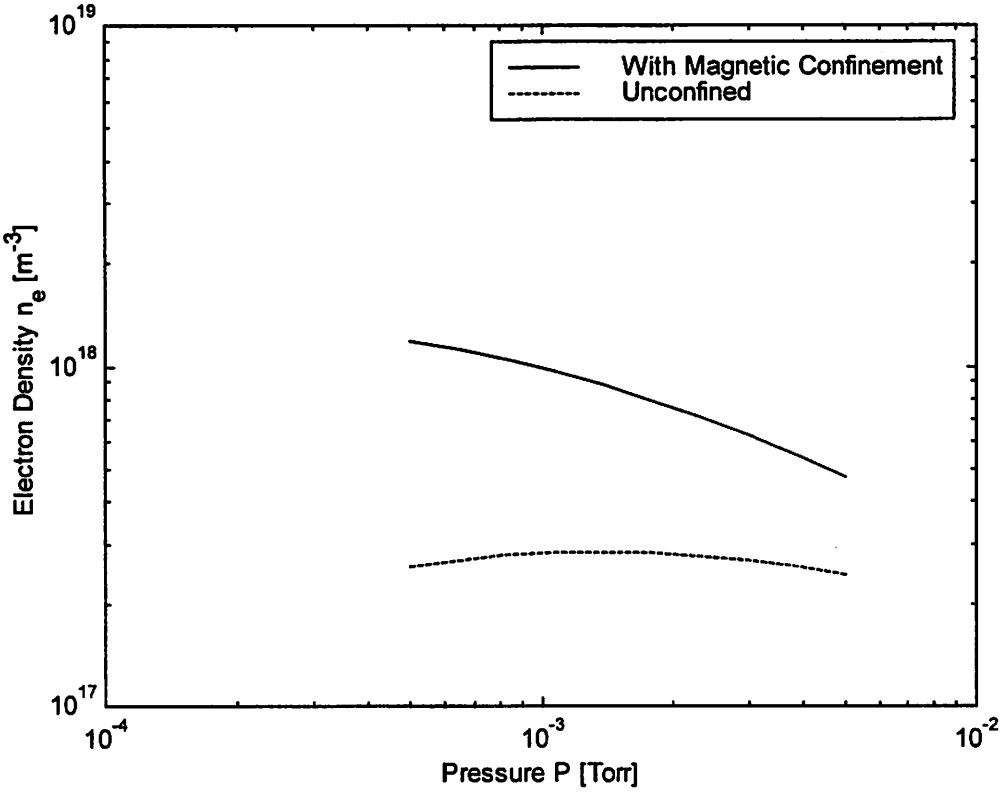

Figure 2.8: Electron density as a function of reactor pressure. Effect of magnetic confinement (20 cusp, 2.5 kG cusp strength) is shown. The electron density equals the total positive ion density as a result of quasi-neutrality condition. The input power is 700 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

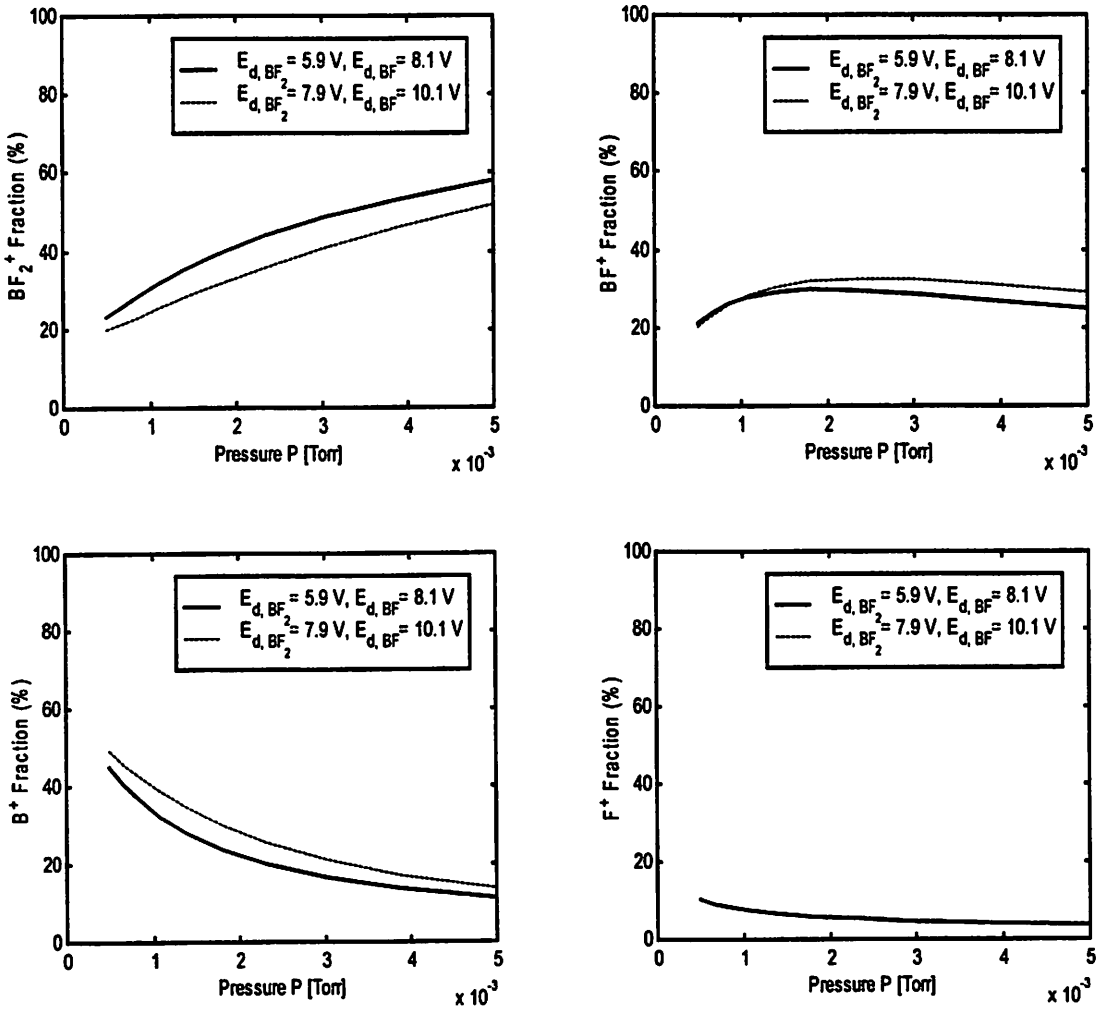

Figure 2.9: Ion flux fractions as a function of reactor pressure. The figure depicts the effect of using slightly higher threshold energy for the dissociation of BF and BF_2 molecules. The F^+ ion flux fraction is for both cases is virtually the same. The input power used is 1500 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

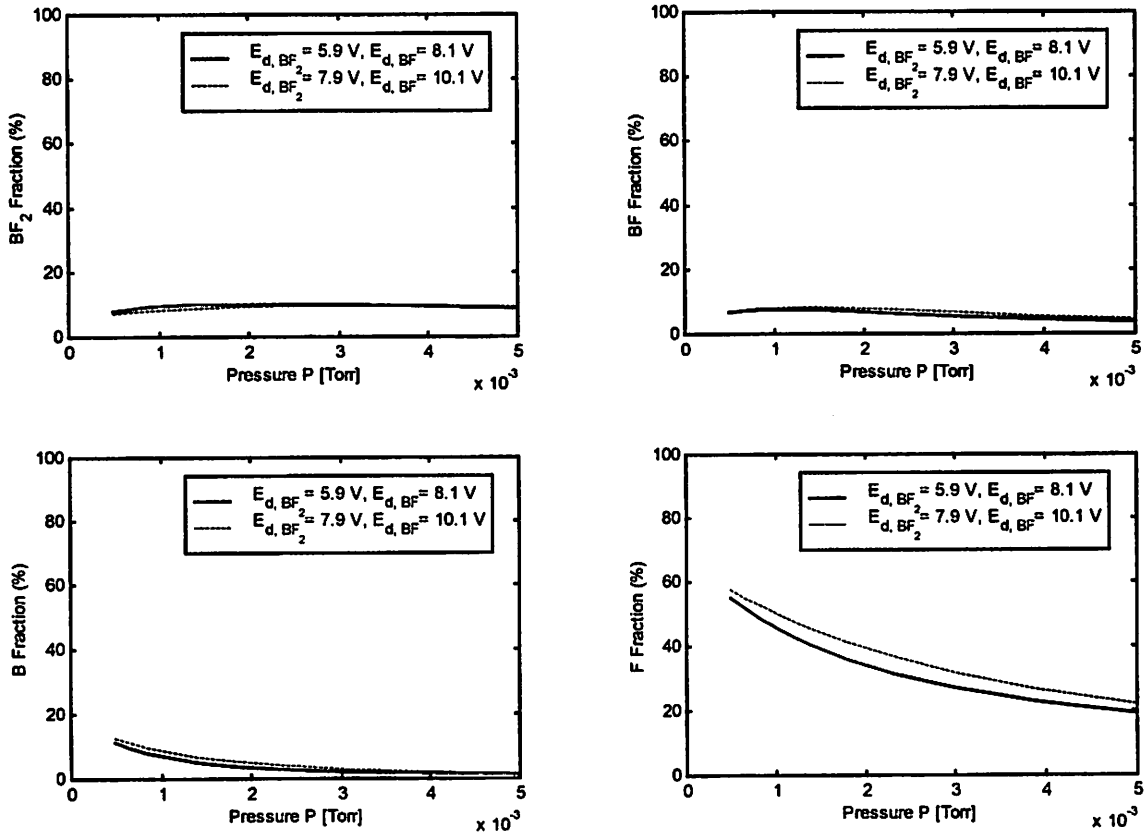

Figure 2.10: Ion flux fractions as a function of reactor pressure. The figure depicts the effect of using slightly higher threshold energy for the dissociation of BF and BF_2 molecules. The F^+ ion flux fraction is for both cases is virtually the same. The input power used is 1500 watts and the pumping is fixed to give 1mTorr at 1.5 SCCM.

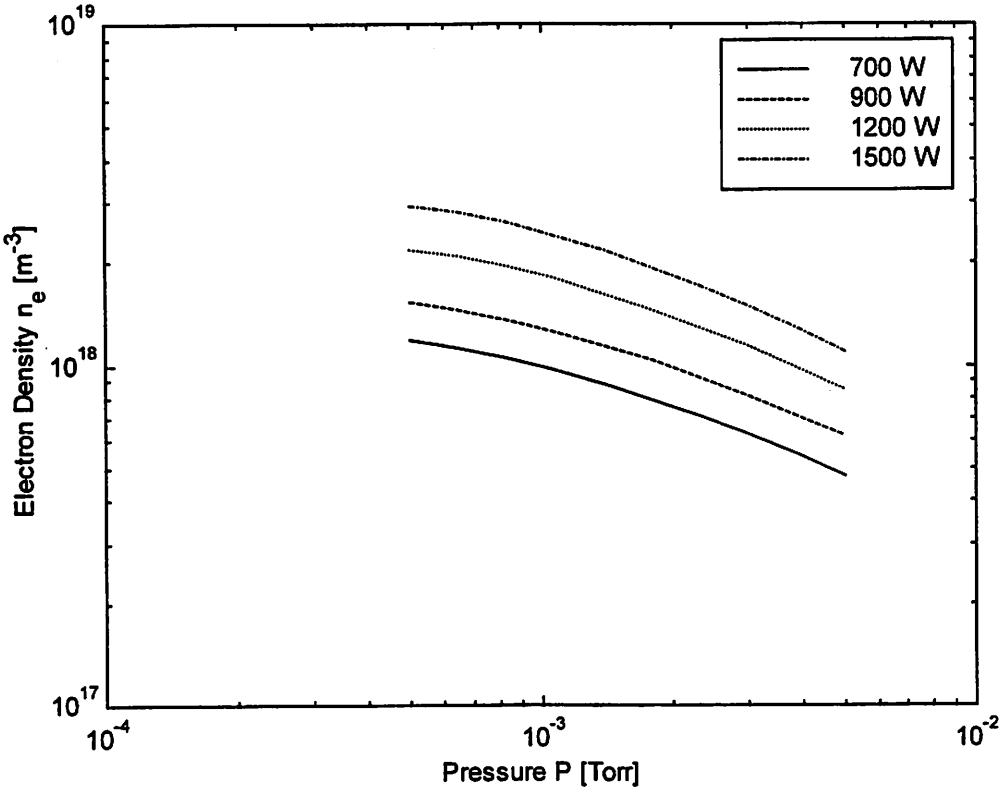

Figure 2.11: Electron density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

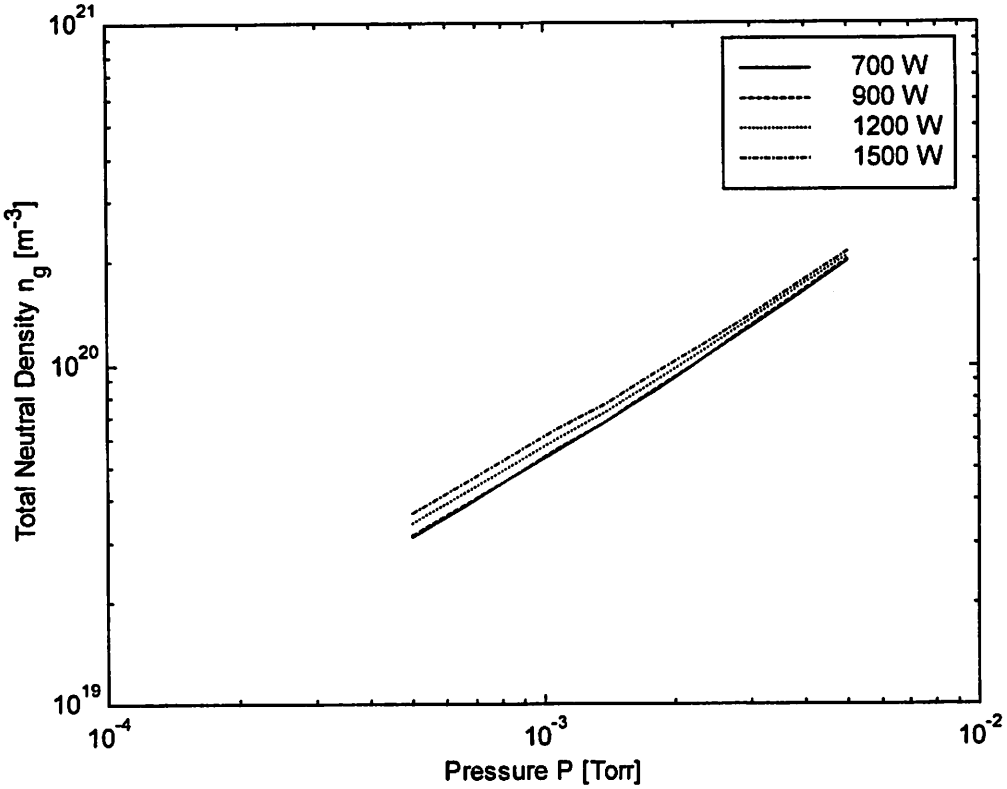

Figure 2.12: Total neutral density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

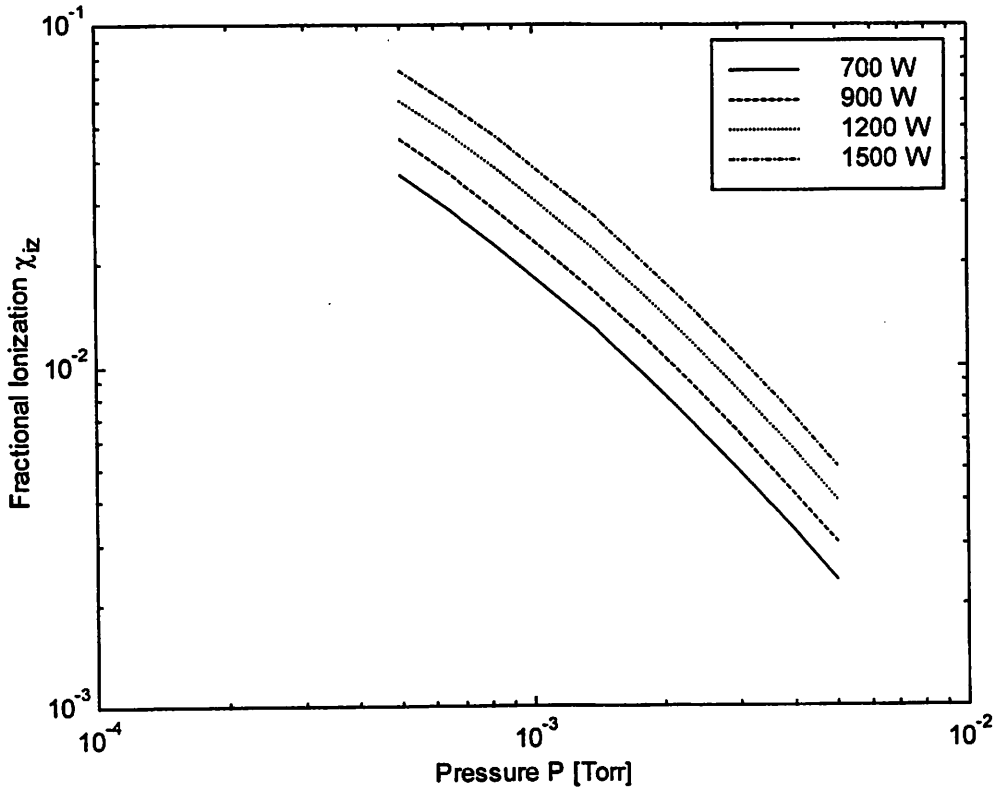

Figure 2.13: Fractional ionization as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

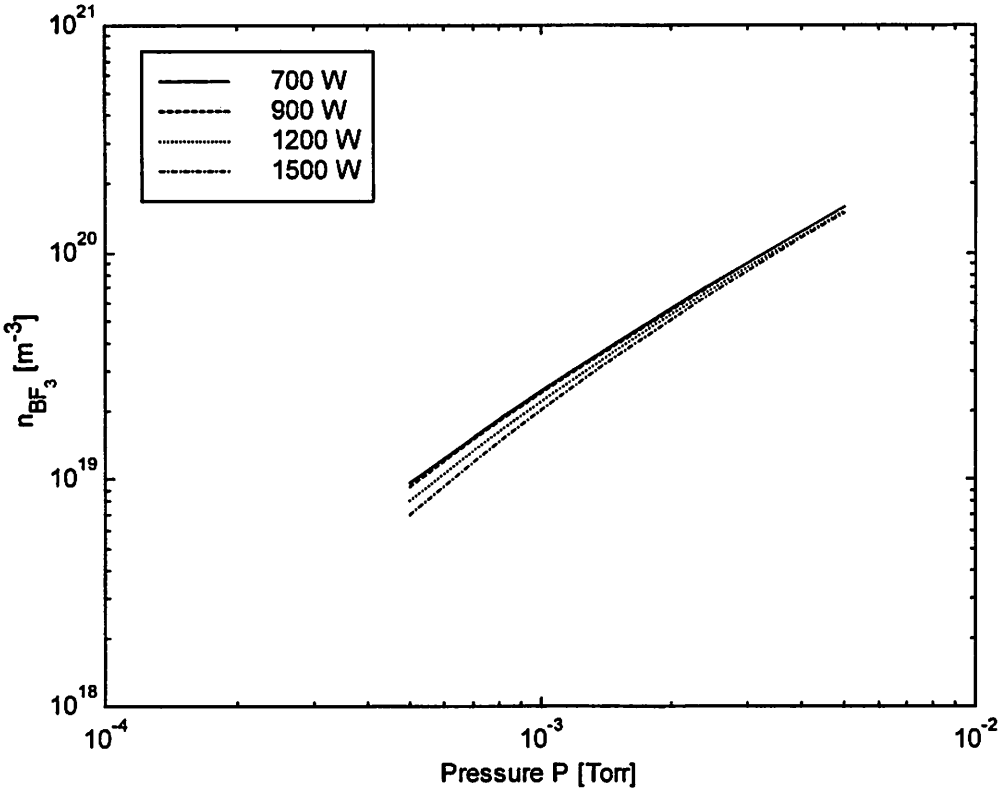

Figure 2.14: BF_3 density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

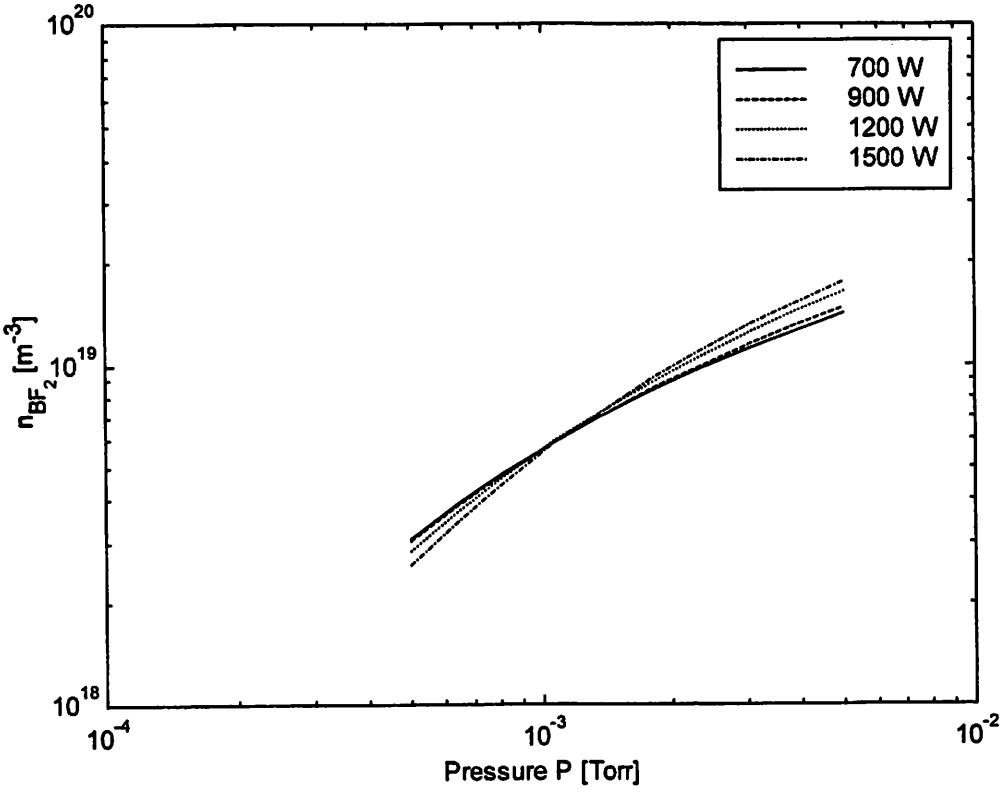

Figure 2.15: BF_2 density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

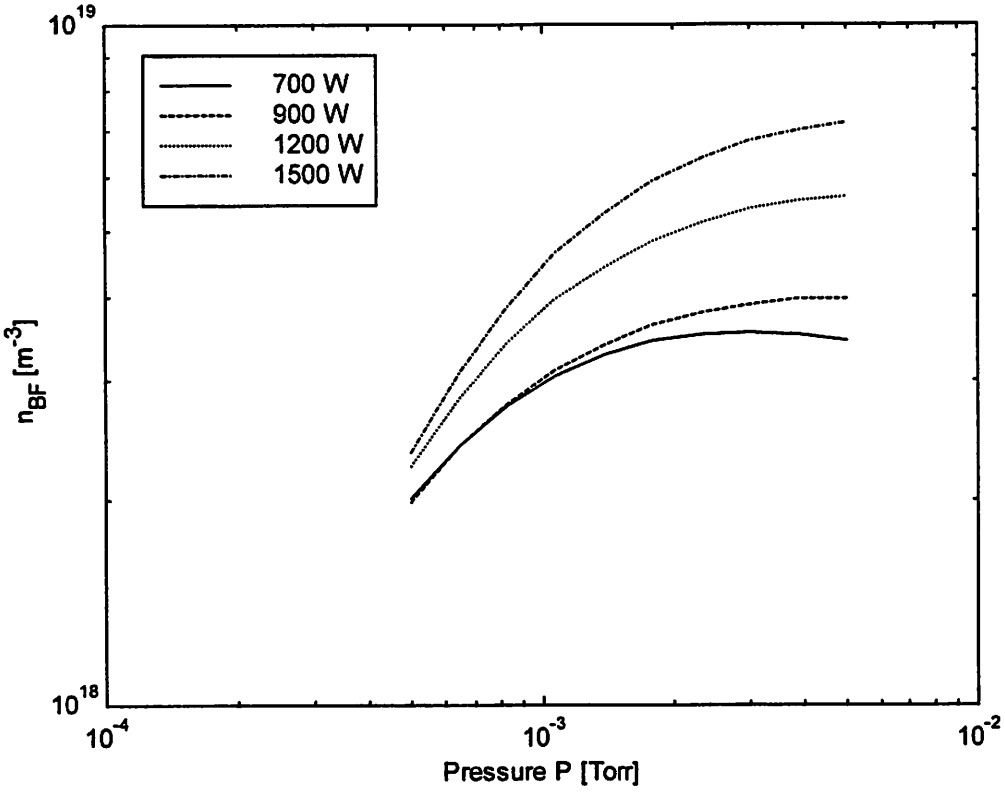

Figure 2.16: BF as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

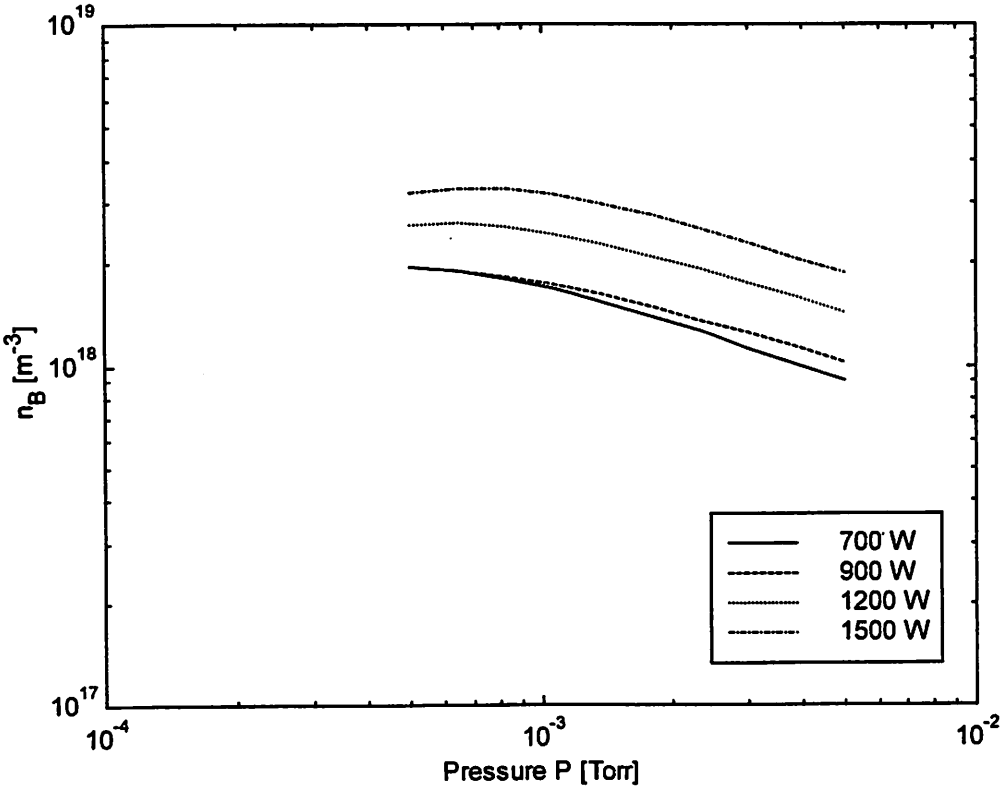

Figure 2.17: Boron density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

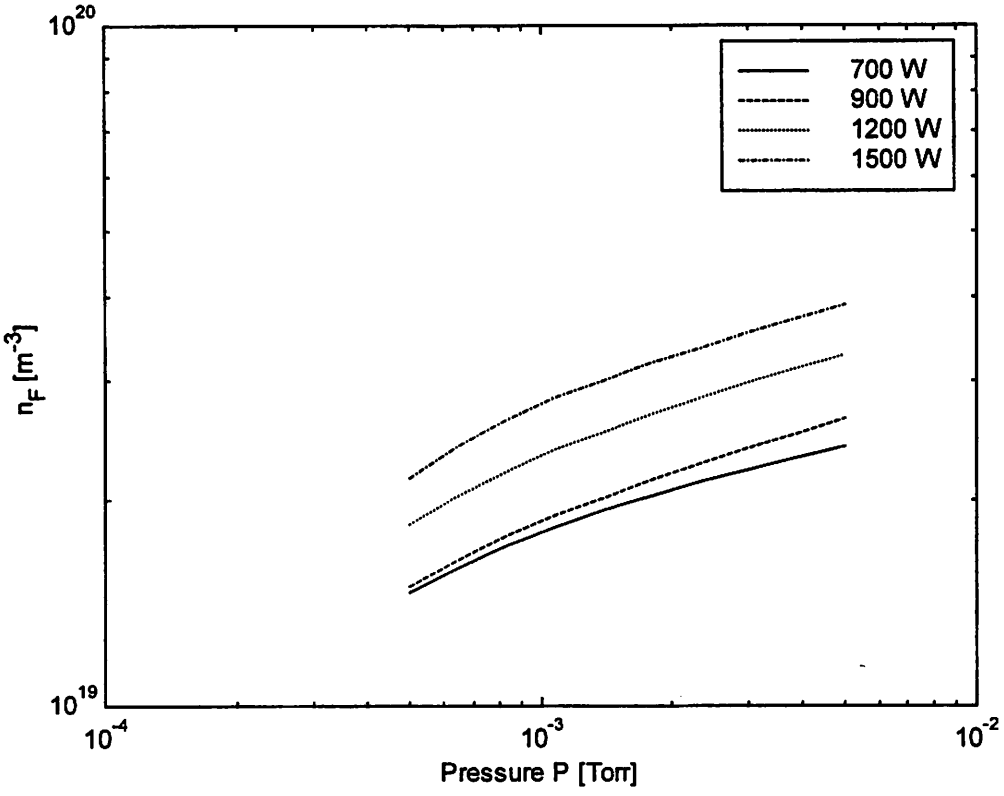

Figure 2.18: Fluorine density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

