

5-2024

## Modeling the Neutral Densities of SPARC Using a Python Version of KN1D

Gwendolyn R. Galleher  
*William & Mary*

Follow this and additional works at: <https://scholarworks.wm.edu/honorsthesis>



Part of the [Dynamical Systems Commons](#), [Nuclear Commons](#), [Numerical Analysis and Scientific Computing Commons](#), and the [Plasma and Beam Physics Commons](#)

---

### Recommended Citation

Galleher, Gwendolyn R., "Modeling the Neutral Densities of SPARC Using a Python Version of KN1D" (2024). *Undergraduate Honors Theses*. William & Mary. Paper 2225.  
<https://scholarworks.wm.edu/honorsthesis/2225>

This Honors Thesis -- Open Access is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Undergraduate Honors Theses by an authorized administrator of W&M ScholarWorks. For more information, please contact [scholarworks@wm.edu](mailto:scholarworks@wm.edu).

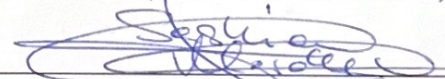
# Modeling the Neutral Densities of SPARC Using a Python Version of KN1D

A thesis submitted in partial fulfillment of the requirement  
for the degree of Bachelor of Science with Honors in  
Physics from the College of William and Mary in Virginia,

by

Gwendolyn R. Galleher

Accepted for Honors  
(Honors or no-Honors)



Saskia Mordijck, Advisor



Marc Sher, Physics



Leah Shaw, Mathematics

Williamsburg, Virginia  
May 3, 2024



## Abstract

Currently, neutral recycling is a crucial contributor to fueling the plasma within tokamaks. However, Commonwealth Fusion System's SPARC Tokamak is expected to be more opaque to neutrals. Thus, we anticipate that the role of neutral recycling in fueling will decrease. Since SPARC is predicted to have a groundbreaking fusion power gain ratio of  $Q \approx 10$ , we must have a concrete understanding of the opacity and whether or not alternative fueling practices must be included. To develop said understanding, we produced neutral density profiles via KN1DPy, a 1D kinetic neutral transport code for atomic and molecular hydrogen in an ionizing plasma. KN1DPy is a one-to-one recode of the IDL program KN1D, originally developed in the early 2000s. We translated KN1D from IDL to Python to increase accessibility and allow for better coupling with other modern programs. We verified KN1DPy using Alcator C-Mod data and comparing the Python and the IDL outputs. Using simulated profiles for electron density ( $n_e$ ), ion temperature ( $T_i$ ), and electron temperature ( $T_e$ ), we can run KN1DPy and produce simulated neutral density profiles for SPARC. These results can then be compared to the expectation from the empirical equation for opacity, which states that most neutrals will ionize before crossing the separatrix. Results consistent with these expectations will mean that neutral recycling plays a weaker role in fueling the plasma core.

# Contents

Abstract	5
Acknowledgments	ii
List of Figures	v
List of Tables	vi
<b>1 Introduction</b>	<b>1</b>
<b>2 Plasma Opacity and it's Affects on Neutral Recycling</b>	<b>6</b>
2.1 Tokamak Physics . . . . .	6
2.2 KN1D . . . . .	9
2.3 KN1DPy . . . . .	12
<b>3 SPARC Predictive Analysis</b>	<b>18</b>
<b>4 KN1DPy SPARC Profiles</b>	<b>22</b>
<b>5 Conclusion</b>	<b>28</b>
<b>A IDL to Python</b>	<b>30</b>
<b>B Code Sample for Synthetic Data</b>	<b>32</b>

# Acknowledgments

I want to start by thanking my parents, Ian and Diane, and my siblings Vivian and Declan. Thank you for being by my side since day one and supporting me even though we are separated by hundreds of miles. When it comes to my academic career, I am especially grateful for my dad. Thank you for patiently helping me through all of those long nights where I struggled through my elementary school math homework. You always believed in me even though I often didn't believe in myself. Furthermore, you were the only person who didn't trust me when my middle school self, confidently declared I hated physics, clearly I was wrong.

Additionally I couldn't have changed my mind regarding math and physics without support from my various teachers, professors and advisors over the years. I want to start by thanking Ms. Martinsek and Ms. Boweman, during my toughest days in college I often found myself thinking back to your classrooms. Thank you for always being such positive female role models and showing students the joys of STEM. Thank you Leah Shawe and David