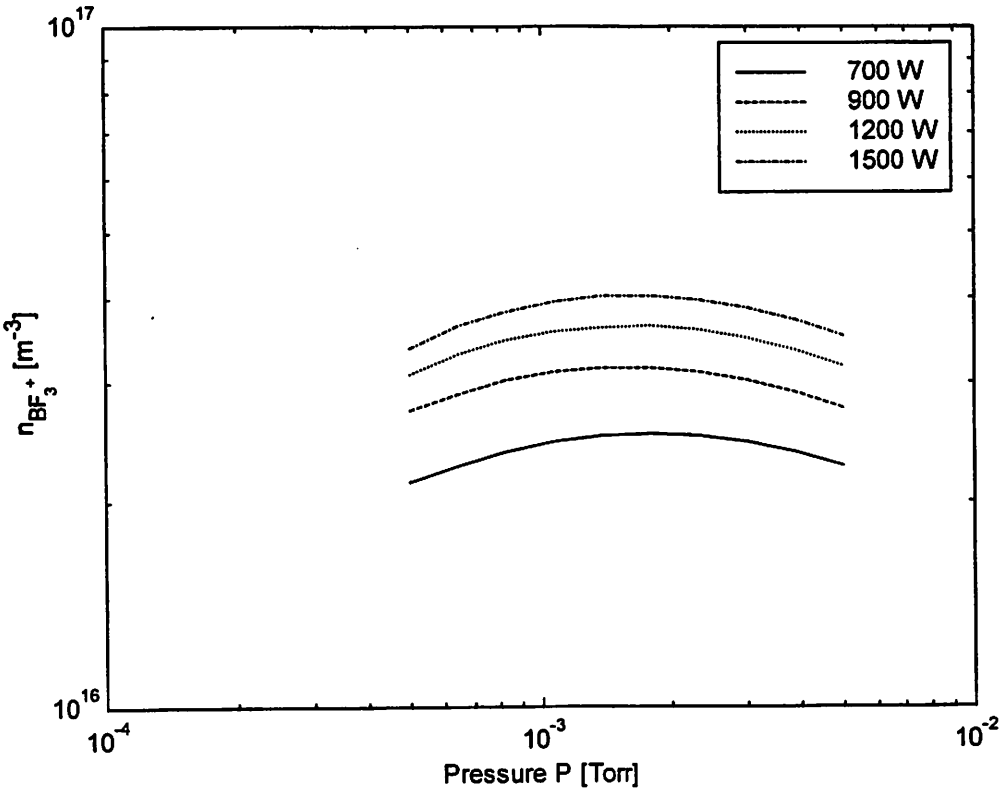

Figure 2.19: BF_3 ion density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

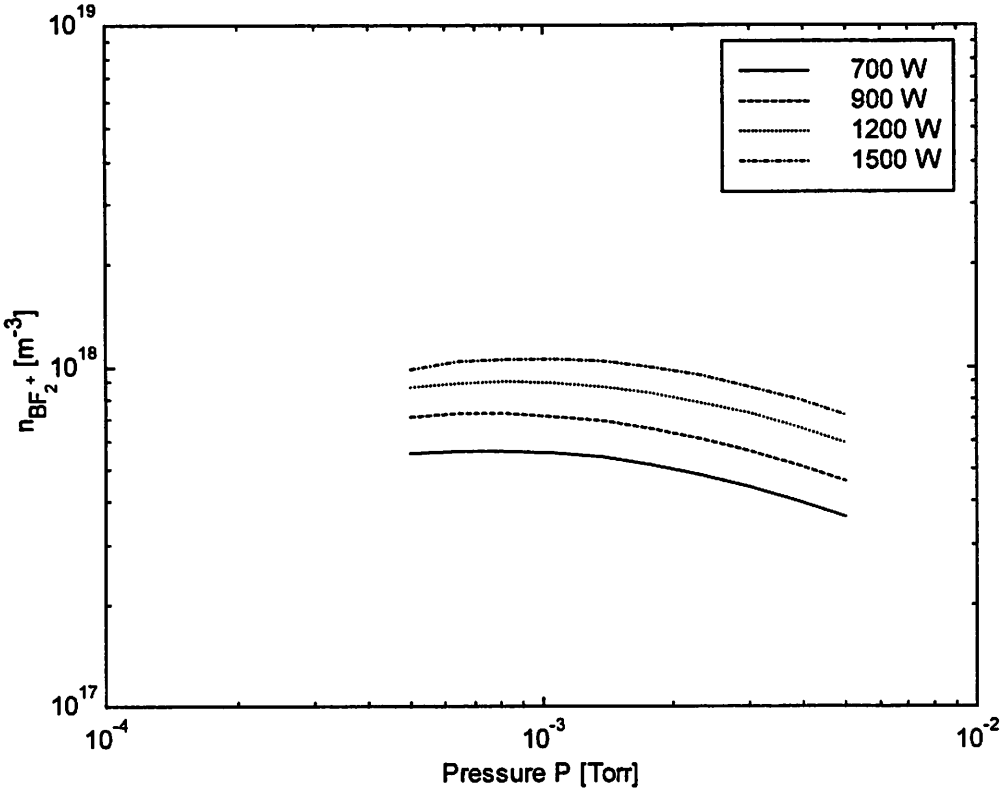

Figure 2.20: BF_2 ion density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

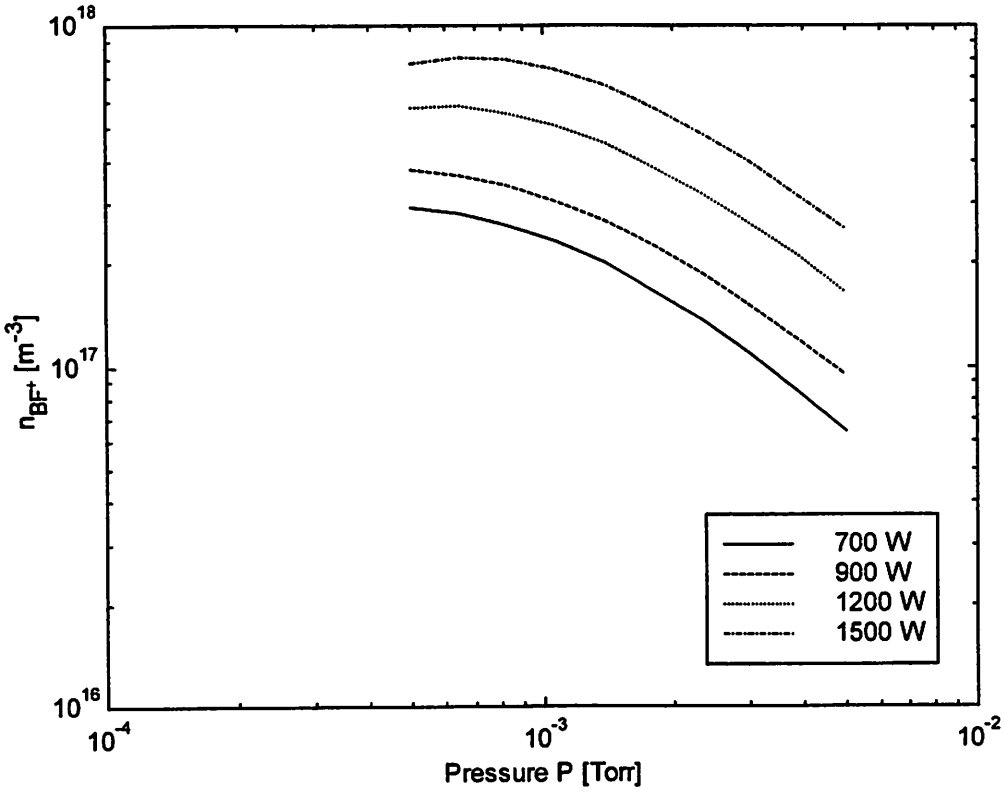

Figure 2.21: BF ion density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

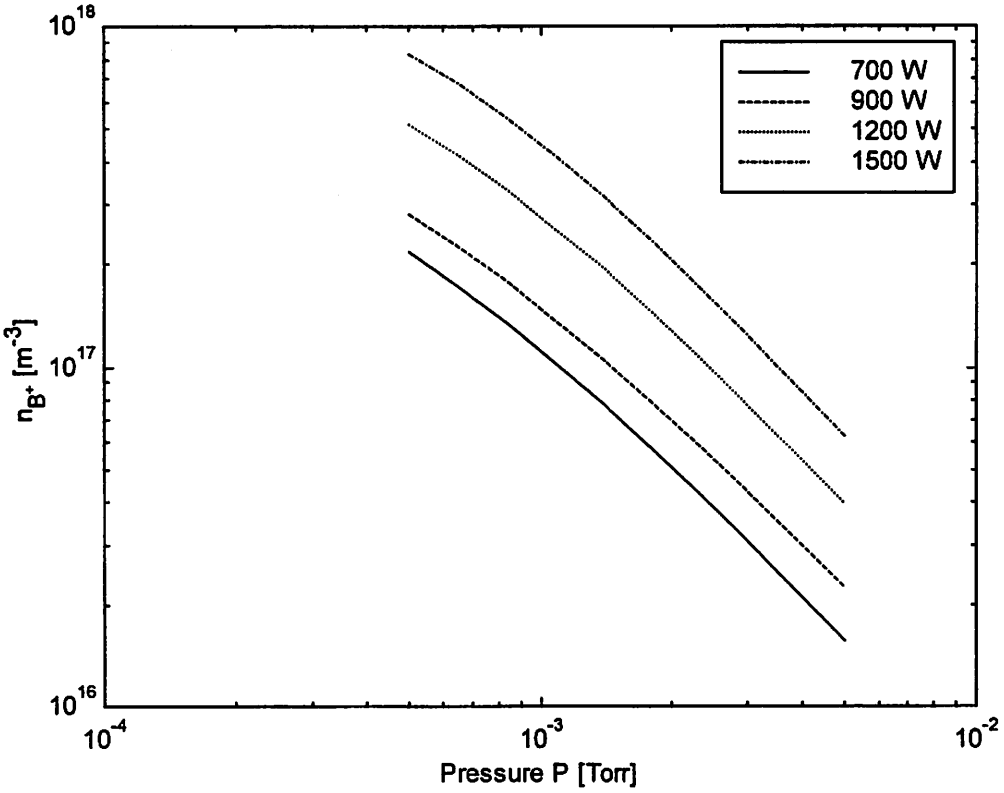

Figure 2.22: Singly ionized boron density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

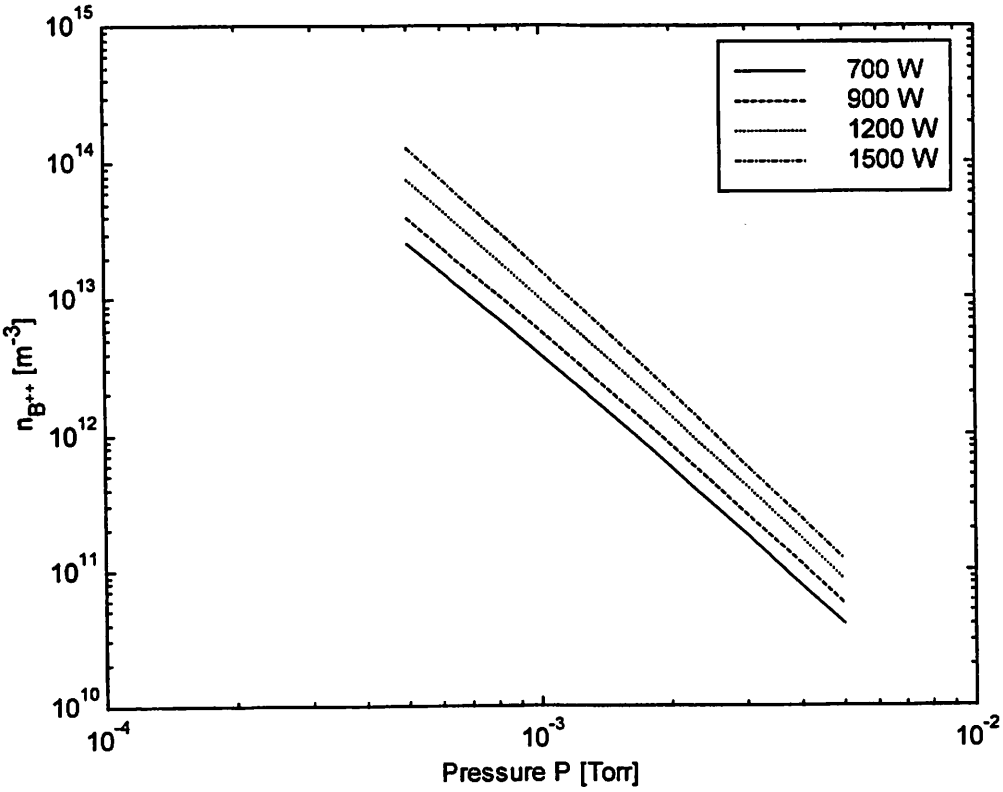

Figure 2.23: Doubly ionized boron density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

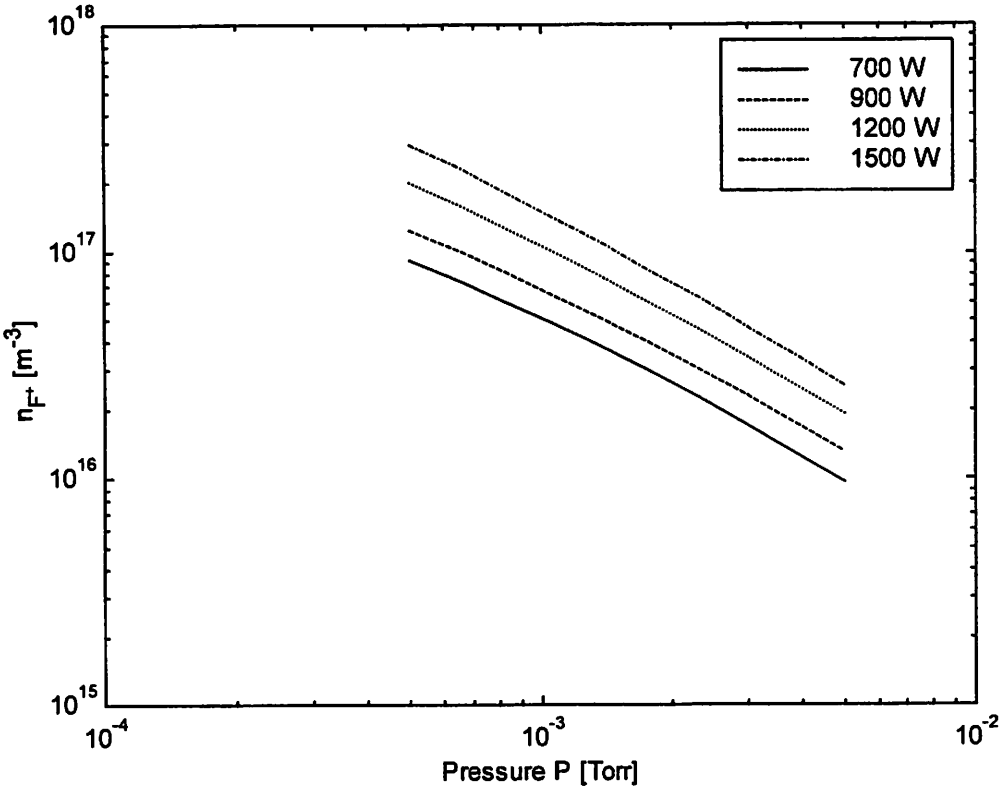

Figure 2.24: Fluorine ion density as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

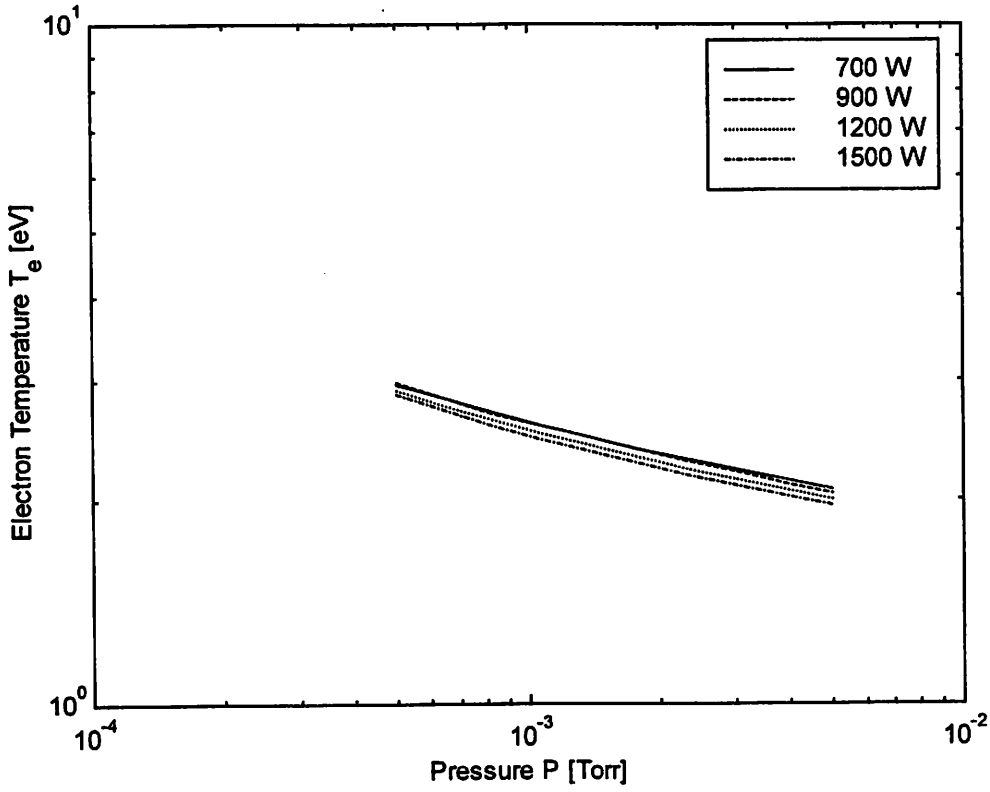

Figure 2.25: Electron temperature as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

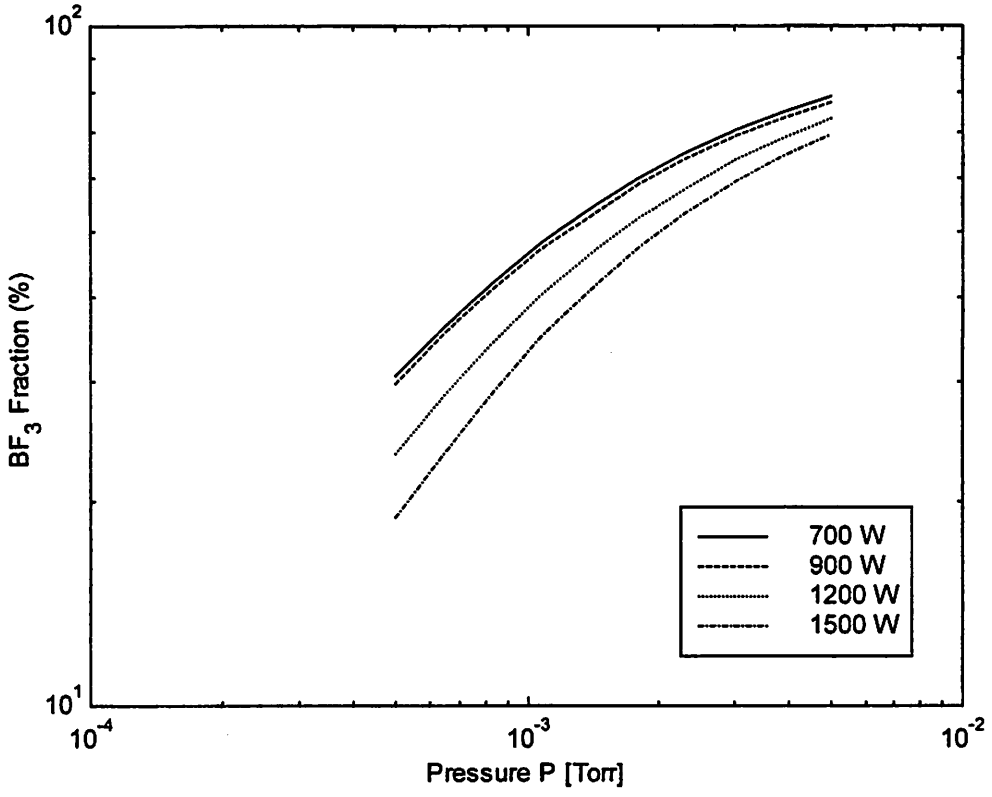

Figure 2.26: BF_3 fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

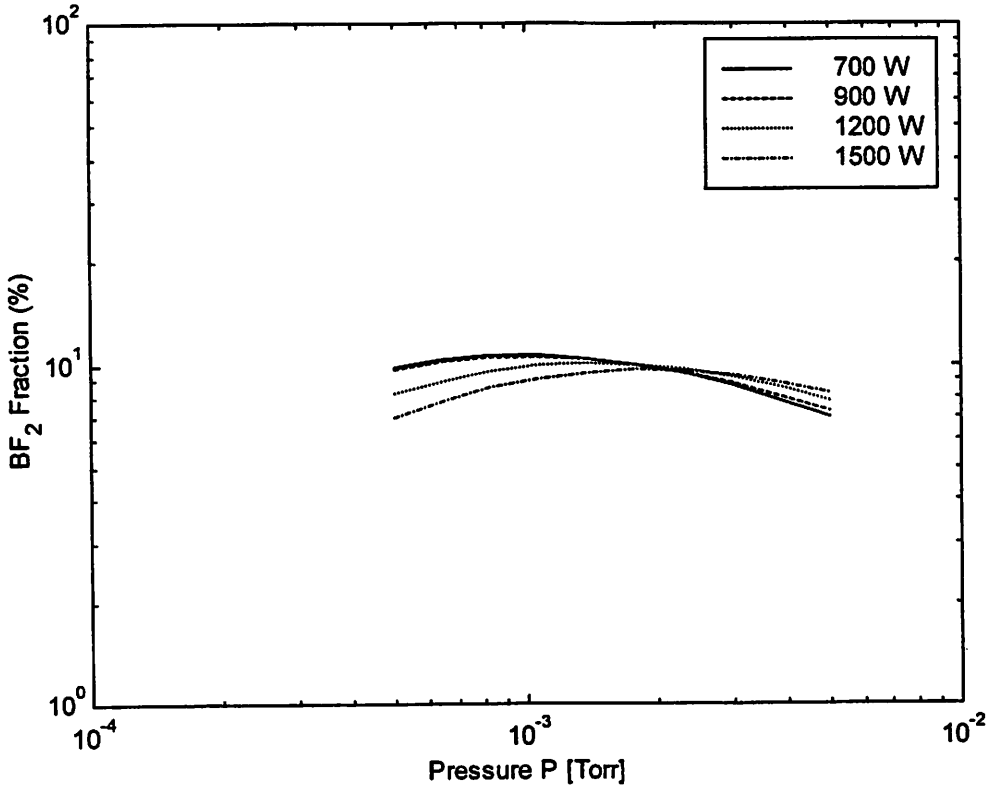

Figure 2.27: BF_2 fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

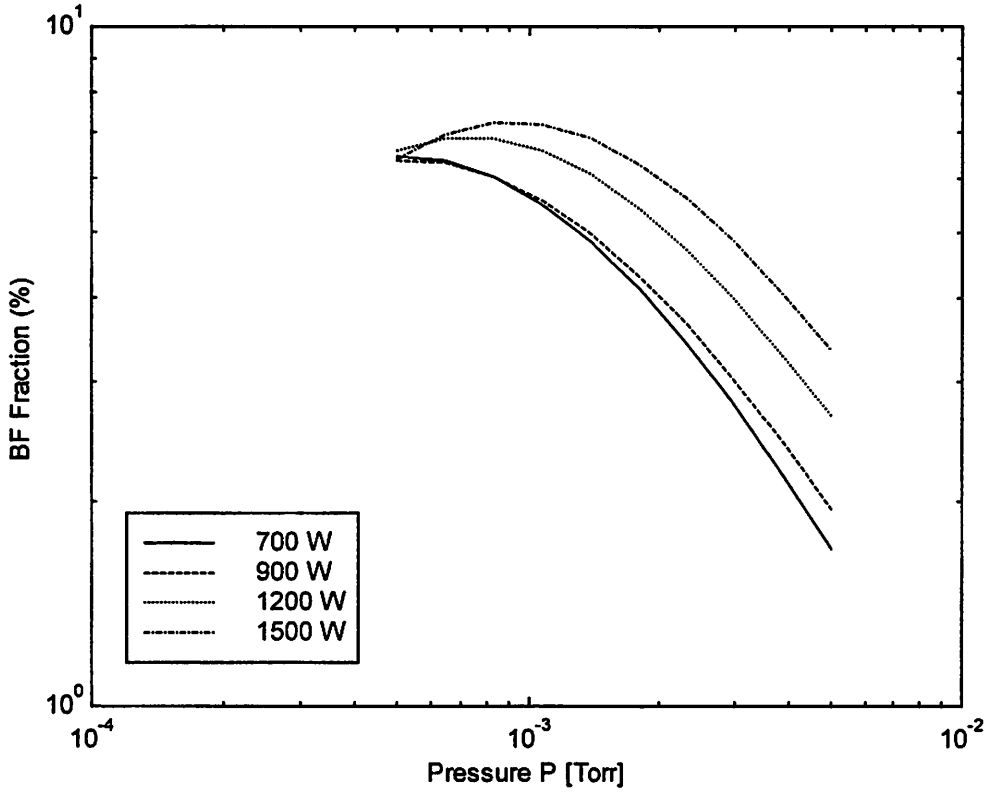

Figure 2.28: BF fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

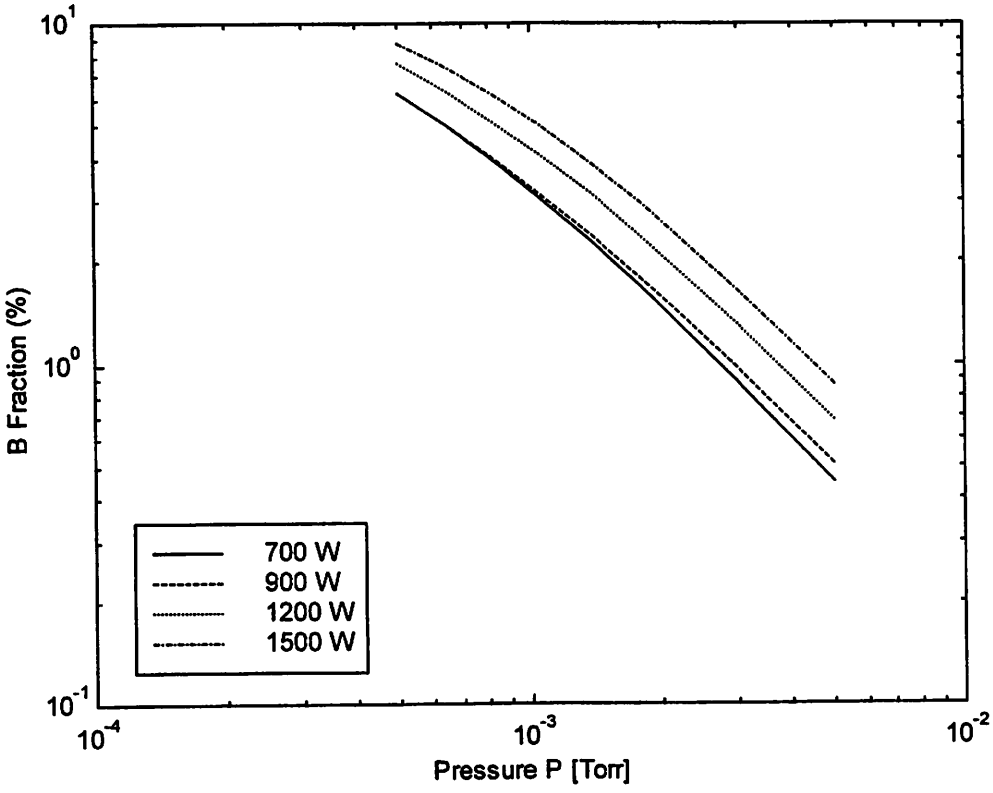

Figure 2.29: Boron fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

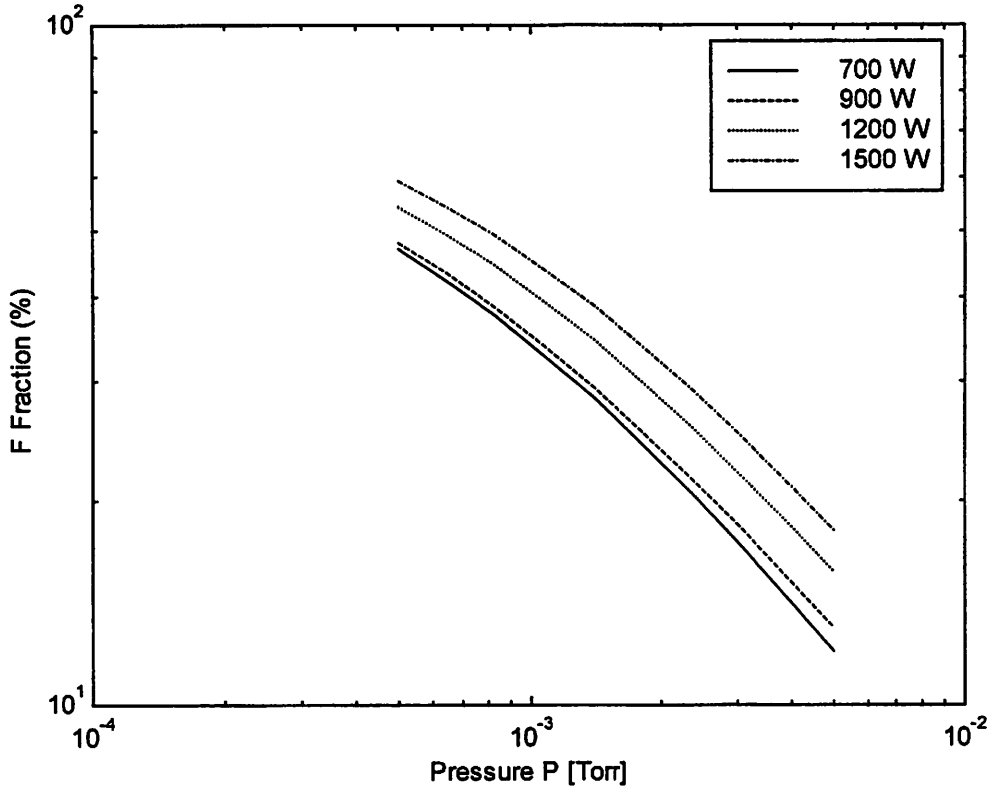

Figure 2.30: Fluorine fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

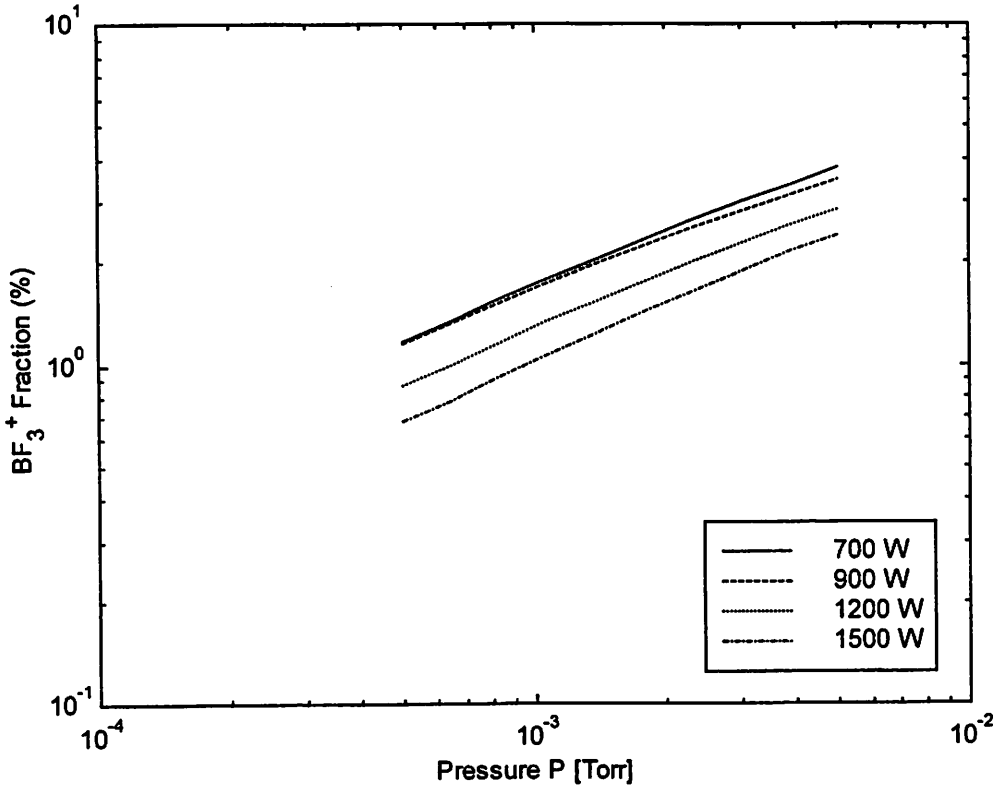

Figure 2.31: BF_3 ion flux fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

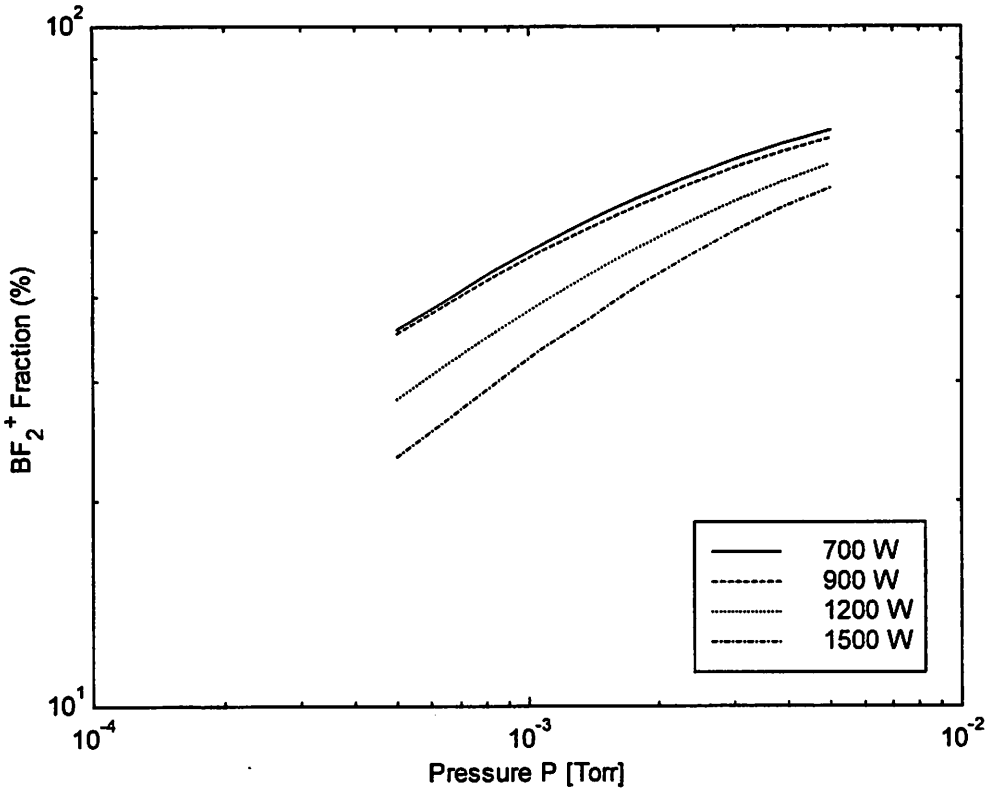

Figure 2.32: BF_2 ion flux fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

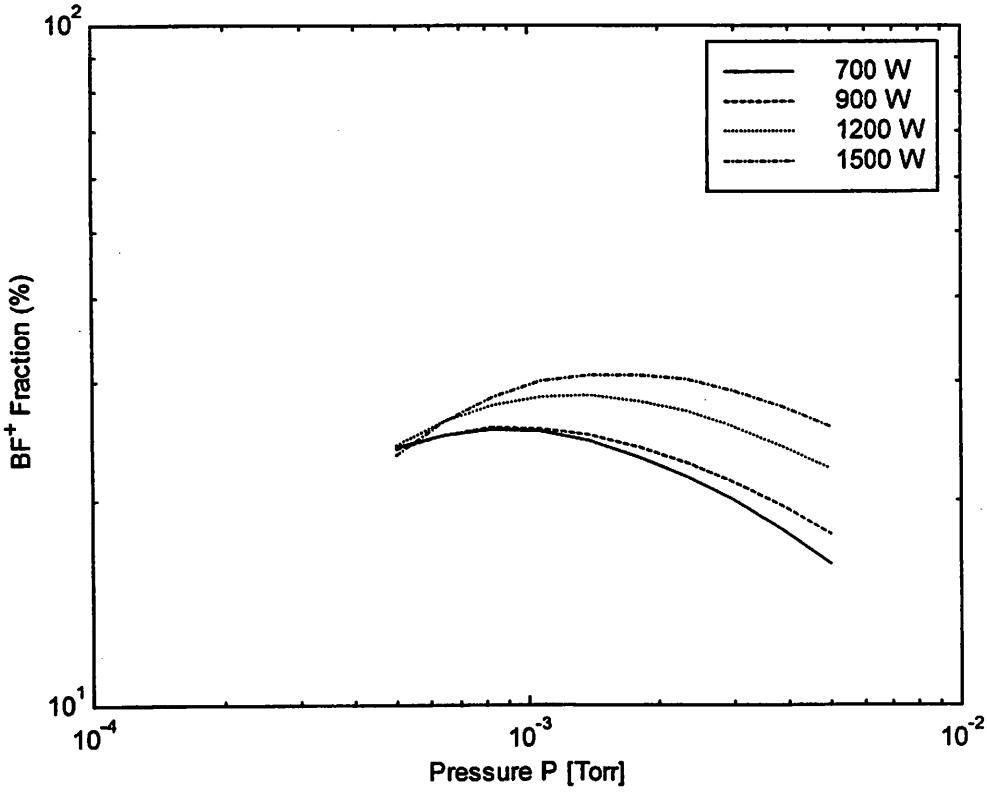

Figure 2.33: BF ion flux fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

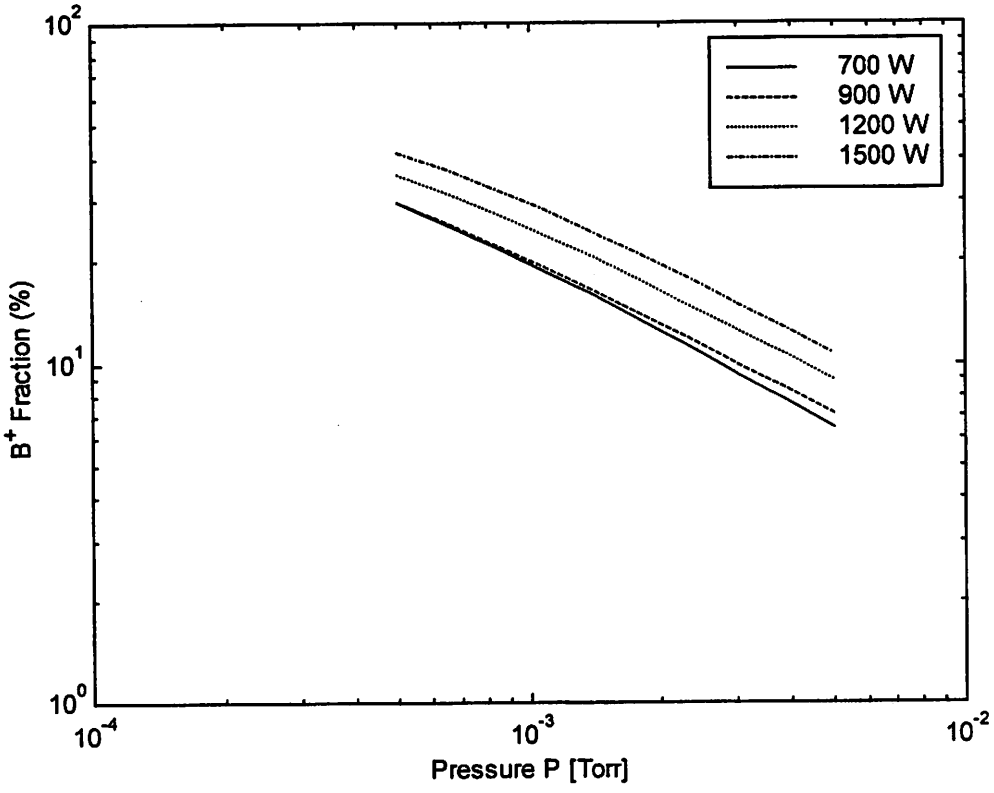

Figure 2.34: Singly ionized boron ion flux fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

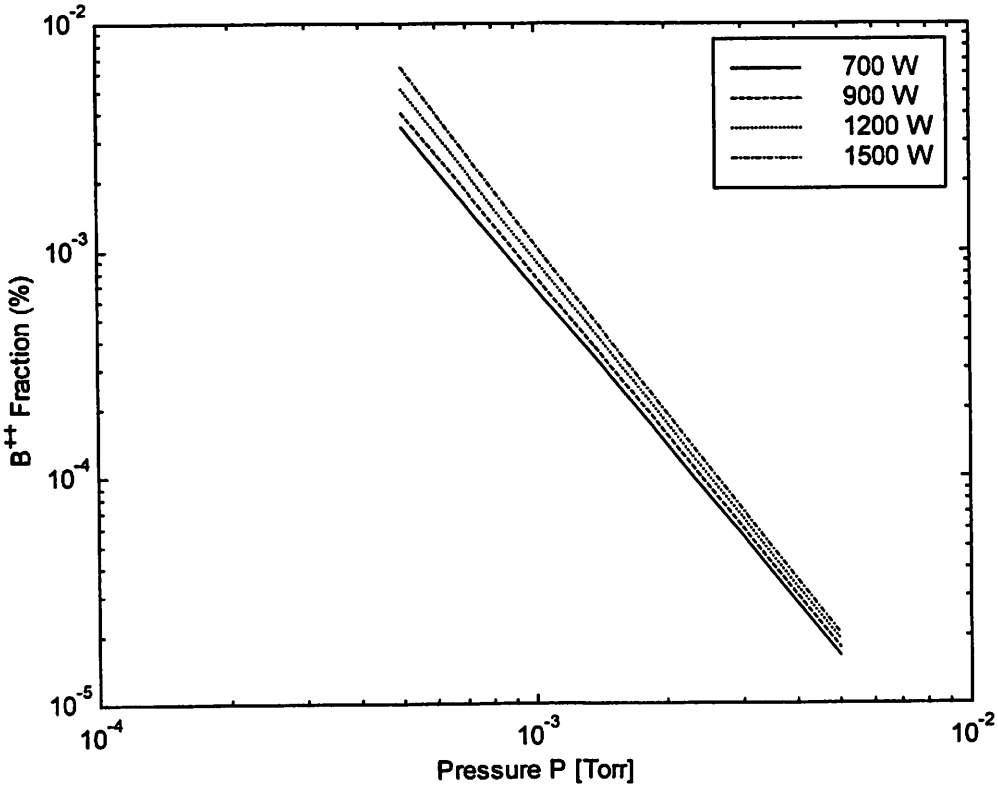

Figure 2.35: Doubly ionized boron ion flux fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

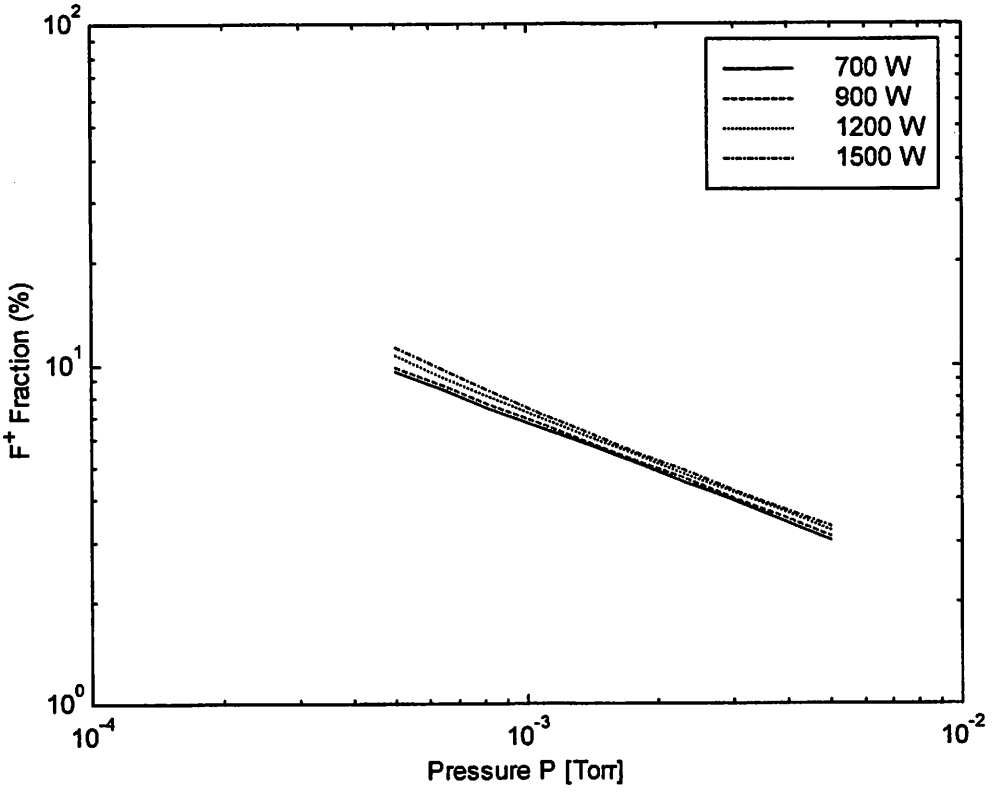

Figure 2.36: Fluorine ion flux fraction as a function of reactor pressure. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

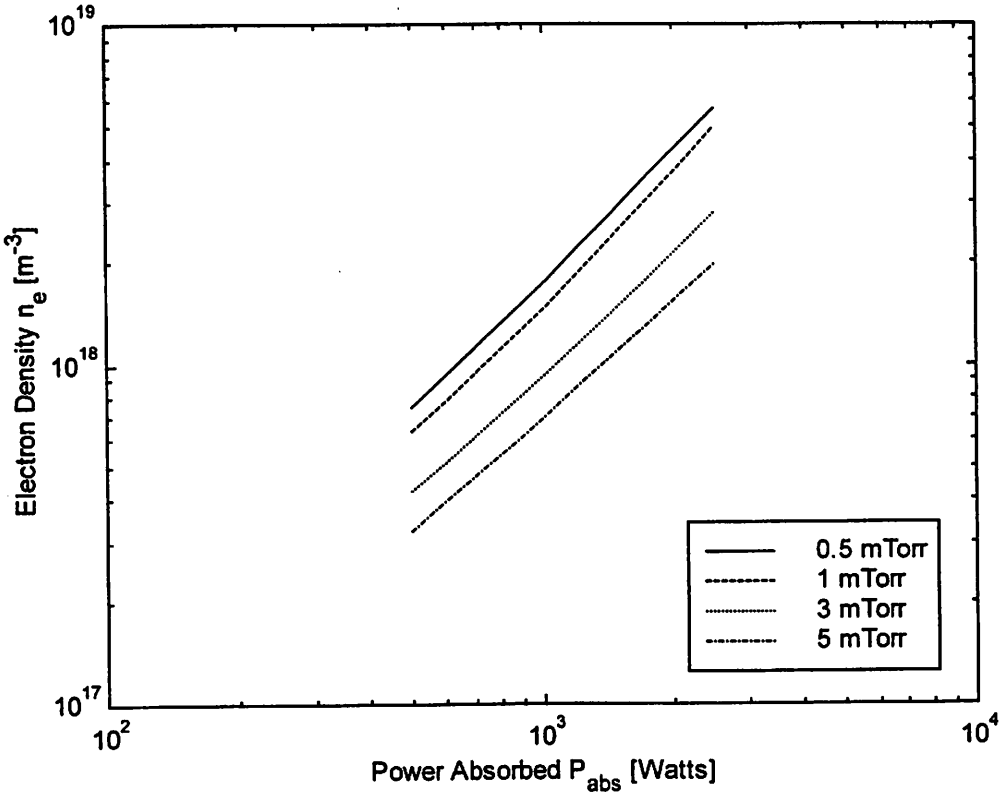

Figure 2.37: Electron density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

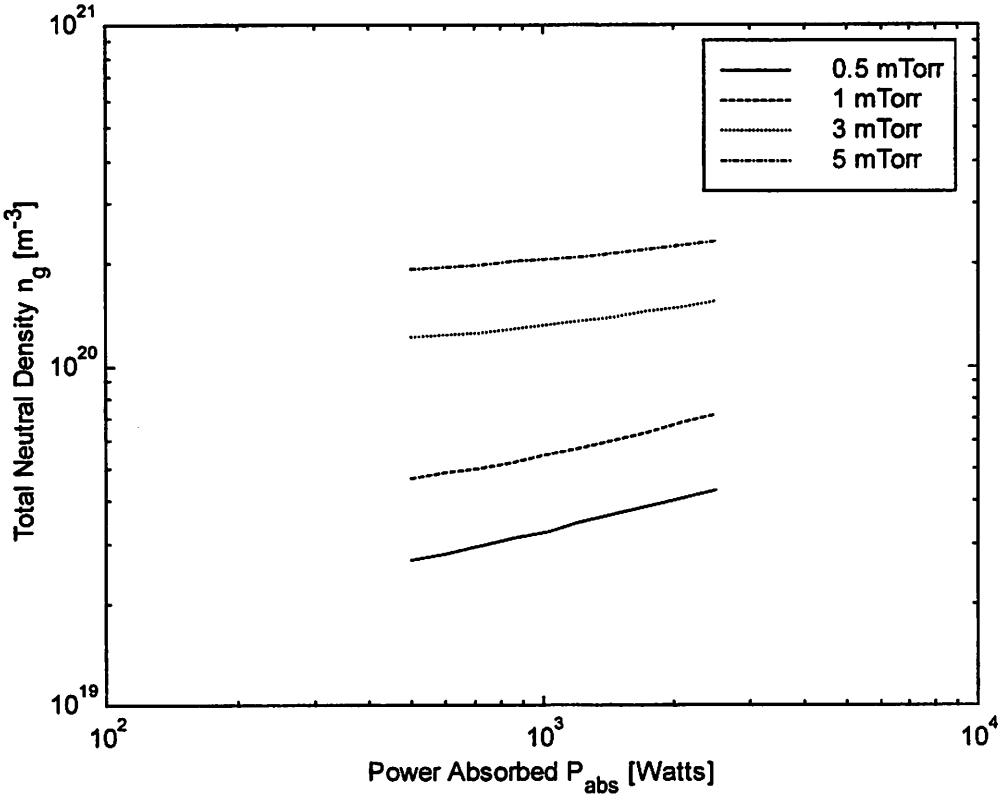

Figure 2.38: Total neutral density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

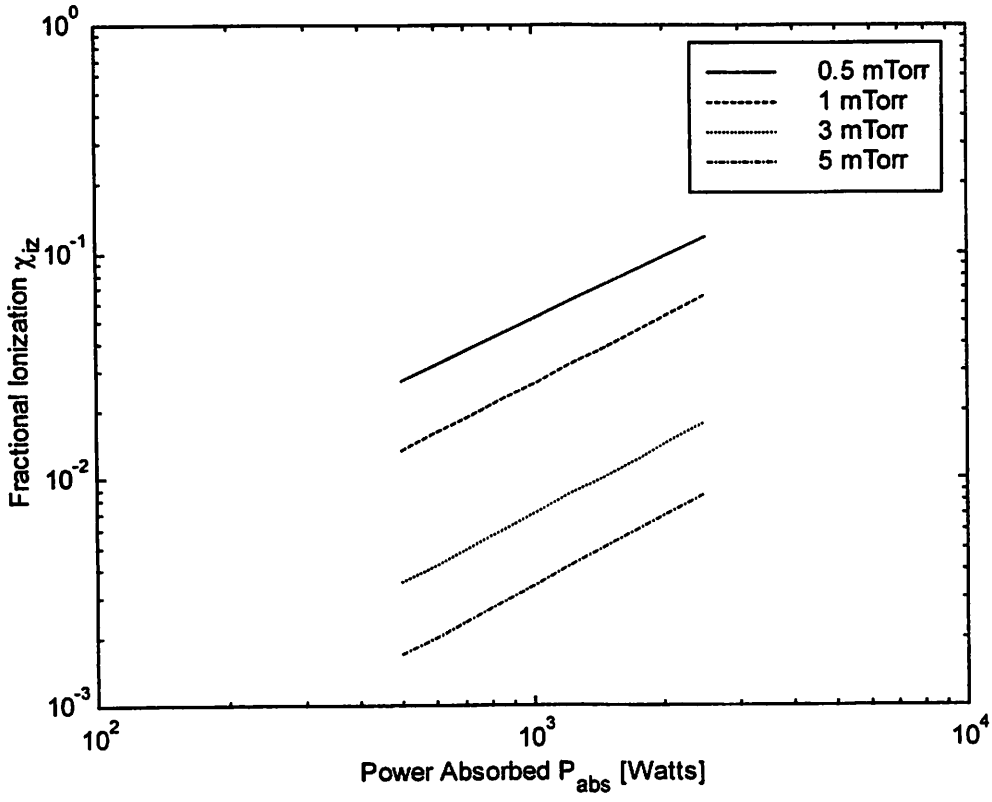

Figure 2.39: Fractional ionization as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

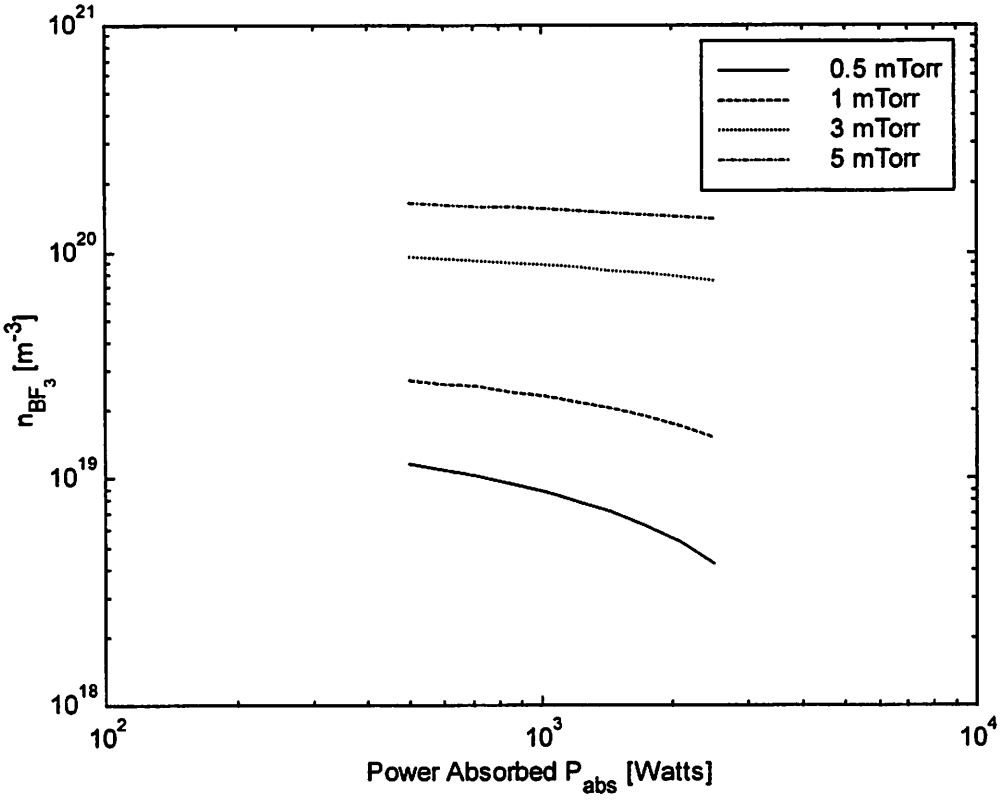

Figure 2.40: BF_3 density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

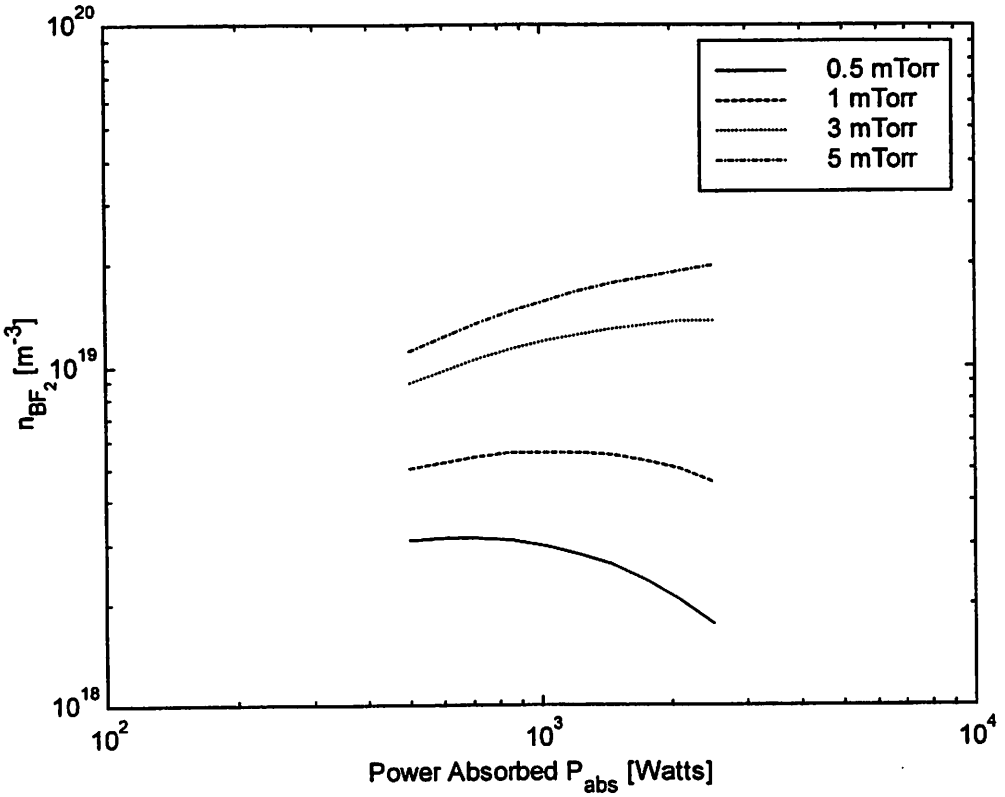

Figure 2.41: BF_2 density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

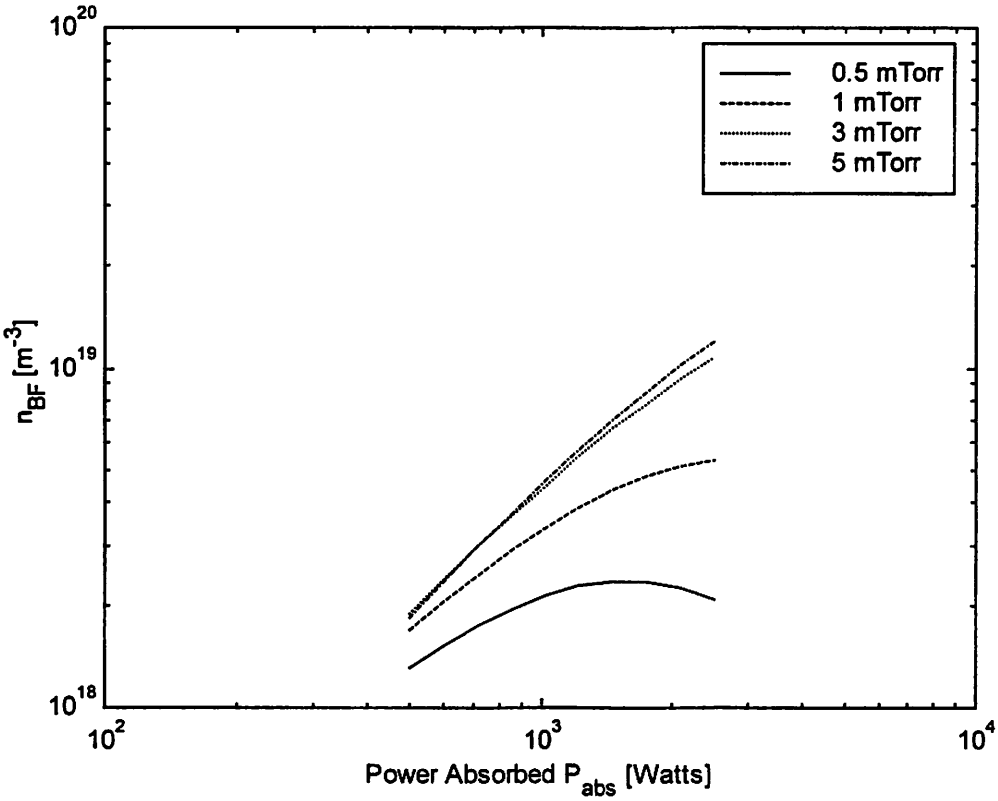

Figure 2.42: BF as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

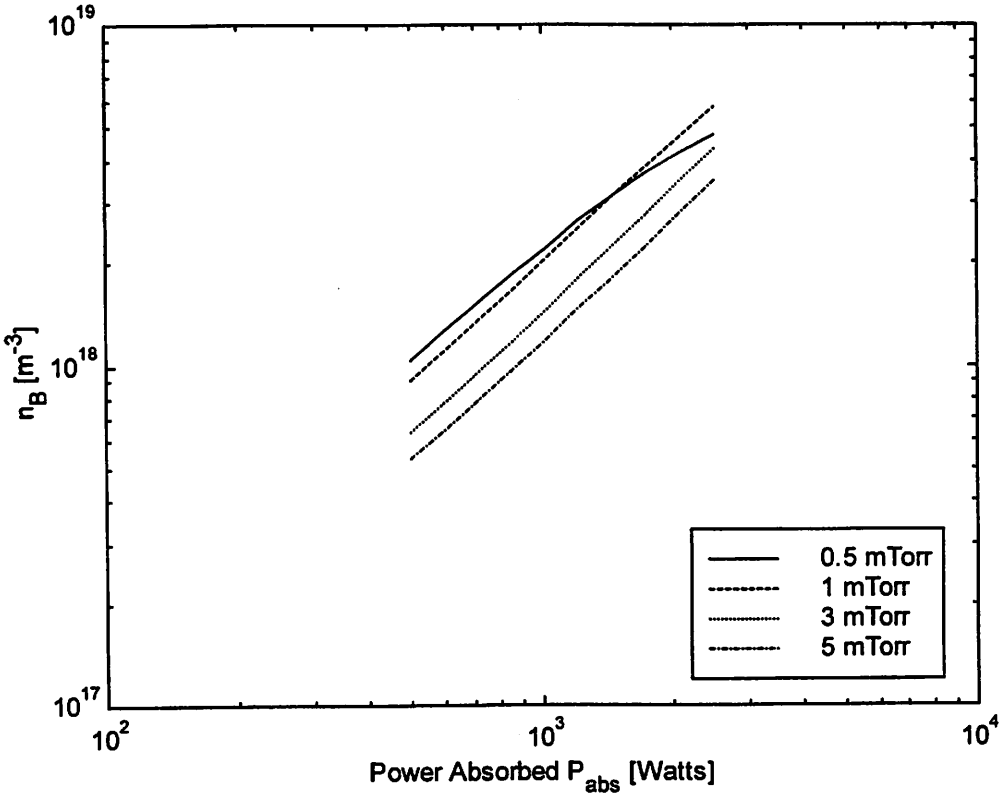

Figure 2.43: Boron density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

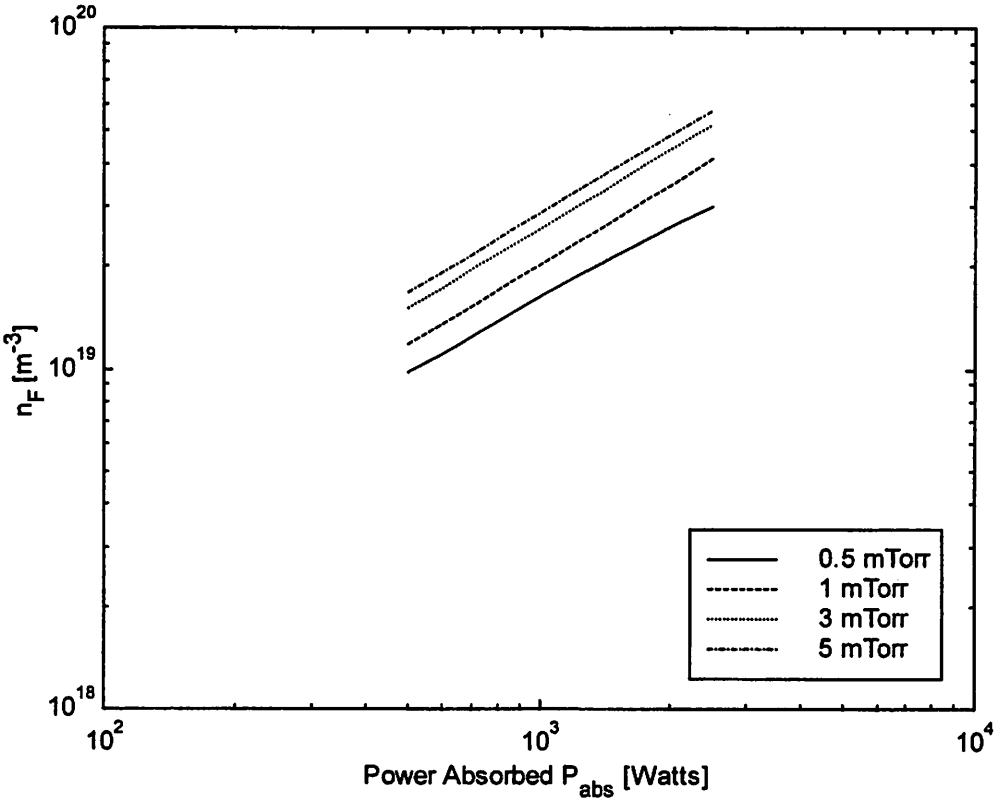

Figure 2.44: Fluorine density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

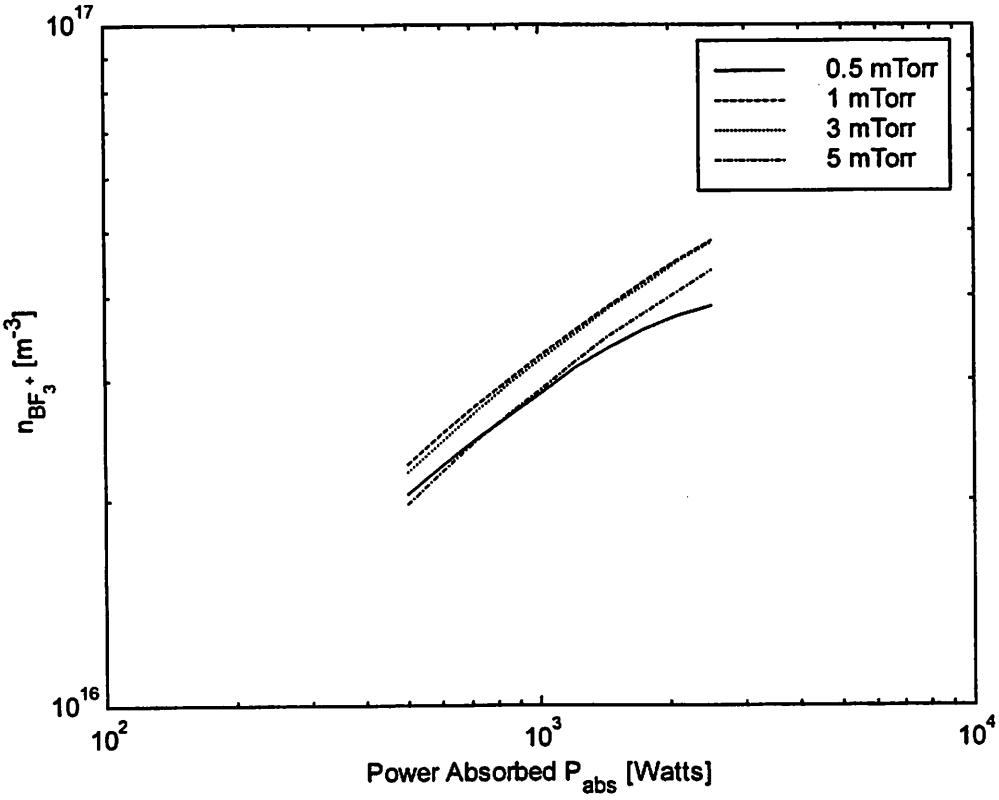

Figure 2.45: BF_3 ion density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

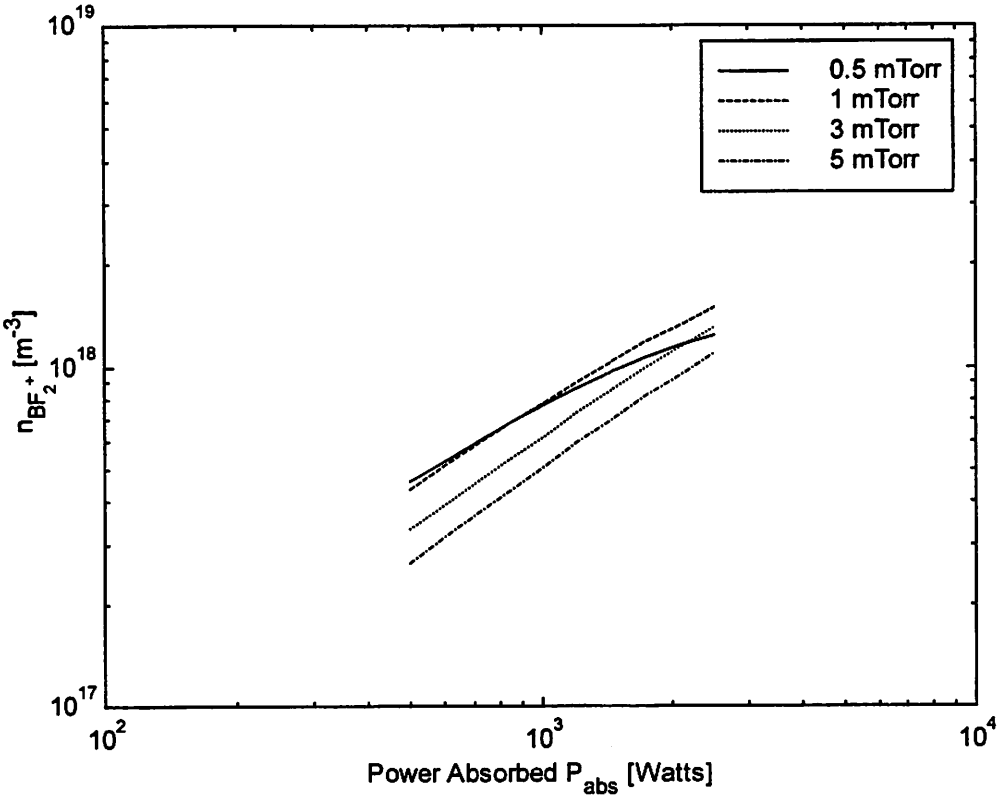

Figure 2.46: BF_2 ion density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

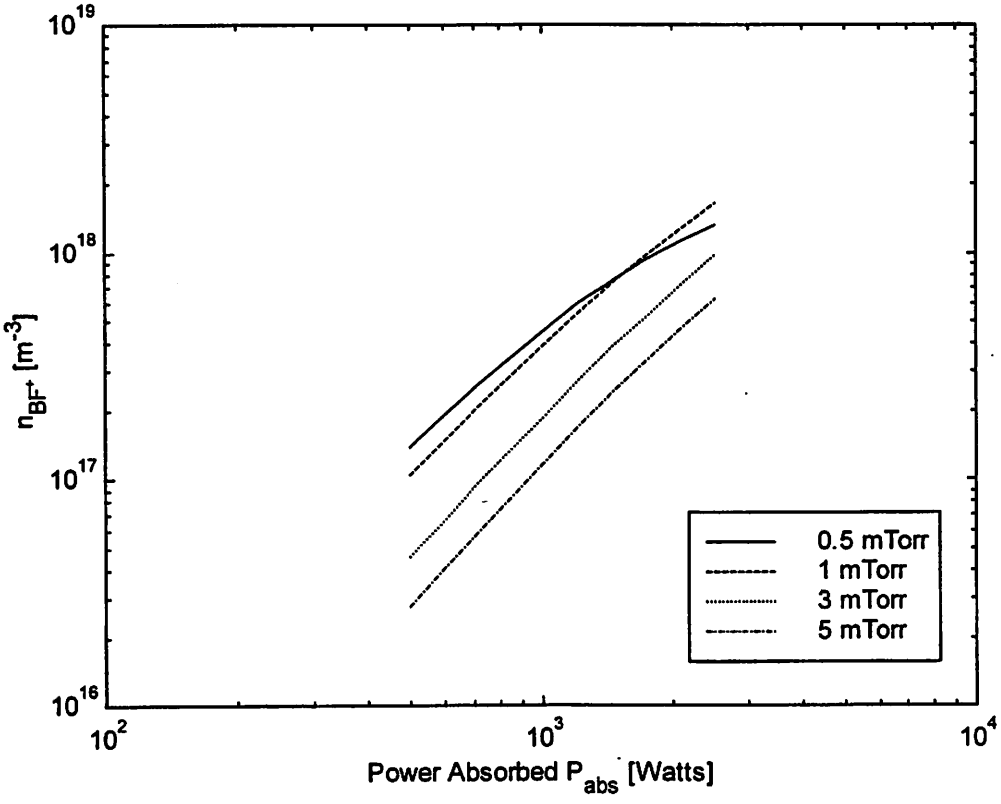

Figure 2.47: BF ion density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

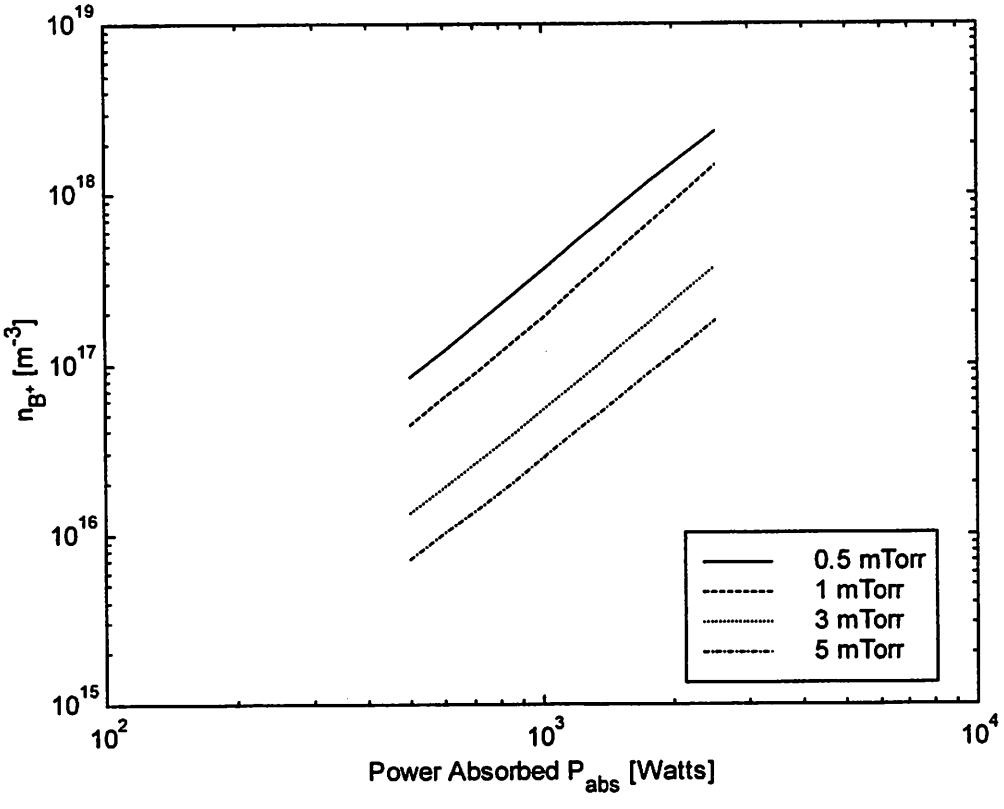

Figure 2.48: Singly ionized boron density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

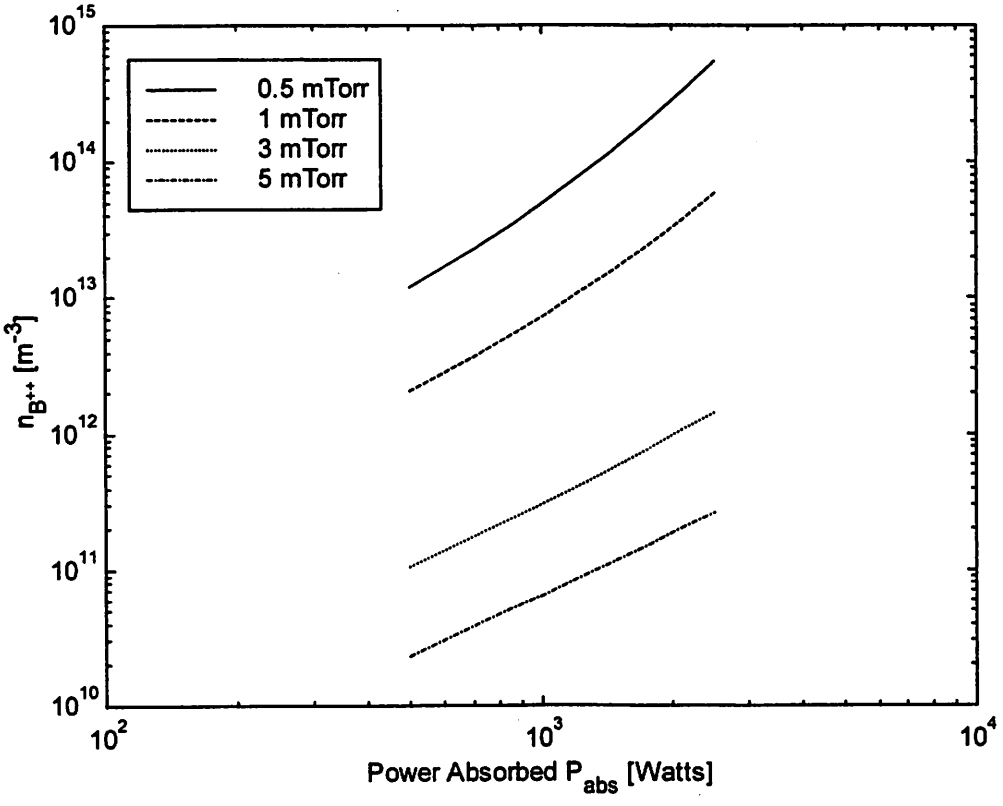

Figure 2.49: Doubly ionized boron density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

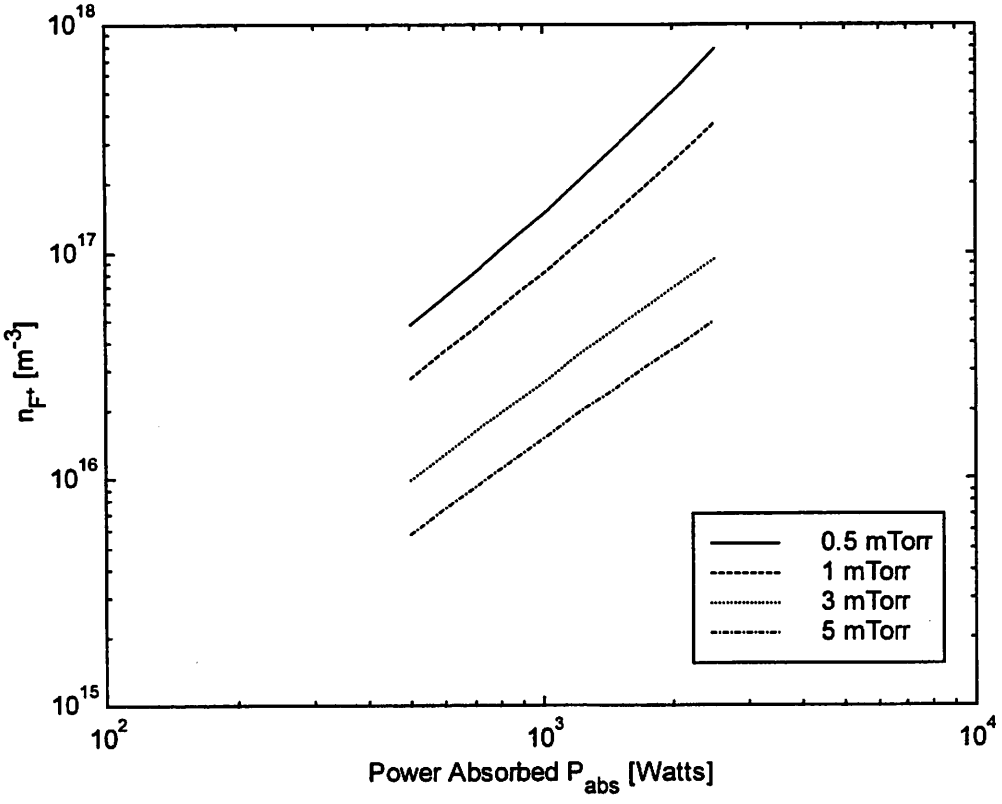

Figure 2.50: Fluorine ion density as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

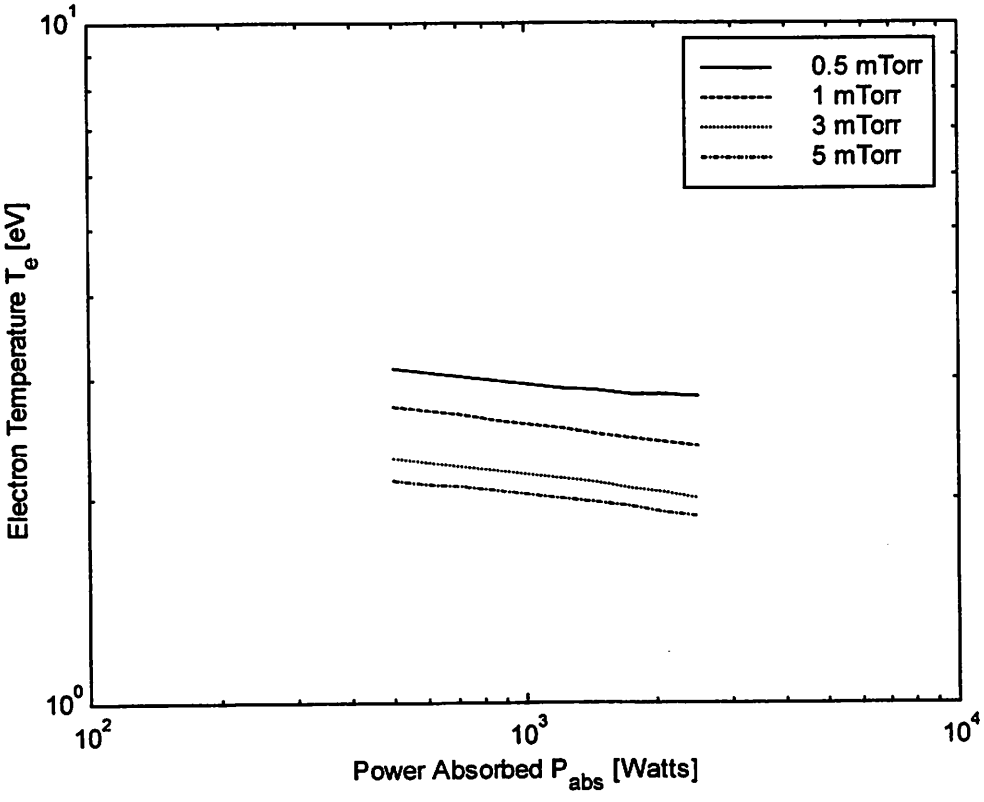

Figure 2.51: Electron temperature as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

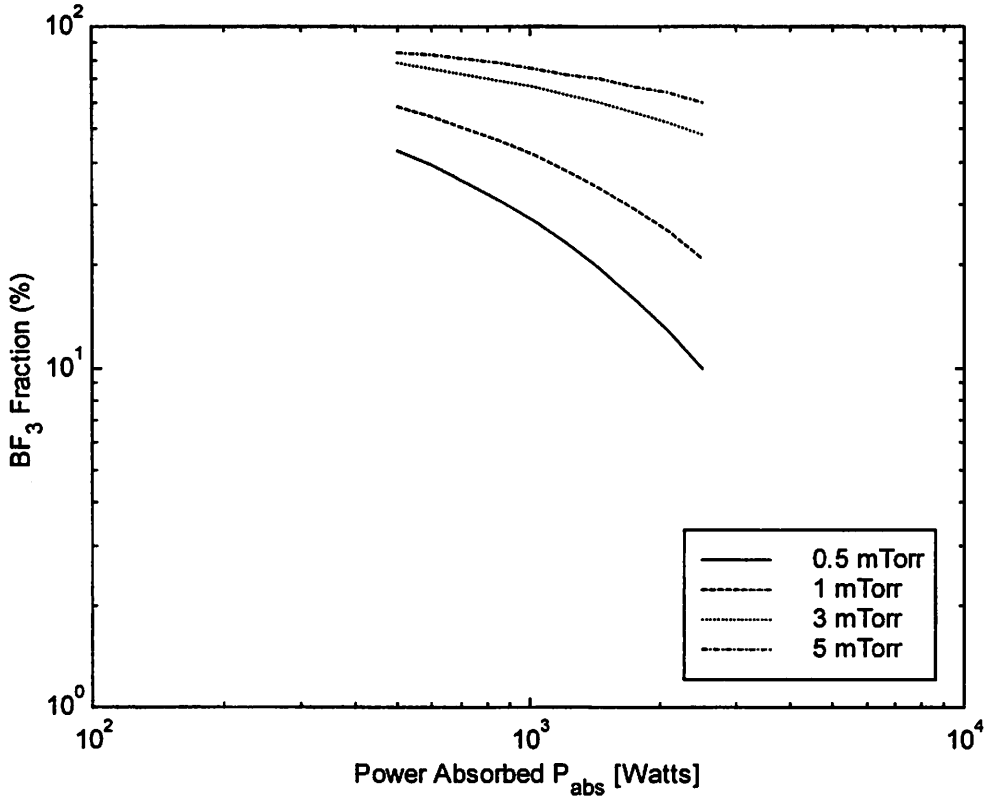

Figure 2.52: BF_3 fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

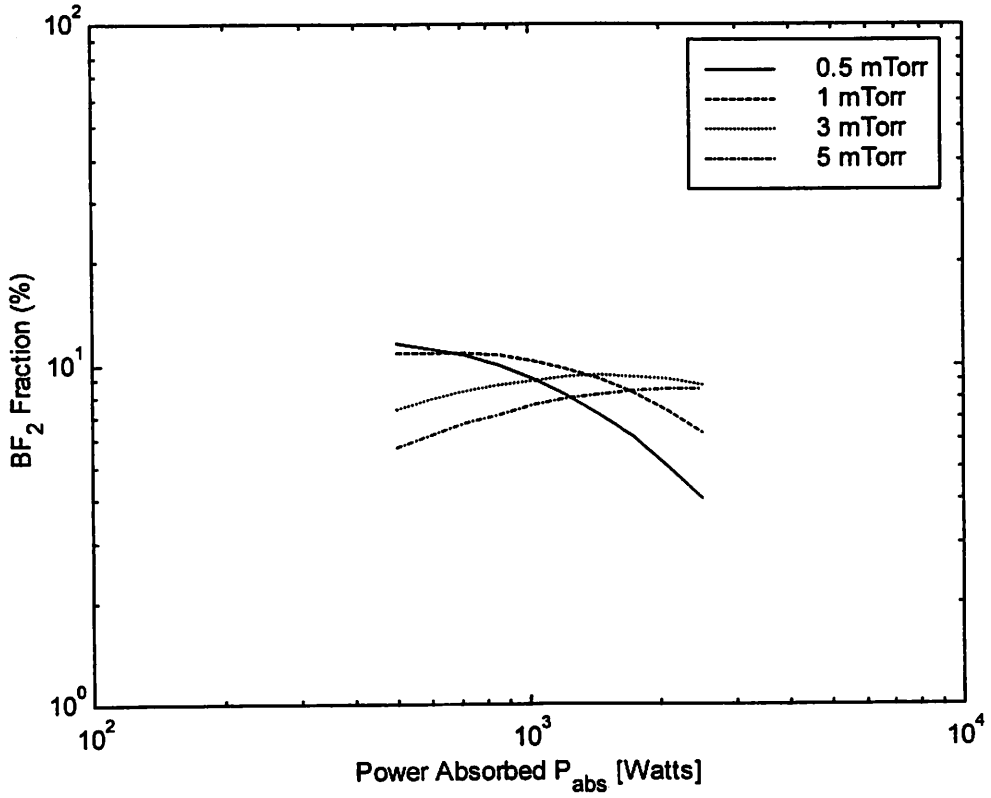

Figure 2.53: BF_2 fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

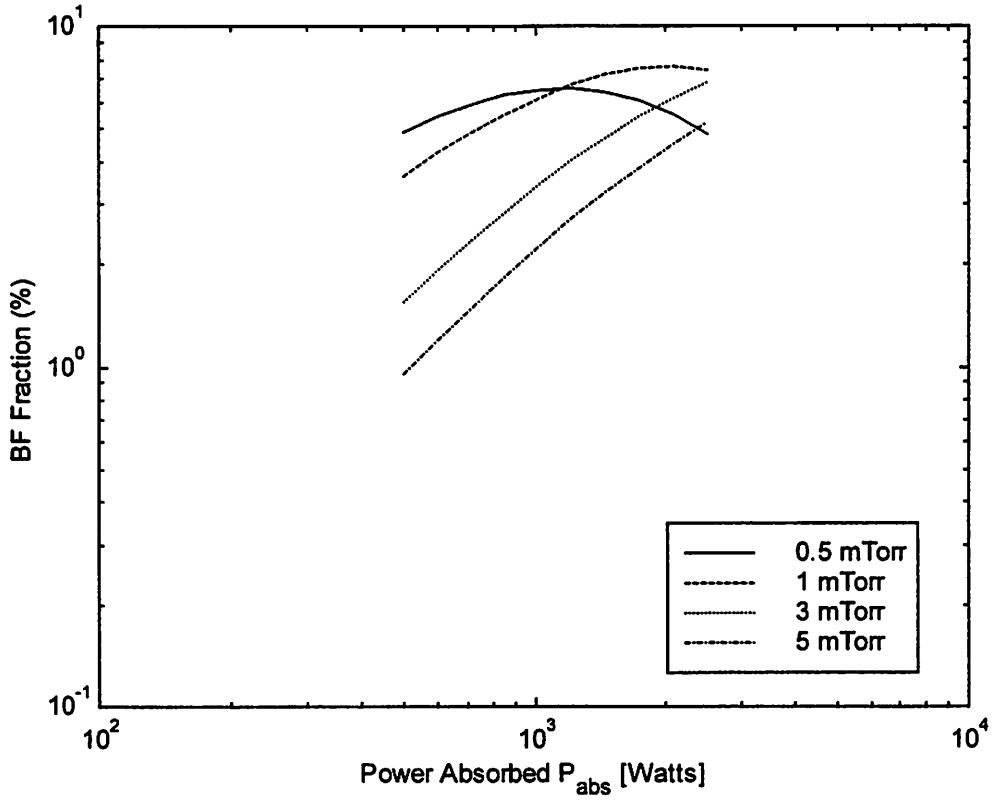

Figure 2.54: BF fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

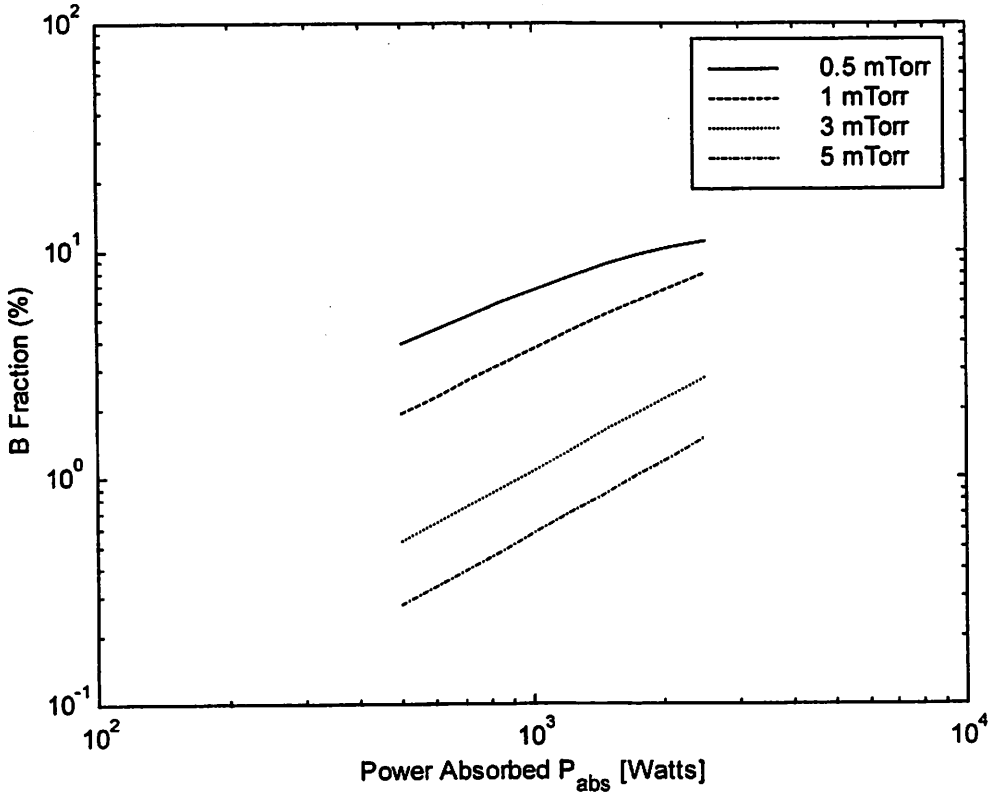

Figure 2.55: Boron fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

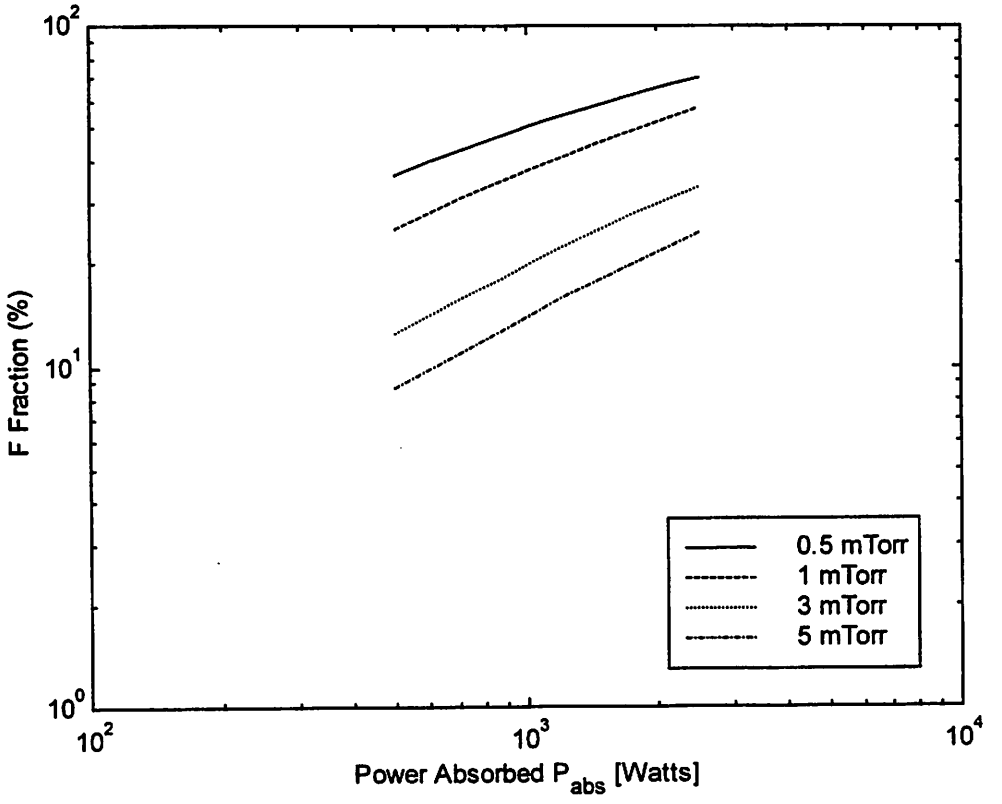

Figure 2.56: Fluorine fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

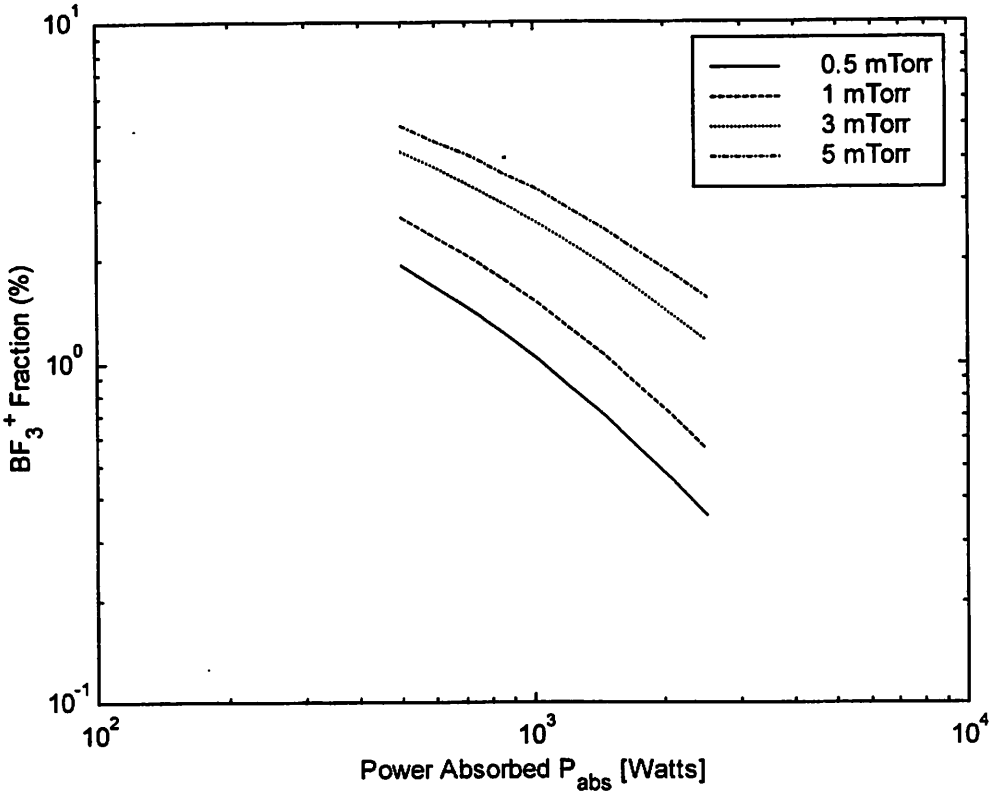

Figure 2.57: BF_3 ion flux fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

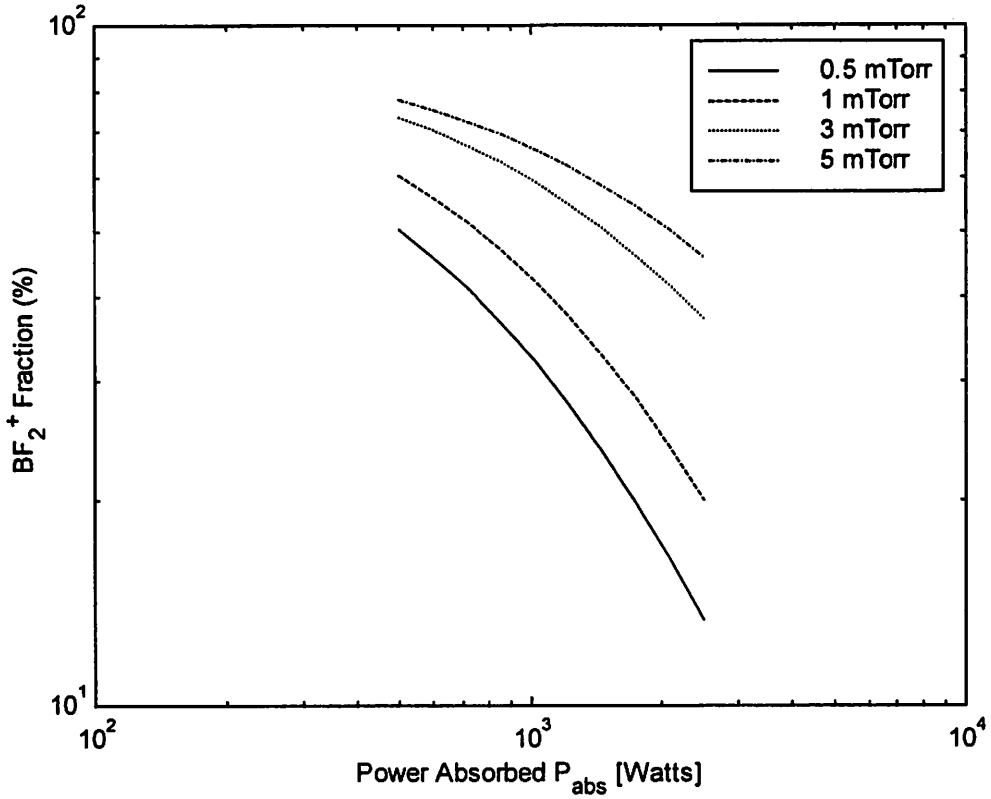

Figure 2.58: BF_2 ion flux fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

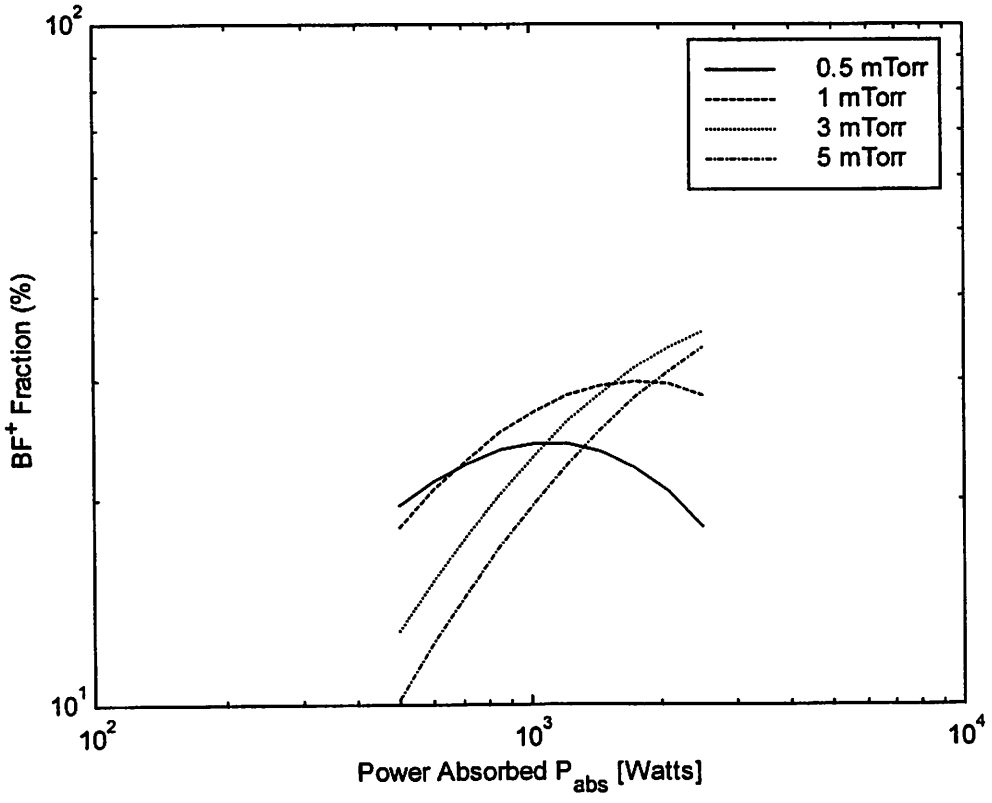

Figure 2.59: BF ion flux fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

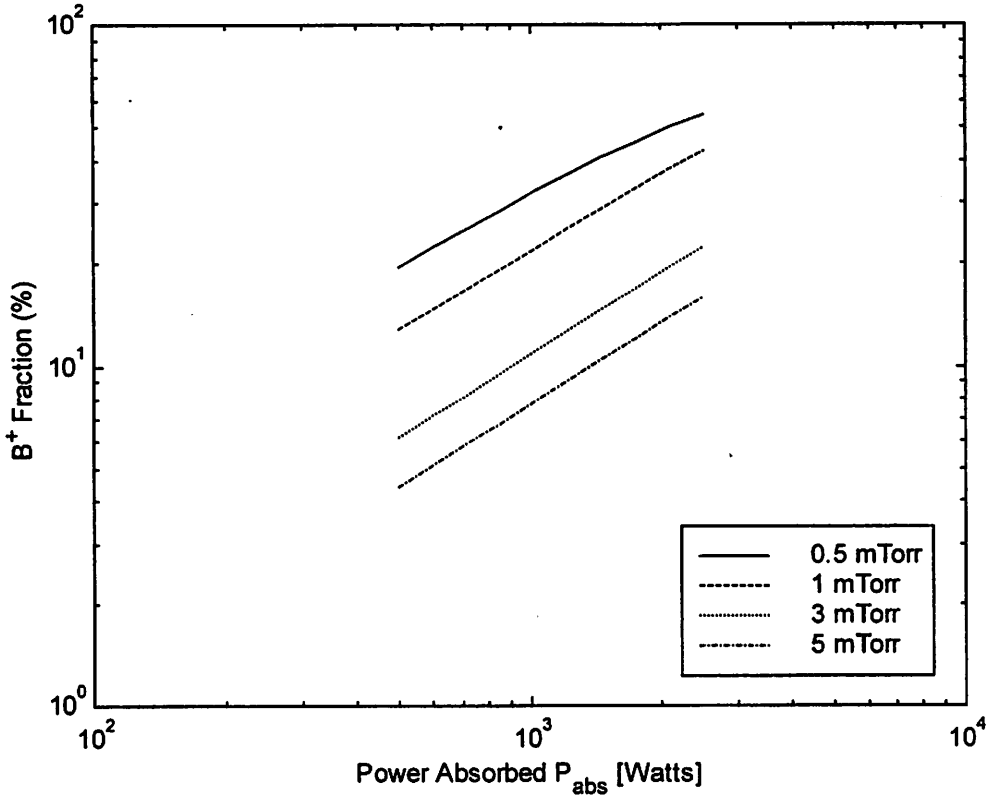

Figure 2.60: Singly ionized boron ion flux fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

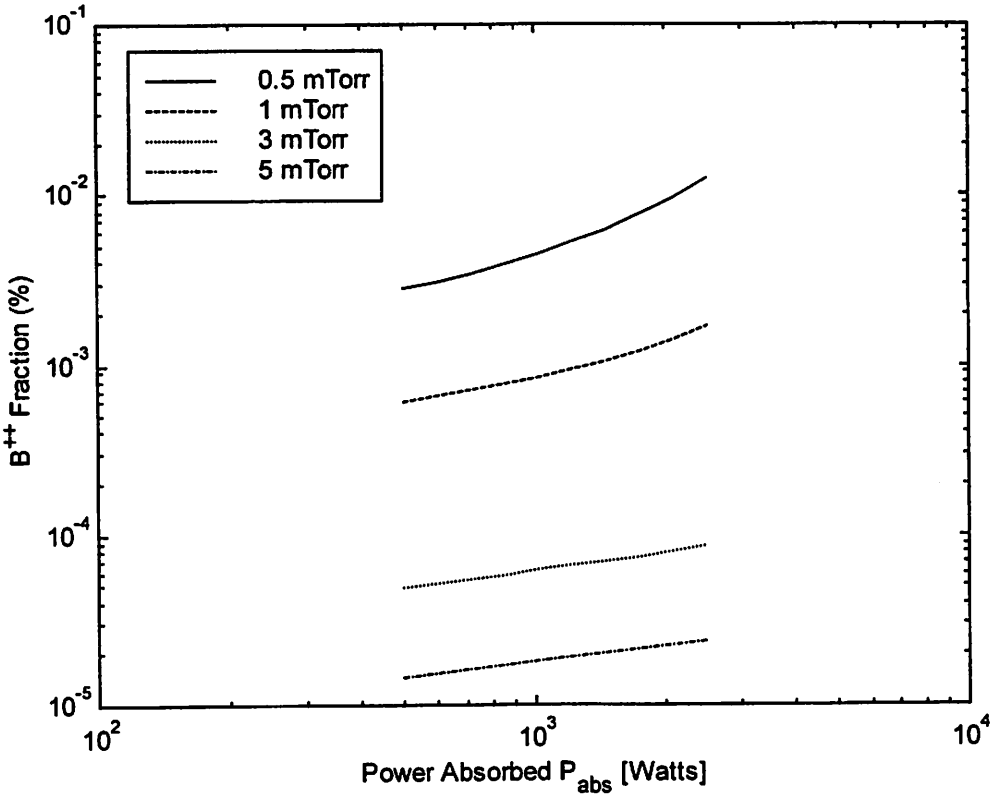

Figure 2.61: Doubly ionized boron ion flux fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

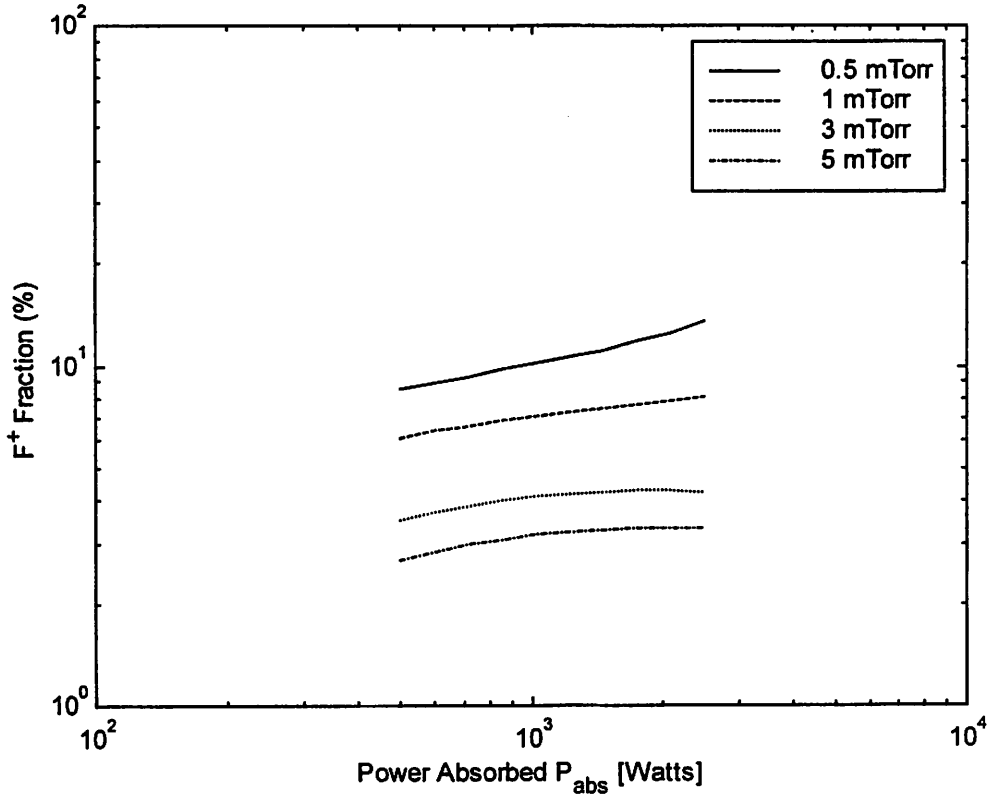

Figure 2.62: Fluorine ion flux fraction as a function of input power. The plasma is magnetically confined and pumping is fixed to give 1mTorr at 1.5 SCCM.

Appendix A

A.1 Input File Structures for Oxygen

Table A.1 Input File Description

File	Description
Discharge.m	This file contains one column for each species, excluding electrons. The rows are mass (in Kg), temperature of the species (in Kelvin), wall recombination coefficient ($0 \le \gamma_{rec} \le 1$), and charge of each species (neutral=0, positive=+1, negative=-1, etc.)
CrossSections.m	This file contains number of columns and rows corresponding to number of species, i.e, number of rows and columns equal number of species (excluding electrons). The units of cross-sections are m^2.
EnergyLossSpecies.m	This file contains information regarding which species are involved in the excitation energy loss reactions with electrons. The columns correspond to each species, excluding the electrons.
EnergyLossConstants.m	This file contains information necessary to assemble the excitation energy rate constants in the form: $k = AT_e^B \exp\left(-\frac{C}{T_e}\right)$. The first three columns are A, B, and C respectively. The fourth column is the threshold energy (in eV) and the fifth column is the power of T_e in the threshold energy, i.e for momentum transfer.
ReactStoic.m	This file contains the stoichiometric coefficients for reactants of all the reactions. The columns correspond to all species, including electrons. The number of rows equal the total number of reactions considered. A reactant species corresponding to no product signifies a pumping reaction for the species.
ProductStoic.m	This file contains the stoichiometric coefficients for products of all the reactions. The columns correspond to all species, including electrons. The number of rows equal the total number of reactions considered. A product species corresponding to no reactant signifies a source for the species.

Table A.1 Input File Description

File	Description
RateStoic.m	This file contains information necessary to assemble the rate constants in the form: $k = AT_e^B \exp\left(-\frac{C}{T_e}\right) T_g^D$. The columns are A, B, C, and D respectively. The fourth column (D) has a non-zero quantity only if the reaction involves only heavy particles (no electrons).

A.1.1 Sample Input Files for Oxygen

Table A.2 Discharge.m

5.34E-26	2.67E-26	2.67E-26	5.34E-26	5.34E-26	2.67E-26
600	600	600	600	600	600
0	1	1	0	0	0
0	0	0	1	1	-1

Table A.3 CrossSections.m

5.00E-19	5.00E-19	5.00E-19	5.00E-19	5.00E-19	0
5.00E-19	5.00E-19	5.00E-19	5.00E-19	5.00E-19	0
5.00E-19	5.00E-19	5.00E-19	5.00E-19	5.00E-19	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Table A.4 EnergyLossSpecies.m

1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0

Table A.5 EnergyLossConstants.m

1.64E-13	0	4.749	5.11E-05	1
9.00E-16	2	12.600	1.21E+01	0
1.37E-15	0	2.144	9.77E-01	0
3.24E-16	0	2.218	1.63E+00	0
2.80E-15	0	3.720	1.90E-01	0
1.28E-15	0	3.670	3.80E-01	0
7.81E-16	0	3.833	5.70E-01	0
4.80E-16	0	4.330	7.50E-01	0
1.07E-15	0	3.428	4.50E+00	0
3.73E-15	0	4.895	6.00E+00	0
3.91E-14	0	8.287	8.40E+00	0
3.92E-16	0	11.480	9.97E+00	0
4.47E-15	0	2.286	1.96E+00	0
4.54E-15	0	4.490	4.18E+00	0
4.54E-15	0	17.340	1.57E+01	0
9.67E-16	0	9.970	9.14E+00	0
9.67E-16	0	9.750	9.51E+00	0
4.31E-14	0	18.600	1.20E+01	0
9.00E-15	0.7	13.600	1.36E+01	0
1.64E-13	0	4.749	1.02E-04	1

Table A.6 ReactStoic.m

1	1	0	0	0	0	0
1	0	0	0	1	0	0
1	1	0	0	0	0	0
1	0	1	0	0	0	0
0	0	0	0	1	0	1
0	0	0	0	0	1	1
1	0	0	0	0	0	1
1	1	0	0	0	0	0
0	0	1	0	0	0	1
1	1	0	0	0	0	0
1	1	0	0	0	0	0
0	1	0	0	0	1	0
1	1	0	0	0	0	0
1	0	1	0	0	0	0
0	1	0	1	0	0	0
0	0	1	1	0	0	0
1	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0

Table A.7 ProdStoic.m

2	0	0	0	1	0	0
0	0	2	0	0	0	0
0	0	1	0	0	0	1
2	0	0	0	0	1	0
0	1	1	0	0	0	0
0	0	2	0	0	0	0
2	0	1	0	0	0	0
1	0	2	0	0	0	0
1	1	0	0	0	0	0
1	0	0	0	0	1	1
2	0	1	0	0	1	0
0	0	1	0	1	0	0
1	0	1	1	0	0	0
1	0	0	1	0	0	0
0	1	1	0	0	0	0
0	0	2	0	0	0	0
2	0	0	0	0	1	0
0	0.5	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	0.5	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Table A.8 RateStoic.m

9.00E-16	2.0	12.60	0.00
5.20E-15	-1.0	0.00	0.00
8.80E-17	0.0	4.40	0.00
9.00E-15	0.7	13.60	0.00
1.50E-13	0.0	0.00	0.50
2.50E-13	0.0	0.00	0.50
2.00E-13	0.0	5.50	0.00
4.20E-15	0.0	5.60	0.00
3.00E-16	0.0	0.00	0.00
7.10E-17	0.5	17.00	0.00
5.30E-16	0.9	20.00	0.00
2.00E-17	0.0	0.00	0.50
5.00E-14	0.0	8.40	0.00
4.50E-15	0.0	2.29	0.00
4.11E-17	0.0	0.00	0.00
8.10E-18	0.0	0.00	0.00
9.00E-15	0.7	11.60	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00
1	0.0	0.00	0.00

A.2 Input File Structures for Boron Trifluoride

Table A.9 Input File Description

File	Description
Discharge.m	Same as oxygen
CrossSections.m	Same as oxygen
EnergyLossSpecies.m	Same as oxygen
EnergyLossConstants.m	Same as oxygen
ReactStoic.m	Same as oxygen
ProductStoic.m	Same as oxygen
RateStoic.m	This file contains information necessary to assemble the rate constants in the form: $k = AT_e^B \exp\left(-\frac{C}{T_e}\right) T_g^D$. The first four columns are A, B, C, and D respectively. The fourth column (D) has a non-zero quantity only if the reaction involves only heavy particles (no electrons). There is an additional fifth column which classifies a particular reaction as either gas phase reaction or surface reaction. "Surface sink" and "surface source" reactions are surface reactions, while physical pumping and feed-gas source are classified as gas phase reactions.
AtomicComposition.m	This file contains information regarding the atomic composition of each species. The number of rows correspond to different atoms and the columns correspond to various species (excluding the electrons). For example, species BF contains one boron and one fluorine atom.

Table A.9 Input File Description

File	Description
SurfaceCoefficients.m	This file is specific for a boron trifluoride (BF_3) discharge. In order to use the BF_3 surface model for other discharges, a modified 'funk.m' has to be used. This file contains information useful in assembling "surface source" reactions only. The number of rows correspond to the number of reactions. "Surface source" produces BF_3, BF_2, BF and B species and the density fluxes coming out of the wall are given by $F_{out,BF_3} = 0.F_{in,F} + \alpha_{BF_3}F_{in,B} + 0.F_{in,B} + 0.F_{in,B} + 0.F_{in,B}$ $F_{out,BF_2} = 0.F_{in,F} + 0.F_{in,B} + \alpha_{BF_2}F_{in,B} + 0.F_{in,B} + 0.F_{in,B}$ $F_{out,BF} = 0.F_{in,F} + 0.F_{in,B} + 0.F_{in,B} + \alpha_{BF}F_{in,B} + 0.F_{in,B}$ $F_{out,B} = 0.F_{in,F} + 0.F_{in,B} + 0.F_{in,B} + 0.F_{in,B} + \alpha_{B}F_{in,B}$ $F_{out,F} = 1.F_{in,F} - 3.\alpha_{BF_3}F_{in,B} - 2.\alpha_{BF_2}F_{in,B} - 1.\alpha_{BF}F_{in,B} - 0.\alpha_{B}F_{in,B}$ Only the coefficients of the above reactions are entered in the columns corresponding to their respective reactions. The alpha fractions are chosen under the following restriction: $\alpha_{BF_3} + \alpha_{BF_2} + \alpha_{BF} + \alpha_{B} = 1$ 'Realmin' (smallest number that can be represented by a computer) is used in place of zero.

A.2.1 Sample Input Files for Boron Trifluoride

Table A.10 RateStoic.m

1.03E-15	0.445	15.374	0	1
6.71E-15	1.056	15.964	0	1
2.68E-14	0.353	10.460	0	1
2.23E-15	1.371	8.367	0	1
5.23E-14	-0.481	8.488	0	1
9.58E-15	0.824	9.618	0	1
1.21E-14	-0.494	10.550	0	1
2.63E-15	1.407	6.942	0	1
4.84E-15	0.251	33.760	0	1
1.30E-14	0.000	16.500	0	1
9.41E-16	1.008	25.190	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	1
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0
1.00E+00	0.000	0.000	0	0

Table A.11 AtomicComposition.m

1	1	1	1	0	1	1	1	1	1	0
3	2	1	0	1	3	2	1	0	0	1

Table A.12 SurfaceCoefficients.m

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	1	0	0	0
0	0	1E-308	0	0
0	0	0	1E-308	0
0	0	0	0	1E-308
1	-3	0	0	0

Appendix B

B.1 Oxygen Model

```
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           Address questions/comments to:
%
%           Kedar Patel
%           Email: kedar@poincare.eecs.berkeley.edu
%
%           (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           File:  Model.m
%           Last Updated: 9 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           Desription:
%              This is the main script that develops a volume
%              averaged model.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           Plasma Assisted Materials Processing Laboratory
%           Department of Electrical Engg. and Computer Sciences
%           University of California at Berkeley
%           188 Cory Hall
%           Berkeley, CA 94704-1774
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
clear *
clc

global ee kB R_m L_m m_Kg T_Volts Gamma_Surface scat_Xsec chg_value
global i_K n_K K_A K_B K_C K_D
global Heavy_Reactions Wall_Reactions Ion_WallReaction
global Pumping_React Source_React Neutral_WallReaction No_Electrons
global Kij_A Kij_B Kij_C Eij_A Eij_B Total_Species Loss_Species

ee= 1.6E-19;     % Electronic charge
kB= 1.38E-23;    % Boltzmann's constant in J/K
format short e

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
eval(['cd user']);

load('Discharge');              % Process condition, gas parameters
load('CrossSections');          % Cross-sections
load('ProductStoic');           % Stoichiometric coefficients of products
load('ReactStoic');             % Stoichiometric coefficients of reactants
load('RateStoic');              % Arrhenius constants of rate coefficient
load('EnergyLossConstants');    % Energy loss constants
load('EnergyLossSpecies');      % Energy loss species

eval(['cd ..']);


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

[TotalReactions, junk]= size(RateStoic);
[junk, Total_Species]= size(Discharge);     % Not including electrons
Species = SelectSpecies;                    % Subset selected
n_Species= length(Species(Species==1));     % Number of chosen species
iSpecies= logical([1 Species]);             % Always include electrons

Variables= logical([Species 1]);            % Add Te


% Find indices of reactions involving only the chosen species:
i_K=~any([ProductStoic(:,~iSpecies) ReactStoic(:,~iSpecies)]')';

junk=1:TotalReactions;
CodeFlag= junk(i_K)';   % List of reactions included in model

% Number of reactions
n_K=length(i_K(i_K==1));

% Restrict Products and Reactants tables to the chosen species
Products= ProductStoic(i_K,iSpecies);
Reactants= ReactStoic(i_K,iSpecies);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

m_Kg= Discharge(1,Species);                 % Mass
T_Volts= Discharge(2,Species).*(kB/ee);     % Temperature in volts
Gamma_Surface= Discharge(3,Species);        % Wall recomb. coeff.
chg_value= Discharge(4,Species);            % Charge

scat_Xsec= CrossSections(Species,Species);  % Scat. cross-section

K_A= RateStoic(i_K,1);
K_B= RateStoic(i_K,2);
K_C= RateStoic(i_K,3);
K_D= RateStoic(i_K,4);

Kij_A= EnergyLossConstants(:,1);
Kij_B= EnergyLossConstants(:,2);
Kij_C= EnergyLossConstants(:,3);
Eij_A= EnergyLossConstants(:,4);
Eij_B= EnergyLossConstants(:,5);

Loss_Species= EnergyLossSpecies;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Get rid of electrons from the Reactants matrix as
% T_Volts does not include electron temperature
```

```
% NoElectron is the switch to do this

No_Electrons=logical([0 ones(1,n_Species)]);

% 'tmp' gives the average temperature of ALL reactions
% This includes reactions involving heavy particles and rest.
% The non-heavy particle reactions would killed to zero by
% multiplication of 'tmp' with 'HeavyReactions'.

tmp0= Reactants(:,No_Electrons);

for i=1:n_K
 Heavy_Reactions(i)=(length(find(Reactants(i,:)'))==2 & ...
                     Reactants(i,1)==0);
 Wall_Reactions(i)=(length(find(Reactants(i,:)'))==1 & ...
                    length(find(Products(i,:)'))==1);
 Ion_WallReaction(i)=(length(find(tmp0(i,chg_value>0)'))==1 & ...
                   Wall_Reactions(i));
 Pumping_React(i)=(length(find(Reactants(i,:)'))==1 & ...
                   length(find(Products(i,:)'))==0);
 Source_React(i)= (length(find(Reactants(i,:)'))==0 & ...
                 length(find(Products(i,:)'))==1);
end
Neutral_WallReaction= ~Ion_WallReaction & Wall_Reactions;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

disp(' '); disp(' ');
disp('   Cylindrical Reactor Geometry:-')
R_m= input('      Enter Radius in METERS: ');
L_m= input('      Enter Length in METERS: ');

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

end_it=0;
while ~end_it                       % Proceed only if QUIT is NOT selected
   [selexn, end_it]=MainMenu;       % Query the user

   if selexn==1

      OneTimeRun(Variables,Reactants,Products,Species);

   elseif selexn==2

      InputPower(Variables,Reactants,Products,Species);

   elseif selexn==3

      FlowPressure(Variables,Reactants,Products,Species);

   elseif selexn==4

      close all;                    % Close all open figures

   end
   status= fclose('all');           % Close all open files
end                                 % Break free if QUIT is selected
close all;                          % Close all open figures
copyright;                          % Display copyright and contact info
clear
clc
```

```
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Address questions/comments to:
%
%            Kedar Patel
%            Email: kedar@poincare.eecs.berkeley.edu
%
%            (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            File:  OneTimeRun.m
%            Last Updated: 14 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Desription:
%               OneTimeRun allows user to study a specific processing conditon.
%               A specific run can be saved as a estimate file for future
%               sweeping runs.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function ans = OneTimeRun(Variables,Reactants,Products,Species)

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

options(1)=1;              % Display the rootfinding results
options(5)=1;              % 0=Gauss-Newton, 1=Levenberg-Marquardt
options(14)=1E20;             % Maximum number of iterations
rite=0;

try_again=1;
while try_again
  clc
  disp(' ');
  disp('One-Time-Run');
  disp(' ');
  Pabs = input('   Enter the Power Absorbed in WATTS: ');
  P =    input('   Enter the Reactor Pressure in TORR: ');
  Qsccm= input('   Enter the Gas Flow Rate in SCCM: ');
  NF=    input('   Enter a Normalization Factor (1E17): ');

  tmp=GetFile('Load Estimate File');
  if isempty(tmp)
     break
  end

  disp(' ');
  disp('   Computation in progress...Please be patient...');
  disp(' ');
  ini=(tmp(Variables',1))';       % Load starting estimate
  clear tmp;
  iniX= gauge(ini,NF,0);          % Normalize all the densities
  tic
  X=fsolve('funk',iniX,options,[],Pabs,P,Qsccm,NF,Reactants,Products,Species);
  toc
  query_save=1;
  while query_save
        disp(' ');
        query2 = input('   Would you like to save this run? (y/n): ', 's');
        if (query2=='y'|query2=='Y')
```

```
          query_save= 0;
          disp(' '); rite=(gauge(X,NF,1))';
           eval(['cd estimates']);
          file=  input('   Enter the output file name: ','s');
          fid = fopen(file, 'w');
          fprintf(fid, ' %9.3e\n', rite);
          fclose(fid);
           eval(['cd ..']);
       elseif (query2=='n'|query2=='N')
          query_save= 0;
       else
          query_save=1;
       end
  end

  query_again=1;
  while query_again
       disp(' ');
       query1 = input('   Would you like another One-Time-Run? (y/n): ', 's');
       if (query1=='y'|query1=='Y')
          query_again= 0; try_again=1;
       elseif (query1=='n'|query1=='N')
          query_again= 0; try_again=0;
       else
          query_again=1;
       end
  end
end


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  Funk.m
%             Last Updated: 2 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Desription:
%                For a non-linear system F(X)=0, this function returns value of
%                F. The system of equation for the volume averaged model
%                includes particle balance equations for each species and a
%                power balance equation.
%
%                The particle balance equation for electrons is introduced using
%                charge
%                neutrality.
%
%                The particle balance equations are self-assembled and are
%                solved for steady-state condition.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function F = funk(iniX, Pabs, P, Qsccm, NF, Reactants, Products, Species)
```

```
global ee kB R_m L_m m_Kg T_Volts Gamma_Surface scat_Xsec chg_value
global i_K n_K K_A K_B K_C K_D
global Heavy_Reactions Wall_Reactions Ion_WallReaction
global Pumping_React Source_React Neutral_WallReaction No_Electrons
global Kij_A Kij_B Kij_C Eij_A Eij_B Total_Species Loss_Species

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Extract densities and electron temperature as separate vectors
n= iniX(1:(length(iniX)-1)).*NF;
Te= iniX(length(iniX));

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ng= n(chg_value==0);    % Extract the neutral density
ni= n(chg_value>0);     % Extract the positive ion density
nneg= n(chg_value<0);   % Extract the negative ion density
ne= sum(ni)-sum(nneg);  % Extract the electron density

mg= m_Kg(chg_value==0); % Extract the neutral mass
mi= m_Kg(chg_value>0);  % Extract the positive ion mass
me= 9.11E-31;           % Electron mass in Kg

Ti=T_Volts(chg_value>0);   % Extract the positive ion temperature

% X-section for neutral-ion collisions
scatgi= scat_Xsec(chg_value==0, chg_value>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
R=R_m; L=L_m;
V =pi*R^2*L;                          % Reactor volume in cubic meters
A=2*pi*R*L;                           % Actual area
leff=((pi/L)^2+(2.405/R)^2)^(-0.5);   % Eff. diffusion length
QtorrLit= Qsccm/79.05;                % SCCM to Torr-Liter/sec
Qmolec= 4.483E17*Qsccm;               % SCCM to molecules/sec
kr= QtorrLit/(P*V*1000);              % Pumping rate coefficient

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ve= (8*ee*Te/(pi*me))^0.5;            % Mean electron velocity
alpha= sum(nneg)/ne;                  % Measure of electronegativity
Gamma= sum((Te.*ni)./Ti)/sum(ni);     % Density weighted Te/Ti

% Bohm velocity of positive ions
Ubi=((ee*Te*(1+alpha))./(mi.*(1+alpha*Gamma))).^0.5;

lambda= 1./(ng*scatgi);               % Mean free path for each ion
lambdai= sum(ni.*lambda)./sum(ni);    % Density weighted ion mean free path

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
N= 20;        % Number of magnetic cusps
Bo= 2.5E3;    % Cusp field strength

rce= 3.37E-2*(Te^0.5)/Bo;
tmp_rci=1.44.*((Ti.*(mi./1.67e-27)).^0.5)./Bo;
rci= sum(ni.*tmp_rci)./sum(ni);
lambdae=ve/(sum(ng)*1e-13);
w=4*((rce*rci)^0.5)*(1 + R/(N*((lambdae*lambdai)^0.5)) );
floss=min([ N*w/(2*pi*R) 1]);

mag_confine=1;   %<-----------------  Enter 1 to turn on, 0 to turn off

if mag_confine
   % Axial and radial scaling factors
   hL= (0.86/(1+alpha))*( 3 + L/(2*lambdai) )^(-0.5);
```

```
   hRw= floss*(0.80/(1+alpha))*( 0.64 + floss*(3.36 + R/lambdai) )^(-0.5);
   hLw= floss*(0.86/(1+alpha))*( 0.74 + floss*(2.26 + L/(2*lambdai)) )^(-0.5);
   Aeff= pi*R*(R*hL + R*hLw + 2*L*hRw);
else
   % Axial and radial scaling factors
   hL= (0.86/(1+alpha))*( 3 + L/(2*lambdai) )^(-0.5);
   hR= (0.8/(1+alpha))*( 4 + (R/lambdai) )^(-0.5);
   Aeff= 2*pi*R*(R*hL + L*hR);
end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
phi= -Te*log(4*(ni*Ubi')/(ne*ve));   % Wall potential
Ei= phi + Te/2;                      % Ion energy lost to the wall
Ee= 2*Te;                            % Electron energy lost to the wall

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
kkp0= Reactants(:,No_Electrons);
kkp=zeros(size(kkp0));
kkp(~kkp0==0)=1;
kkp1=sum((ones(n_K,1)*T_Volts.*kkp)')./2;

kkp2= (Heavy_Reactions.*kkp1)';   % Kill reactions NOT involving Heavy particles
kkp3=kkp2.*(ee/(300*kB));         % Prepare multiplier; Replace '0's with '1's
kkp4= kkp3+(~kkp3>0);             % Multiplier has the form (300/T);
HeavyMultiplier=(1./kkp4).^K_D;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% The following statement calculates the bohm velocities of ALL selected
% species excluding electrons.

VirtualUbi=((ee*Te*(1+alpha))./(m_Kg.*(1+alpha*Gamma))).^0.5;
kkp5=sum((ones(n_K,1)*VirtualUbi.*kkp)');
kkp6= (Ion_WallReaction.*kkp5)';     % Kill reactions NOT involving ions
kkp7= kkp6.*(Aeff/V);
IonWallMultiplier= kkp7+(~kkp7>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Virtual mean neutral velocity
Virtualvo= ((8.*ee.*T_Volts)./(pi.*m_Kg)).^0.5;

kkp8=zeros(size(scat_Xsec));
kkp8(chg_value==0,chg_value==0)= ones(size(kkp8(chg_value==0,chg_value==0)));
kkp9= kkp8.*scat_Xsec;
kkp9(:,~chg_value==0)= ones(size(kkp9(:,~chg_value==0)));
scatg= kkp9;

% The scatg calculated above will have the correct cross-sections for the
% indices (charge=0,charge=0). It will have zeros in indices
% (~ charge=0, charge=0) so that the densities other than the neutrals
% don't contribute in calculation of mean free path.
% All the remaining places will have '1's.  These will get weeded out later.

VirtualDoo= (ee.*T_Volts)./(m_Kg.*Virtualvo.*(n*scatg));

% The line below replaces all '0's with a really small number in the wall
% recombination coefficient. It leaves the user entered values unchanged
% but forces all zeros to become the smallest number that the computer can
% represent i.e "realmin".  This has to be done because the computation of
% 'kkp10' involves division by 'GammaSurface'.

GammaSurface=Gamma_Surface;
GammaSurface(Gamma_Surface==0)=realmin;
```

```
kkp10=((leff^2)./VirtualDoo +((4*V)./GammaSurface-2*V)./(A.*Virtualvo)).^(-1);
kkp11=sum( (ones(n_K,1)*kkp10.*kkp)' );
kkp12= (Neutral_WallReaction.*kkp11)';
NeutralWallMultiplier=kkp12+(~kkp12>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
kkp13= (Pumping_React.*kr)';
PumpingMultiplier=kkp13+(~kkp13>0);

kkp14=(Source_React.*(Qmolec/V))';
SourceMultiplier=kkp14+(~kkp14>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
k1= K_A.*((Te).^K_B).*exp(-K_C./Te).*HeavyMultiplier.*IonWallMultiplier;
k2=  NeutralWallMultiplier.*PumpingMultiplier.*SourceMultiplier;

RateConstants= k1.*k2;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Kij= Kij_A.*(Te.^Kij_B).*exp(-Kij_C./Te);
Eij= Eij_A.*(Te.^Eij_B);

kkp15= ((ones(Total_Species,1)*(Kij'))').*Loss_Species;
kkp16= ((ones(Total_Species,1)*(Eij'))').*Loss_Species;
kkp17= sum(kkp15.*kkp16)';
Kij_Eij = kkp17(Species);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Density=[ne n];

% Find the term associated with each reaction:
tt=(ones(n_K,1)*Density.*Reactants)';
Terms=prod(tt+(~tt>0))'.*RateConstants;

% Assemble the RHS of each particle balance equation:
ff=(Products-Reactants)'*Terms;
F=ff(2:length(ff)); % RHS of all particle balance except electrons
F= F./NF;        % Normalize the F-vector

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Power Balance Equation: Volume and Surface electron energy loss
F(length(F)+1)=  Pabs - ee*ne*V*(n*Kij_Eij) - ee*(Ubi*ni')*Aeff*(Ei+Ee) ;



%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  Gauge.m
%             Last Updated: 26 February 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Desription:
```

```
%               This function normalizes or unnormalizes the variables to be
%               passed to
%               'Funk.m'.  Normalization is necessary to prevent ill-
%               conditioning of gradient matrices.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function out= gauge(in, NF, d)

size_in = length(in);
if d==0
   out= in./NF;  % Normalize all the densities
elseif d==1
   out= in.*NF;  % Un-normalize all the densities
end
out(size_in)= in(size_in); % Leave electron temp alone
```

```
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  Extract.m
%             Last Updated: 26 February 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Desription:
%                Calculates additional plasma parameters and formats output.
%
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function Xi = extract(Pabs, P, Qsccm, X, NF)
global ee kB m_Kg chg_value T_Volts

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
tmp= gauge(X,NF,1);
size_X = length(X);
n= tmp(1:(size_X-1));
Te= tmp(size_X);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ng= n(chg_value==0);    % Extract the neutral density vector
ni= n(chg_value>0);     % Extract the positive ion density vector
nneg= n(chg_value<0);   % Extract the negative ion density vector
ne= sum(ni)-sum(nneg);  % Extract the electron density

mi= m_Kg(chg_value>0);     % Extract the positive ion mass
Ti=T_Volts(chg_value>0);   % Extract the positive ion temperature

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
n_neutral= sum(ng);     % Total neutral density
```

```
n_plus= sum(ni);        % Total positive ion density
n_minus= sum(nneg);     % Total negative ion density
alpha= n_minus/ne;      % Measure of electronegativity
x00= ng./n_neutral;              % Fractional concentration of neutrals
xii= ni./n_plus;                 % Fractional concentration of positive ions
xiz= n_plus/(n_plus + n_neutral);% Fractional Ionization

Gamma= sum((Te.*ni)./Ti)/sum(ni); % Density weighted Te/Ti
Ubi=((ee*Te*(1+alpha))./(mi.*(1+alpha*Gamma))).^0.5;

ionflux= Ubi.*ni;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Xi= [ Pabs P Qsccm ne n_neutral n_plus n_minus alpha xiz tmp x00 xii ionflux];



%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  GetVariable.m
%             Last Updated: 14 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Desription:
%                This function allows user to interactivally select and load
%                data files as matrices into the current workspace.
%
%                This file need to be modified if used on a Macitosh operating
%                system.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function data = GetVariable(dispmsg)

eval(['cd estimates'])

[file,path] = uigetfile('*.m',dispmsg);

if isstr(file)
   eval(['load ',strrep(file,'.m','')]);
end

if ~path==0
   data= eval(strrep(file,'.m',''));
elseif path==0
   data=[];
end
eval(['cd ..'])
clear file path

%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
```

```
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  Copyright.m
%             Last Updated: 26 February 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Desription:
%                Copyright notice and contact information.
%
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function ans = copyright()

disp(' ');
disp(' ');
disp('*********************************************************************')
disp('           Address questions/comments to:');
disp(' ');
disp('           Kedar Patel');
disp('           Email: kedar@poincare.eecs.berkeley.edu');
disp(' ');
disp('           (c) University of California 1997, 1998');
disp(' ');
disp('*********************************************************************')
disp(' ');
disp(' ');


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  FlowPressure.m
%             Last Updated: 2 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Desription:
%                Sweeps feed gas flow-rate, reactor pressure, and pumping.
%                Two of the three parameters are held fixed for each sweep.
%
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function iXi = FlowPressure(Variables,Reactants, Products,Species)
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
options(1)=0;              % Display the rootfinding results
options(5)=0;              % 0=Gauss-Newton, 1=Levenberg-Marquardt
options(14)=1E20;          % Maximum number of iterations

clc
header0= ('Dependence on Reactor Pressure/Gas Flow');
choice1= ('1)  Sweep Reactor Pressure, Variable Flow Rate, Constant Pumping');
choice2= ('2)  Sweep Reactor Pressure, Variable Pumping, Constant Flow Rate');
choice3= ('3)  Sweep Flow Rate, Variable Pumping, Constant Reactor Pressure');

move_on=0;
while ~move_on
  disp(' ');
  disp(header0);
  disp(' ');
  disp(choice1);
  disp(choice2);
  disp(choice3);
  disp(' ');
  sweep = input('   Please make your your selection: ');
  if ((sweep==1)|(sweep==2))
     move_on=1;
  else
     move_on=0;
     clc
  end
end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
clc
disp(' ');
disp(header0);
disp(' ');
file=  input('   Enter the output file name: ','s');
fid = fopen(['Output/' file], 'w');
NF=    input('   Enter a Normalization Factor (1E17): ');
Pabs = input('   Enter the Power Absorbed in WATTS: ');
disp(' ');

if sweep==1
   disp('   Calibration of Pumping')
   cal_Q= input('      Enter Calibration Flow-rate in SCCM: ');
   cal_P= input('      Enter Calibration Pressure in Torr: ');
   disp(' ');
   disp('   Sweeping Range:');
   disp(' ');
   Pmin= input('           Enter the MINIMUM Pressure in TORR: ');
   Pmax= input('           Enter the MAXIMUM Pressure in TORR: ');
   P= logspace(log10(Pmin), log10(Pmax), 10);
   disp(' ');
   tmp=GetFile('Load Estimate File');

   if ~isempty(tmp)

   ini=(tmp(Variables',1))';     % Load starting estimate
   clear tmp;
   iniX= gauge(ini,NF,0);        % Normalize all the densities
   for i = 1:length(P)
      cycle=sprintf('   Computation in progress...[%i] of [%i]',i,length(P));
      disp(' '); disp(cycle);
   Qsccm= P(i)*(cal_Q/cal_P);
   X=fsolve('funk',iniX,options,[],Pabs,P(i),Qsccm,NF,Reactants,Products, ...
```

```
            Species);
    iniX=X;
    Xi= Extract(Pabs, P(i), Qsccm, X, NF);
    iXi(i,:)= Xi;
    fprintf(fid, '%12.4e', Xi);
    fprintf(fid, '\n');
    end

    end

elseif sweep==2
    Qsccm= input('   Enter the Gas Flow Rate in SCCM: ');
    disp(' ');
    disp('   Sweeping Range:');
    disp(' ');
    Pmin= input('           Enter the MINIMUM Pressure in TORR: ');
    Pmax= input('           Enter the MAXIMUM Pressure in TORR: ');
    P= logspace(log10(Pmin), log10(Pmax), 10);
    disp(' ');
    tmp=GetFile('Load Estimate File');

    if ~isempty(tmp)

    ini=(tmp(Variables',1))';      % Load starting estimate
    clear tmp;
    iniX= gauge(ini,NF,0);         % Normalize all the densities
    for i = 1:length(P)
       cycle=sprintf('   Computation in progress...[%i] of [%i]',i,length(P));
       disp(' '); disp(cycle);
    X=fsolve('funk', iniX, options,[],Pabs,P(i),Qsccm,NF,Reactants,Products,...
            Species);
    iniX=X;
    Xi= Extract(Pabs, P(i), Qsccm, X, NF);
    iXi(i,:)= Xi;
    fprintf(fid, '%12.4e', Xi);
    fprintf(fid, '\n');
    end

    end

elseif sweep==3
    P =    input('   Enter the Reactor Pressure in TORR: ');
    disp(' ');
    disp('   Sweeping Range:');
    Qmin= input('           Enter the MINIMUM Flow-rate in SCCM: ');
    Qmax= input('           Enter the MAXIMUM Flow-rate in SCCM: ');
    Qsccm= logspace(log10(Qmin), log10(Qmax), 10 );
    disp(' ');
    tmp=GetFile('Load Estimate File');

    if ~isempty(tmp)

    ini=(tmp(Variables',1))';  % Load starting estimate
    clear tmp;
    iniX= gauge(ini,NF,0);     % Normalize all the densities
    for i = 1:length(Qsccm)
       cycle=sprintf('Computation in progress...[%i] of [%i]',i,length(Qsccm));
       disp(' '); disp(cycle);
    X=fsolve('funk',iniX,options,[],Pabs,P,Qsccm(i),NF,Reactants,Products,...
            Species);
    iniX=X;
    Xi= Extract(Pabs, P, Qsccm(i), X, NF);
    iXi(i,:)= Xi;
```

```
   fprintf(fid, '%12.4e', Xi);
   fprintf(fid, '\n');
   end

   end

end

PlotFlowPressure(iXi);


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Address questions/comments to:
%
%            Kedar Patel
%            Email: kedar@poincare.eecs.berkeley.edu
%
%            (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            File:  InputPower.m
%            Last Updated: 26 February 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Desription:
%               Sweeps the power absorbed by the plasma for fixed feed gas
%               flow-rate, reactor pressure, and pumping.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function iXi = InputPower(Variables,Reactants, Products,Species)

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
options(1)=0;              % Display the rootfinding results
options(5)=0;              % 0=Gauss-Newton, 1=Levenberg-Marquardt
options(14)=1E20;            % Maximum number of iterations
clc
header= ('Dependence on Absorbed Power');
disp(header);
disp(' ');
file= input('   Enter the output file name: ','s');
fid = fopen(['Output/' file], 'w');
NF=    input('   Enter a Normalization Factor (1E17): ');
Qsccm= input('   Enter the Gas Flow Rate in SCCM: ');
P =    input('   Enter the Reactor Pressure in TORR: ');
disp(' ');
disp('   Sweeping Range:');
disp(' ');
Pabs_min= input('           Enter the MINIMUM Power in WATTS: ');
Pabs_max= input('           Enter the MAXIMUM Power in WATTS: ');
Pabs= logspace(log10(Pabs_min), log10(Pabs_max), 10);
disp(' ');

tmp=GetFile('Load Estimate File');

if ~isempty(tmp)

   ini=(tmp(Variables',1))';     % Load starting estimate
   clear tmp;
```

```
   iniX= gauge(ini,NF,0);         % Normalize all the densities

   for i = 1:length(Pabs)
   cycle=sprintf('   Computation in progress...[%i] of [%i]',i,length(Pabs));
   disp(' '); disp(cycle);
   X=fsolve('funk',iniX,options,[],Pabs(i),P,Qsccm,NF,Reactants,Products,...
            Species);
   iniX=X;
   Xi= Extract(Pabs(i), P, Qsccm, X, NF);
   iXi(i,:)= Xi;
   fprintf(fid, ' %9.3e', Xi);
   fprintf(fid, '\n');
   end
   PlotPower(iXi);

end


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Address questions/comments to:
%
%            Kedar Patel
%            Email: kedar@poincare.eecs.berkeley.edu
%
%            (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            File:  SelectSpecies.m
%            Last Updated: 26 February 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Desription:
%               This function allows the user to use a restricted set of
%               species in the volume averaged model. Only the reactions
%               involving the selected species are used in the model.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function Species = SelectSpecies()

global Total_Species

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
done=0;
while ~done
   clc
   disp(' '); disp(' ');
   disp('   For each of the species,')
   disp('   Enter ''1'' to turn ON a species, ''0'' to turn it OFF.');
   disp(' ');
   disp('   For example, consider species [ nO2 nO nO* nO2+ nO+ nO- ].')
   disp('   A input sequence ''[1 1 1 1 1 0]'' will turn OFF species ''O-''.');
   disp(' ');
   disp('   To turn ALL species ON, simply enter ''1''.');
   disp(' ');
   Species= input('   Species: ');
   disp(' ');
   err=0;
   if Species==1
```

```
      Species= ones(1,Total_Species); done=1;
   else
   if ~(length(Species)==Total_Species)
       disp('   Error! You are missing an entry!');
       disp(' ');
       disp('   Strike any key when ready ...');
       pause
       err=1;
   else
       for i=1:length(Species)
           if ~(Species(i)==0 | Species(i)==1)
               disp('   Error! Enter "1" or "0" only!');
               err=1;
               disp(' ');
               disp('   Strike any key when ready ...');
               pause
               break
           end
       end
   end
   if ~err
       done=1;
      end
   end
end

Species= logical(Species);


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Address questions/comments to:
%
%            Kedar Patel
%            Email: kedar@poincare.eecs.berkeley.edu
%
%            (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            File:  PlotMenu.m
%            Last Updated: 26 February 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Desription:
%               This function allows the user to decide which parameter is
%               plotted on the X-axis following a flow-rate or reactor pressure
%               sweep.
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function plot_sel = PlotMenu()

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
plot_menu=('   Plot Menu');
header=   ('   You have the following choices: ');
choice1=  ('   1)  Plot Flow Rate');
choice2=  ('   2)  Plot Reactor Pressure');
choice3=  ('   3)  Return to the Main Menu');

move_on=0;
```

```
while ~move_on
  clc
  disp(' ');
  disp(plot_menu);
  disp(' ');
  disp(header);
  disp(' ');
  disp(choice1);
  disp(choice2);
  disp(choice3);
  disp(' ');
  plot_sel = input('   Please make your your selection: ');
  if((plot_sel==1)|(plot_sel==2)|(plot_sel==3))
      move_on=1;
  else
      move_on=0;
      clc
  end
end


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  PlotFlowPressure.m
%             Last Updated: 26 February 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function ans = PlotFlowPressure(iXi)

plot_sel=PlotMenu;
if plot_sel==1

    figure(1)
    loglog(iXi(:,3), iXi(:,4))
    xlabel('Flow Rate Q [SCCM]')
    ylabel('Electron Density n_e [m^{-3}]')

    figure(2)
    loglog(iXi(:,3), iXi(:,5))
    xlabel('Flow Rate Q [SCCM]')
    ylabel('Total Neutral Density n_g [m^{-3}]')

    figure(3)
    loglog(iXi(:,3), iXi(:,6))
    xlabel('Flow Rate Q [SCCM]')
    ylabel('Total Positive Ion Density n_+ [m^{-3}]')

    figure(4)
    loglog(iXi(:,3), iXi(:,7))
    xlabel('Flow Rate Q [SCCM]')
    ylabel('Total Negative Ion Density n_- [m^{-3}]')

    figure(5)
```

```
loglog(iXi(:,3), iXi(:,8))
xlabel('Flow Rate Q [SCCM]')
ylabel('\alpha')

figure(6)
loglog(iXi(:,3), iXi(:,9))
xlabel('Flow Rate Q [SCCM]')
ylabel('Fractional Ionization {\chi}_{iz}')

figure(7)
loglog(iXi(:,3), iXi(:,10))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{O_2} [m^{-3}]')

figure(8)
loglog(iXi(:,3), iXi(:,11))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{O} [m^{-3}]')

figure(9)
loglog(iXi(:,3), iXi(:,12))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{O^*} [m^{-3}]')

figure(10)
loglog(iXi(:,3), iXi(:,13))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{{O_2}^+} [m^{-3}]')

figure(11)
loglog(iXi(:,3), iXi(:,14))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{O^+} [m^{-3}]')

figure(12)
loglog(iXi(:,3), iXi(:,15))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{O^-} [m^{-3}]')

figure(13)
loglog(iXi(:,3), iXi(:,16))
xlabel('Flow Rate Q [SCCM]')
ylabel('Electron Temperature T_e [eV]')

figure(14)
loglog(iXi(:,3), iXi(:,17))
xlabel('Flow Rate Q [SCCM]')
ylabel('O_2 Fraction')

figure(15)
loglog(iXi(:,3), iXi(:,18))
xlabel('Flow Rate Q [SCCM]')
ylabel('O Fraction')

figure(16)
loglog(iXi(:,3), iXi(:,19))
xlabel('Flow Rate Q [SCCM]')
ylabel('O^* Fraction')

figure(17)
loglog(iXi(:,3), iXi(:,20))
xlabel('Flow Rate Q [SCCM]')
ylabel('{O_2}^+ Fraction')
```

```
    figure(18)
    loglog(iXi(:,3), iXi(:,21))
    xlabel('Flow Rate Q [SCCM]')
    ylabel('O^+ Fraction')

    figure(19)
    loglog(iXi(:,3), iXi(:,23)./iXi(:,22))
    xlabel('Flow Rate Q [SCCM]')
    ylabel('n_{O^+}/n_{(O_2)^+}')

elseif plot_sel==2

    figure(1)
    loglog(iXi(:,2), iXi(:,4))
    xlabel('Pressure P [Torr]')
    ylabel('Electron Density n_e [m^{-3}]')

    figure(2)
    loglog(iXi(:,2), iXi(:,5))
    xlabel('Pressure P [Torr]')
    ylabel('Total Neutral Density n_g [m^{-3}]')

    figure(3)
    loglog(iXi(:,2), iXi(:,6))
    xlabel('Pressure P [Torr]')
    ylabel('Total Positive Ion Density n_+ [m^{-3}]')

    figure(4)
    loglog(iXi(:,2), iXi(:,7))
    xlabel('Pressure P [Torr]')
    ylabel('Total Negative Ion Density n_- [m^{-3}]')

    figure(5)
    loglog(iXi(:,2), iXi(:,8))
    xlabel('Pressure P [Torr]')
    ylabel('\alpha')

    figure(6)
    loglog(iXi(:,2), iXi(:,9))
    xlabel('Pressure P [Torr]')
    ylabel('Fractional Ionization {\chi}_{iz}')

    figure(7)
    loglog(iXi(:,2), iXi(:,10))
    xlabel('Pressure P [Torr]')
    ylabel('n_{O_2} [m^{-3}]')

    figure(8)
    loglog(iXi(:,2), iXi(:,11))
    xlabel('Pressure P [Torr]')
    ylabel('n_{O} [m^{-3}]')

    figure(9)
    loglog(iXi(:,2), iXi(:,12))
    xlabel('Pressure P [Torr]')
    ylabel('n_{O^*} [m^{-3}]')

    figure(10)
    loglog(iXi(:,2), iXi(:,13))
    xlabel('Pressure P [Torr]')
    ylabel('n_{(O_2)^+} [m^{-3}]')
```

```
    figure(11)
    loglog(iXi(:,2), iXi(:,14))
    xlabel('Pressure P [Torr]')
    ylabel('n_{O^+} [m^{-3}]')

    figure(12)
    loglog(iXi(:,2), iXi(:,15))
    xlabel('Pressure P [Torr]')
    ylabel('n_{O^-} [m^{-3}]')

    figure(13)
    loglog(iXi(:,2), iXi(:,16))
    xlabel('Pressure P [Torr]')
    ylabel('Electron Temperature T_e [eV]')

    figure(14)
    loglog(iXi(:,2), iXi(:,17))
    xlabel('Pressure P [Torr]')
    ylabel('O_2 Fraction')

    figure(15)
    loglog(iXi(:,2), iXi(:,18))
    xlabel('Pressure P [Torr]')
    ylabel('O Fraction')

    figure(16)
    loglog(iXi(:,2), iXi(:,19))
    xlabel('Pressure P [Torr]')
    ylabel('O^* Fraction')

    figure(17)
    loglog(iXi(:,2), iXi(:,20))
    xlabel('Pressure P [Torr]')
    ylabel('{O_2}^+ Fraction')

    figure(18)
    loglog(iXi(:,2), iXi(:,21))
    xlabel('Pressure P [Torr]')
    ylabel('O^+ Fraction')

    figure(19)
    loglog(iXi(:,2), iXi(:,14)./iXi(:,13))
    xlabel('Pressure P [Torr]')
    ylabel('n_{O^+}/n_{{O_2}^+}')

end


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  PlotPower.m
%             Last Updated: 26 February 1998
%
```

```
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
function ans = PlotPower(iXi)

    figure(1)
    loglog(iXi(:,1), iXi(:,4))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Electron Density n_e [m^{-3}]')

    figure(2)
    loglog(iXi(:,1), iXi(:,5))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Total Neutral Density n_g [m^{-3}]')

    figure(3)
    loglog(iXi(:,1), iXi(:,6))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Total Positive Ion Density n_+ [m^{-3}]')

    figure(4)
    loglog(iXi(:,1), iXi(:,7))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Total Negative Ion Density n_- [m^{-3}]')

    figure(5)
    loglog(iXi(:,1), iXi(:,8))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('\alpha')

    figure(6)
    loglog(iXi(:,1), iXi(:,9))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Fractional Ionization {\chi}_{iz}')

    figure(7)
    loglog(iXi(:,1), iXi(:,10))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{O_2} [m^{-3}]')

    figure(8)
    loglog(iXi(:,1), iXi(:,11))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{O} [m^{-3}]')

    figure(9)
    loglog(iXi(:,1), iXi(:,12))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{O^*} [m^{-3}]')

    figure(10)
    loglog(iXi(:,1), iXi(:,13))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{{O_2}^+} [m^{-3}]')

    figure(11)
    loglog(iXi(:,1), iXi(:,14))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{O^+} [m^{-3}]')

    figure(12)
    loglog(iXi(:,1), iXi(:,15))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{O^-} [m^{-3}]')
```

```
figure(13)
loglog(iXi(:,1), iXi(:,16))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('Electron Temperature T_e [eV]')

figure(14)
loglog(iXi(:,1), iXi(:,17))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('O_2 Fraction')

figure(15)
loglog(iXi(:,1), iXi(:,18))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('O Fraction')

figure(16)
loglog(iXi(:,1), iXi(:,19))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('O^* Fraction')

figure(17)
loglog(iXi(:,1), iXi(:,20))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('{O_2}^+ Fraction')

figure(18)
loglog(iXi(:,1), iXi(:,21))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('O^+ Fraction')

figure(19)
loglog(iXi(:,1), iXi(:,14)./iXi(:,13))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{O^+}/n_{{O_2}^+}')
```

B.2 Boron Trifluoride Model

```
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           Address questions/comments to:
%
%           Kedar Patel
%           Email: kedar@poincare.eecs.berkeley.edu
%
%           (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           File:  Model.m
%           Last Updated: 26 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
clear all
clc

global ee kB R_m L_m m_Kg T_Volts Gamma_Surface scat_Xsec chg_value
global i_K n_K K_A K_B K_C K_D
global Heavy_Reactions Pumping_React Source_React No_Electrons
global Kij_A Kij_B Kij_C Eij_A Eij_B Total_Species Loss_Species
global Ion_Surface Neutral_SurfacePumping Neutral_SurfaceSource
```

```
global Atomic_Composition
global Surface_Coeff

ee= 1.6E-19;     % Electronic charge
kB= 1.38E-23;    % Boltzmann's constant in J/K
format short e

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
eval(['cd user']);

load('Discharge');            % Process condition, gas parameters
load('CrossSections');        % Cross-sections
load('ProductStoic');         % Stoichiometric coefficients of products
load('ReactStoic');           % Stoichiometric coefficients of reactants
load('RateStoic');            % Arrhenius constants of rate coefficient
load('EnergyLossConstants');  % Energy loss constants
load('EnergyLossSpecies');    % Energy loss species
load('AtomicComposition');    % Atomic composition
load('SurfaceCoefficients');  % Alpha values

eval(['cd ..']);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

[TotalReactions, junk]= size(RateStoic);
[junk, Total_Species]= size(Discharge);     % Not including electrons
Species = SelectSpecies;                    % Subset selected
n_Species= length(Species(Species==1));     % Number of chosen species
iSpecies= logical([1 Species]);             % Always include electrons

Variables= logical([Species 1]);            % Add Te


% Find indices of reactions involving only the chosen species:
i_K=~any([ProductStoic(:,~iSpecies) ReactStoic(:,~iSpecies)]')';

junk=1:TotalReactions;
CodeFlag= junk(i_K)';  % List of reactions included in model

% Number of reactions
n_K=length(i_K(i_K==1));

% Restrict Products and Reactants tables to the chosen species
Products= ProductStoic(i_K,iSpecies);
Reactants= ReactStoic(i_K,iSpecies);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

m_Kg= Discharge(1,Species);                 % Mass
T_Volts= Discharge(2,Species).*(kB/ee);     % Temperature in volts
Gamma_Surface= Discharge(3,Species);        % Wall recombination coeff.
chg_value= Discharge(4,Species);            % Charge

scat_Xsec= CrossSections(Species,Species);  % Scattering cross-section

K_A= RateStoic(i_K,1);
K_B= RateStoic(i_K,2);
K_C= RateStoic(i_K,3);
K_D= RateStoic(i_K,4);

GasPhaseReactions= logical(RateStoic(i_K,5))';

Kij_A= EnergyLossConstants(:,1);
```

```
Kij_B= EnergyLossConstants(:,2);
Kij_C= EnergyLossConstants(:,3);
Eij_A= EnergyLossConstants(:,4);
Eij_B= EnergyLossConstants(:,5);

Loss_Species= EnergyLossSpecies;
Atomic_Composition= AtomicComposition(:,Species);
Surface_Coeff=SurfaceCoefficients;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Get rid of electrons from the Reactants matrix as
% T_Volts does not include electron temperature
% NoElectron is the switch to do this

No_Electrons=logical([0 ones(1,n_Species)]);

% 'tmp' gives the average temperature of ALL reactions
% This includes reactions involving heavy particles and rest.
% The non-heavy particle reactions would killed to zero by
% multiplication of 'tmp' with 'HeavyReactions'.

tmp0= Reactants(:,No_Electrons);

Heavy_Reactions=0;
Neutral_SurfacePumping=0;
Ion_Surface=0;
Pumping_Class=0;
Source_Class=0;
Pumping_React=0;
Neutral_SurfaceSource=0;
Source_React=0;

for i=1:n_K
 Heavy_Reactions(i)=(length(find(Reactants(i,:)'))==2 & Reactants(i,1)==0);
 Pumping_Class(i)=(length(find(Reactants(i,:)'))==1 & ...
                   length(find(Products(i,:)'))==0);
 Source_Class(i)= (length(find(Reactants(i,:)'))==0 & ...
                  length(find(Products(i,:)'))==1);
 Ion_Surface(i)=(length(find(tmp0(i,chg_value>0)'))==1 & ...
                  length(find(Products(i,:)'))>=1 & (~GasPhaseReactions(i)));
 Neutral_SurfacePumping(i)=(length(find(tmp0(i,chg_value==0)'))==1 & ...
                            Pumping_Class(i) & (~GasPhaseReactions(i)));

end
Pumping_React= Pumping_Class & GasPhaseReactions;
Neutral_SurfaceSource= Source_Class & (~GasPhaseReactions);
Source_React= Source_Class & (~Neutral_SurfaceSource);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

disp(' '); disp(' ');
disp('   Cylindrical Reactor Geometry:-')
R_m= input('      Enter Radius in METERS: ');
L_m= input('      Enter Length in METERS: ');

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
end_it=0;
while ~end_it                          % Proceed only if QUIT is NOT selected
   [selexn, end_it]=MainMenu;     % Query the user

   if selexn==1     %-------------------------------------------------------

      OneTimeRun(Variables,Reactants,Products,Species);
```

```
   elseif selexn==2 %-----------------------------------------------------

      InputPower(Variables,Reactants,Products,Species);

   elseif selexn==3 %-----------------------------------------------------

      FlowPressure(Variables,Reactants,Products,Species);

   elseif selexn==4 %-----------------------------------------------------

      close all;                 % Close all open figures

   end              %-----------------------------------------------------
   status= fclose('all');        % Close all open files
end                              % Break free if QUIT is selected

close all;                          % Close all open figures
copyright;                          % Display copyright and contact info


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             Address questions/comments to:
%
%             Kedar Patel
%             Email: kedar@poincare.eecs.berkeley.edu
%
%             (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%             File:  Funk.m
%             Last Updated: 26 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function F = funk(iniX, Pabs, P, Qsccm, NF, Reactants, Products, Species)

global ee kB R_m L_m m_Kg T_Volts Gamma_Surface scat_Xsec chg_value
global i_K n_K K_A K_B K_C K_D
global Kij_A Kij_B Kij_C Eij_A Eij_B Total_Species Loss_Species
global Heavy_Reactions Pumping_React Source_React No_Electrons
global Ion_Surface Neutral_SurfacePumping Neutral_SurfaceSource
global Atomic_Composition
global Surface_Coeff

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Extract densities and electron temperature as separate vectors
n= iniX(1:(length(iniX)-1)).*NF;
Te= iniX(length(iniX));

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

ng= n(chg_value==0);      % Extract the neutral density
ni= n(chg_value>0);       % Extract the positive ion density
nneg= n(chg_value<0);     % Extract the negative ion density
ne= sum(ni)-sum(nneg);    % Extract the electron density

mg= m_Kg(chg_value==0);  % Extract the neutral mass
mi= m_Kg(chg_value>0);    % Extract the positive ion mass
me= 9.11E-31;             % Electron mass in Kg
```

```
Ti=T_Volts(chg_value>0);% Extract the positive ion temperature

% X-section for neutral-ion collisions
scatgi= scat_Xsec(chg_value==0, chg_value>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

R=R_m; L=L_m;
V =pi*R^2*L;                          % Reactor volume in cubic meters
A=2*pi*R*L;                           % Actual area
leff=((pi/L)^2+(2.405/R)^2)^(-0.5);  % Eff. diffusion length
QtorrLit= Qsccm/79.05;                % SCCM to Torr-Liter/sec
Qmolec= 4.483E17*Qsccm;               % SCCM to molecules/sec
kr= QtorrLit/(P*V*1000);              % Pumping rate coefficient

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

ve= (8*ee*Te/(pi*me))^0.5;            % Mean electron velocity
alpha= sum(nneg)/ne;                  % Measure of electronegativity
Gamma= sum((Te.*ni)./Ti)/sum(ni);     % Density weighted Te/Ti

% Bohm velocity of positive ions
Ubi=((ee*Te*(1+alpha))./(mi.*(1+alpha*Gamma))).^0.5;

lambda= 1./(ng*scatgi);               % Mean free path for each ion
lambdai= sum(ni.*lambda)./sum(ni);    % Density weighted ion mean free path

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

N= 20;        % Number of magnetic cusps
Bo= 2.5E3;    % Cusp field strength

rce= 3.37E-2*(Te^0.5)/Bo;
tmp_rci=1.44.*((Ti.*(mi./1.67e-27)).^0.5)./Bo;
rci= sum(ni.*tmp_rci)./sum(ni);
lambdae=ve/(sum(ng)*1e-13);
w=4*((rce*rci)^0.5)*(1 + R/(N*((lambdae*lambdai)^0.5)) );
floss=min([ N*w/(2*pi*R) 1]);

mag_confine=1;

if mag_confine
   % Axial and radial scaling factors
   hL= (0.86/(1+alpha))*( 3 + L/(2*lambdai) )^(-0.5);
   hRw= floss*(0.80/(1+alpha))*( 0.64 + floss*(3.36 + R/lambdai) )^(-0.5);
   hLw= floss*(0.86/(1+alpha))*( 0.74 + floss*(2.26 + L/(2*lambdai)) )^(-0.5);
   Aeff= pi*R*(R*hL + R*hLw + 2*L*hRw);
else
   % Axial and radial scaling factors
   hL= (0.86/(1+alpha))*( 3 + L/(2*lambdai) )^(-0.5);
   hR= (0.8/(1+alpha))*( 4 + (R/lambdai) )^(-0.5);
   Aeff= 2*pi*R*(R*hL + L*hR);
end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

phi= -Te*log(4*(ni*Ubi')/(ne*ve));   % Wall potential
Ei= phi + Te/2;                       % Ion energy lost to the wall
Ee= 2*Te;                             % Electron energy lost to the wall

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
kkp0= Reactants(:,No_Electrons);
kkp=zeros(size(kkp0));
kkp(~kkp0==0)=1;
kkp1=sum((ones(n_K,1)*T_Volts.*kkp)')./2;

kkp2= (Heavy_Reactions.*kkp1)';   % Kill reactions NOT involving Heavy particles
kkp3=kkp2.*(ee/(300*kB));         % Multiplier has the form (300/T);
kkp4= kkp3+(~kkp3>0);             % Prepare multiplier; Replace '0's with '1's
HeavyMultiplier=(1./kkp4).^K_D;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% The following statement calculates the bohm velocities of ALL selected
% species excluding electrons.

VirtualUbi=((ee*Te*(1+alpha))./(m_Kg.*(1+alpha*Gamma))).^0.5;
kkp5=sum((ones(n_K,1)*VirtualUbi.*kkp)');
kkp6= (Ion_Surface.*kkp5)';       % Kill reactions NOT involving ions
kkp7= kkp6.*(Aeff/V);
IonWallMultiplier= kkp7+(~kkp7>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Virtual mean neutral velocity
Virtualvo= ((8.*ee.*T_Volts)./(pi.*m_Kg)).^0.5;

kkp8=zeros(size(scat_Xsec));
kkp8(chg_value==0,chg_value==0)= ones(size(kkp8(chg_value==0,chg_value==0)));
kkp9= kkp8.*scat_Xsec;
kkp9(:,~chg_value==0)= ones(size(kkp9(:,~chg_value==0)));
scatg= kkp9;

% The scatg calculated above will have the correct cross-sections for the
% indices (charge=0,charge=0). It will have zeros in indices
% (~ charge=0, charge=0) so that the densities other than the neutrals
% don't contribute in calculation of mean free path.
% All the remaining places will have '1's.  These will get weeded out later.

VirtualDoo= (ee.*T_Volts)./(m_Kg.*Virtualvo.*(n*scatg));

% The line below replaces all '0's with a really small number in the wall
% recombination coefficient. It leaves the user entered values unchanged
% but forces all zeros to become the smallest number that the computer can
% represent i.e "realmin".  This has to be done because the computation of
% 'kkp10' involves division by 'GammaSurface'.

GammaSurface=Gamma_Surface;
GammaSurface(Gamma_Surface==0)=realmin;

kkp10=((leff^2)./VirtualDoo+((4*V)./GammaSurface-2*V)./(A.*Virtualvo)).^(-1);
kkp11=sum( (ones(n_K,1)*kkp10.*kkp)' );
kkp12= (Neutral_SurfacePumping.*kkp11)';
NeutralWallMultiplier=kkp12+(~kkp12>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

kkp13= (Pumping_React.*kr)';
PumpingMultiplier=kkp13+(~kkp13>0);

kkp14=(Source_React.*(Qmolec/V))';
SourceMultiplier=kkp14+(~kkp14>0);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%kkp15=VirtualUbi.*(Aeff/V);
```

```
%kkp15(~chg_value>0)=0;    % Rate of ion flow to wall

kkp16=kkp10;
kkp16(~chg_value==0)=0;    % Rate of neutral flow to wall

%kkp17=kkp15 + kkp16;          % Rates of ion and neutral flow to wall


% AtomicFlux is total no. of Boron and Flourine atoms per second per cu. m
% leaving as "surface sink"

AtomicFlux= (kkp16.*n)*Atomic_Composition';

%IonFlux= sum(Ubi.*(Aeff/V).*ni);
%thetaB= AtomicFlux(1)/(AtomicFlux(1) + IonFlux);
%kkp18= [AtomicFlux(2) IonFlux*thetaB IonFlux*thetaB IonFlux*thetaB
IonFlux*thetaB]';

kkp18= [AtomicFlux(2) AtomicFlux(1) AtomicFlux(1) AtomicFlux(1)
AtomicFlux(1)]';

%kkp18= [AtomicFlux(2) IonFlux*thetaB IonFlux*thetaB IonFlux*thetaB ...
%        IonFlux*thetaB]';

kkp19=(Surface_Coeff*kkp18)';
kkp20= kkp19(i_K);

SurfaceSourceMultiplier= (kkp20+(~kkp20>0))';

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

k1= SurfaceSourceMultiplier.*HeavyMultiplier.*IonWallMultiplier;
k2= NeutralWallMultiplier.*PumpingMultiplier.*SourceMultiplier;

RateConstants= K_A.*((Te).^K_B).*exp(-K_C./Te).*k1.*k2;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Kij= Kij_A.*(Te.^Kij_B).*exp(-Kij_C./Te);
Eij= Eij_A.*(Te.^Eij_B);

kkp21= ((ones(Total_Species,1)*(Kij'))').*Loss_Species;
kkp22= ((ones(Total_Species,1)*(Eij'))').*Loss_Species;
kkp23= sum(kkp21.*kkp22)';
Kij_Eij = kkp23(Species);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Density=[ne n];

% Find the term associated with each reaction:
tt=(ones(n_K,1)*Density.*Reactants)';
Terms=prod(tt+(~tt>0))'.*RateConstants;

% Assemble the RHS of each particle balance equation:
ff=(Products-Reactants)'*Terms;
F=ff(2:length(ff));     % RHS of all particle balance except electrons
F= F./NF;           % Normalize the F-vector

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Power Balance Equation: Volume and Surface electron energy loss
F(length(F)+1)=  Pabs - ee*ne*V*(n*Kij_Eij)  - ...
```

```
            ee*(Ubi*ni')*Aeff*(Ei+Ee) ;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           Address questions/comments to:
%
%           Kedar Patel
%           Email: kedar@poincare.eecs.berkeley.edu
%
%           (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%           File:  PlotPower.m
%           Last Updated: 26 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%

function ans = PlotPower(iXi)

    figure(1)
    semilogy(iXi(:,1), iXi(:,4))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Electron Density n_e [m^{-3}]')

    figure(2)
    loglog(iXi(:,1), iXi(:,5))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Total Neutral Density n_g [m^{-3}]')

    figure(3)
    loglog(iXi(:,1), iXi(:,6))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Total Positive Ion Density n_+ [m^{-3}]')

    figure(4)
    loglog(iXi(:,1), iXi(:,7))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('Fractional Ionization {\chi}_{iz}')

    figure(5)
    loglog(iXi(:,1), iXi(:,8))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{BF_3} [m^{-3}]')

    figure(6)
    loglog(iXi(:,1), iXi(:,9))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{BF_2} [m^{-3}]')

    figure(7)
    loglog(iXi(:,1), iXi(:,10))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{BF} [m^{-3}]')

    figure(8)
    loglog(iXi(:,1), iXi(:,11))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('n_{B} [m^{-3}]')

    figure(9)
```

```
loglog(iXi(:,1), iXi(:,12))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{F} [m^{-3}]')

figure(10)
loglog(iXi(:,1), iXi(:,13))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{(BF_3)^+} [m^{-3}]')

figure(11)
loglog(iXi(:,1), iXi(:,14))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{(BF_2)^+} [m^{-3}]')

figure(12)
loglog(iXi(:,1), iXi(:,15))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{(BF)^+} [m^{-3}]')

figure(13)
loglog(iXi(:,1), iXi(:,16))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{B^+} [m^{-3}]')

figure(14)
loglog(iXi(:,1), iXi(:,17))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{B^{++}} [m^{-3}]')

figure(15)
loglog(iXi(:,1), iXi(:,18))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('n_{F^+} [m^{-3}]')

figure(16)
loglog(iXi(:,1), iXi(:,19))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('Electron Temperature T_e [eV]')

figure(17)
loglog(iXi(:,1), iXi(:,20))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('BF_3 Fraction (%)')

figure(18)
loglog(iXi(:,1), iXi(:,21))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('BF_2 Fraction (%)')

figure(19)
loglog(iXi(:,1), iXi(:,22))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('BF Fraction (%)')

figure(20)
loglog(iXi(:,1), iXi(:,23))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('B Fraction (%)')

figure(21)
loglog(iXi(:,1), iXi(:,24))
xlabel('Power Absorbed P_{abs} [Watts]')
ylabel('F Fraction (%)')
```

```
    figure(22)
    loglog(iXi(:,1), iXi(:,25))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('{BF_3}^+ Fraction (%)')

    figure(23)
    loglog(iXi(:,1), iXi(:,26))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('{BF_2}^+ Fraction (%)')

    figure(24)
    loglog(iXi(:,1), iXi(:,27))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('{BF}^+ Fraction (%)')

    figure(25)
    loglog(iXi(:,1), iXi(:,28))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('{B}^+ Fraction (%)')

    figure(26)
    loglog(iXi(:,1), iXi(:,29))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('B^{++} Fraction (%)')

    figure(27)
    loglog(iXi(:,1), iXi(:,30))
    xlabel('Power Absorbed P_{abs} [Watts]')
    ylabel('F^+ Fraction (%)')

%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            Address questions/comments to:
%
%            Kedar Patel
%            Email: kedar@poincare.eecs.berkeley.edu
%
%            (c) University of California 1997, 1998');
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
%
%            File:  PlotFlowPressure.m
%            Last Updated: 26 April 1998
%
%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%~%
function ans = PlotFlowPressure(iXi)

plot_sel=PlotMenu;
if plot_sel==1

     figure(1)
     loglog(iXi(:,3), iXi(:,4))
     xlabel('Flow Rate Q [SCCM]')
     ylabel('Electron Density n_e [m^{-3}]')

     figure(2)
     loglog(iXi(:,3), iXi(:,5))
     xlabel('Flow Rate Q [SCCM]')
     ylabel('Total Neutral Density n_g [m^{-3}]')

     figure(3)
     loglog(iXi(:,3), iXi(:,6))
```

```
xlabel('Flow Rate Q [SCCM]')
ylabel('Total Positive Ion Density n_+ [m^{-3}]')

figure(4)
loglog(iXi(:,3), iXi(:,7))
xlabel('Flow Rate Q [SCCM]')
ylabel('Fractional Ionization {\chi}_{iz}')

figure(5)
loglog(iXi(:,3), iXi(:,8))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{BF_3} [m^{-3}]')

figure(6)
loglog(iXi(:,3), iXi(:,9))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{BF_2} [m^{-3}]')

figure(7)
loglog(iXi(:,3), iXi(:,10))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{BF} [m^{-3}]')

figure(8)
loglog(iXi(:,3), iXi(:,11))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{B} [m^{-3}]')

figure(9)
loglog(iXi(:,3), iXi(:,12))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{F} [m^{-3}]')

figure(10)
loglog(iXi(:,3), iXi(:,13))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{{BF_3}^+} [m^{-3}]')

figure(11)
loglog(iXi(:,3), iXi(:,14))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{{BF_2}^+} [m^{-3}]')

figure(12)
loglog(iXi(:,3), iXi(:,15))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{{BF}^+} [m^{-3}]')

figure(13)
loglog(iXi(:,3), iXi(:,16))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{B^+} [m^{-3}]')

figure(14)
loglog(iXi(:,3), iXi(:,17))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{B^{++}} [m^{-3}]')

figure(15)
loglog(iXi(:,3), iXi(:,18))
xlabel('Flow Rate Q [SCCM]')
ylabel('n_{F^+} [m^{-3}]')
```

```
figure(16)
loglog(iXi(:,3), iXi(:,19))
xlabel('Flow Rate Q [SCCM]')
ylabel('Electron Temperature T_e [eV]')

figure(17)
plot(iXi(:,3), iXi(:,20))
xlabel('Flow Rate Q [SCCM]')
ylabel('BF_3 Fraction (%)')
set(gca, 'YLim', [0 100])

figure(18)
plot(iXi(:,3), iXi(:,21))
xlabel('Flow Rate Q [SCCM]')
ylabel('BF_2 Fraction (%)')
set(gca, 'YLim', [0 100])

figure(19)
plot(iXi(:,3), iXi(:,22))
xlabel('Flow Rate Q [SCCM]')
ylabel('BF Fraction (%)')
set(gca, 'YLim', [0 100])

figure(20)
plot(iXi(:,3), iXi(:,23))
xlabel('Flow Rate Q [SCCM]')
ylabel('B Fraction (%)')
set(gca, 'YLim', [0 100])

figure(21)
plot(iXi(:,3), iXi(:,24))
xlabel('Flow Rate Q [SCCM]')
ylabel('F Fraction (%)')
set(gca, 'YLim', [0 100])

figure(22)
plot(iXi(:,3), iXi(:,25))
xlabel('Flow Rate Q [SCCM]')
ylabel('{BF_3}^+ Fraction (%)')
set(gca, 'YLim', [0 100])

figure(23)
plot(iXi(:,3), iXi(:,26))
xlabel('Flow Rate Q [SCCM]')
ylabel('{BF_2}^+ Fraction (%)')
set(gca, 'YLim', [0 100])

figure(24)
plot(iXi(:,3), iXi(:,27))
xlabel('Flow Rate Q [SCCM]')
ylabel('{BF}^+ Fraction (%)')
set(gca, 'YLim', [0 100])

figure(25)
plot(iXi(:,3), iXi(:,28))
xlabel('Flow Rate Q [SCCM]')
ylabel('{B}^+ Fraction (%)')
set(gca, 'YLim', [0 100])

figure(26)
plot(iXi(:,3), iXi(:,29))
xlabel('Flow Rate Q [SCCM]')
ylabel('B^{++} Fraction (%)')
```

```
    set(gca, 'YLim', [0 100])

    figure(27)
    plot(iXi(:,3), iXi(:,30))
    xlabel('Flow Rate Q [SCCM]')
    ylabel('F^+ Fraction (%)')
    set(gca, 'YLim', [0 100])

elseif plot_sel==2


    figure(1)
    loglog(iXi(:,2), iXi(:,4))
    xlabel('Pressure P [Torr]')
    ylabel('Electron Density n_e [m^{-3}]')

    figure(2)
    loglog(iXi(:,2), iXi(:,5))
    xlabel('Pressure P [Torr]')
    ylabel('Total Neutral Density n_g [m^{-3}]')

    figure(3)
    loglog(iXi(:,2), iXi(:,6))
    xlabel('Pressure P [Torr]')
    ylabel('Total Positive Ion Density n_+ [m^{-3}]')

    figure(4)
    loglog(iXi(:,2), iXi(:,7))
    xlabel('Pressure P [Torr]')
    ylabel('Fractional Ionization {\chi}_{iz}')

    figure(5)
    loglog(iXi(:,2), iXi(:,8))
    xlabel('Pressure P [Torr]')
    ylabel('n_{BF_3} [m^{-3}]')

    figure(6)
    loglog(iXi(:,2), iXi(:,9))
    xlabel('Pressure P [Torr]')
    ylabel('n_{BF_2} [m^{-3}]')

    figure(7)
    loglog(iXi(:,2), iXi(:,10))
    xlabel('Pressure P [Torr]')
    ylabel('n_{BF} [m^{-3}]')

    figure(8)
    loglog(iXi(:,2), iXi(:,11))
    xlabel('Pressure P [Torr]')
    ylabel('n_{B} [m^{-3}]')

    figure(9)
    loglog(iXi(:,2), iXi(:,12))
    xlabel('Pressure P [Torr]')
    ylabel('n_{F} [m^{-3}]')

    figure(10)
    loglog(iXi(:,2), iXi(:,13))
    xlabel('Pressure P [Torr]')
    ylabel('n_{{BF_3}^+} [m^{-3}]')

    figure(11)
    loglog(iXi(:,2), iXi(:,14))
```

```
xlabel('Pressure P [Torr]')
ylabel('n_{(BF_2)^+} [m^{-3}]')

figure(12)
loglog(iXi(:,2), iXi(:,15))
xlabel('Pressure P [Torr]')
ylabel('n_{(BF)^+} [m^{-3}]')

figure(13)
loglog(iXi(:,2), iXi(:,16))
xlabel('Pressure P [Torr]')
ylabel('n_{B^+} [m^{-3}]')

figure(14)
loglog(iXi(:,2), iXi(:,17))
xlabel('Pressure P [Torr]')
ylabel('n_{B^{++}} [m^{-3}]')

figure(15)
loglog(iXi(:,2), iXi(:,18))
xlabel('Pressure P [Torr]')
ylabel('n_{F^+} [m^{-3}]')

figure(16)
loglog(iXi(:,2), iXi(:,19))
xlabel('Pressure P [Torr]')
ylabel('Electron Temperature T_e [eV]')

figure(17)
plot(iXi(:,2), iXi(:,20))
xlabel('Pressure P [Torr]')
ylabel('BF_3 Fraction (%)')
set(gca, 'YLim', [0 100])

figure(18)
plot(iXi(:,2), iXi(:,21))
xlabel('Pressure P [Torr]')
ylabel('BF_2 Fraction (%)')
set(gca, 'YLim', [0 100])

figure(19)
plot(iXi(:,2), iXi(:,22))
xlabel('Pressure P [Torr]')
ylabel('BF Fraction (%)')
set(gca, 'YLim', [0 100])

figure(20)
plot(iXi(:,2), iXi(:,23))
xlabel('Pressure P [Torr]')
ylabel('B Fraction (%)')
set(gca, 'YLim', [0 100])

figure(21)
plot(iXi(:,2), iXi(:,24))
xlabel('Pressure P [Torr]')
ylabel('F Fraction (%)')
set(gca, 'YLim', [0 100])

figure(22)
plot(iXi(:,2), iXi(:,25))
xlabel('Pressure P [Torr]')
ylabel('(BF_3)^+ Fraction (%)')
set(gca, 'YLim', [0 100])
```

```
    figure(23)
    plot(iXi(:,2), iXi(:,26))
    xlabel('Pressure P [Torr]')
    ylabel('{BF_2}^+ Fraction (%)')
    set(gca, 'YLim', [0 100])

    figure(24)
    plot(iXi(:,2), iXi(:,27))
    xlabel('Pressure P [Torr]')
    ylabel('{BF}^+ Fraction (%)')
    set(gca, 'YLim', [0 100])

    figure(25)
    plot(iXi(:,2), iXi(:,28))
    xlabel('Pressure P [Torr]')
    ylabel('{B}^+ Fraction (%)')
    set(gca, 'YLim', [0 100])

    figure(26)
    plot(iXi(:,2), iXi(:,29))
    xlabel('Pressure P [Torr]')
    ylabel('B^{++} Fraction (%)')
    set(gca, 'YLim', [0 100])

    figure(27)
    plot(iXi(:,2), iXi(:,30))
    xlabel('Pressure P [Torr]')
    ylabel('F^+ Fraction (%)')
    set(gca, 'YLim', [0 100])

end
```

Bibliography

[1] D. Graves, *IEEE Trans. Plasma Sci.* **22**, 31 (1994)

[2] *Plasma Processing of Materials, Scientific Opportunities and Technological Challenges,* National Research Council, Washington D.C., 1991

[3] M. A. Lieberman and R. A. Gottscho, in *Physics of Thin Films,* eds. M. Francombe and J. Vossen (Academic, New York, 1994), Vol. 18.

[4] C. Lee, D. B. Graves, M. A. Lieberman, and D. W. Hess, *J. Electrochemical Soc.* **141**, 1546 (1994)

[5] C. Lee and M. A. Lieberman, *J. Vac. Sci. Technol. A* **13**, 368 (1995)

[6] Y. T. Lee, M. A. Lieberman, A. J. Lichtenberg, F. Bose, H. Baltes, and R. Patrick, *J. Vac. Sci. Technol. A* **15**, 113 (1997)

[7] S. Ashida, C. Lee and M. A. Lieberman, *J. Vac. Sci. Technol. A* **13**, 2498 (1995)

[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, New York, 1986)

[9] I. A. Kossyi, A. Y. Kostinsky, A. A. Matveyev, and V. P. Silakov, *Plasma Sources Sci. Technol.* **1**, 207 (1992)

[10] M. A. Lieberman and A. J. Lichtenberg in *Principles of Plasma Discharges and Materials Processing* (Wiley Interscience, New York, 1994)

[11] B. Eliasson and U. Kogelschatz, *Basic Data for Modelling of Electrical Discharges in Gases: Oxygen,* KLR 86-11 C. (1986)

[12] J. P. Booth and N. Sadeghi, *J. Appl. Phys.* **70**, 611 (1991)

[13] J. C. Greaves and J. W. Linnett, *Trans. Faraday Soc.* **55,** 1355 (1959)

[14] I. G. Kouznetsov, A. J. Lichtenberg, and M. A. Lieberman, *Plasma Sources Sci. Technol.* **5,** 662 (1996)

[15] E. Stoffels, W. W. Stoffels, D. Vender, M. Kando, G. M. W. Kroesen, and F. J. de Hoog, *Phys. Rev. E* **51**, 2425 (1995)

[16] A. V. Phelps, *JILA Information Center Report*, No. 28 (1995)

[17] A. V. Phelps, *J. Res. Natl. Inst. Stand. Technol.*, **95**, 407 (1990)

[18] J. T. Gudmundsson and M. A. Lieberman, *Plasma Sources Sci. Technol.* **7,** 1 (1998)

[19] L. D. Tsendin, *Sov. Phys. Tech. Phys.* **34**, 11 (1989)

[20] A. J. Lichtenberg, V. Vahedi, M. A. Lieberman, and T. Rognlien, *J. Appl. Phys.* **75**, 2339 (1994)

[21] V. A. Godyak, *Soviet Radio Frequency Discharge Research* (Delphic Associates Inc., Falls Church, VA 1986)

[22] E. C. Jones, W. En, S. Ogawa, D. B. Fraser, and N. W. Cheung, *J. Vac. Sci. Techol. B* **12**, 956 (1994)

[23] R. J. Matyi, D. L. Chapek, D. P. Brunco, S. B. Felch, and B. S. Lee, *Surf. and Coatings Techol.* **93**, 247 (1997)

[24] S. Qin, Y. Zhou, C. Chan, J. Shao, S. Denholm, *IEEE Conf. Plasma Sci.*, San Diego, CA,

USA, 19-22 May 1997, 151 (1997)

[25] P. J. Chantry, *J. Appl. Phys.* **62**, 1141 (1987)

[26] H. Toyoda, A. Hanami, M. Yamage, and H. Sugai, *Jap. J. Appl. Phys.* **30**, 514 (1991)

[27] G. Kota, J. Coburn, and D. Graves, *J. Vac. Sci. Technol. A* **16**, 270 (1998)

[28] *CRC Handbook of Chemistry and Physics*, 65th ed., (CRC Press, Boca Raton, 1984)

[29] K. Lau and D. Hildenbrand, *J. Chem. Phys.* **72**, 4928 (1980)

[30] C. Lee, D. Graves, and M. Lieberman, *Plasma Chem. and Plasma Proc.* **16**, 99 (1996)

[31] K. N. Leung, T. K. Samec, and A. Lamm, *Phys. Lett. A* **51A**, 490 (1975)

[32] K. N. Leung, G. R. Taylor, J. M. Barrick, S. L. Paul, and R. E. Kribel, *Phys. Lett. A* **57A**, 145 (1976)

[33] N. Hershkowitz, K. N. Leung, and T. Romesser, *Phys. Rev. Lett.* **35**, 277 (1975)

[34] G. Matthieussent and J. Pelletier in *Microwave Excited Plasmas*, eds. M. Moissan and J. Pelletier, Elsevier, Amsterdam (1992)

[35] M. Farber and R. D. Srivastava, *J. Chem. Phys.* **81**, 241 (1984)

[36] M. Hayashi in a unpublished Tokyo University report (Courtesy of Mark Kushner).

[37] W. D. Robb and R. J. W. Henry, *Phys. Rev. A* **16**, 2491 (1977)

[38] E. J. Robinson and S. Geltman, *Phys. Rev.* **153**, 4 (1967)

[39] K. S. Baliyan and A. K. Bhatia, *Phys. Rev. A* **50**, 2981(1994)

[40] W. M. Huo and Y.-K. Kim, (to be published)

[41] K. L. Bell, H. B. Gilbody, J. G. Hughes, A. E. Kingston, and F. J. Smith, *J. Phys. Chem. Ref. Data* **12**, 891 (1993)

[42] T. R. Hayes, R. C. Wetzel, and R. S. Freund, *Phys. Rev. A* **35**, 578 (1987)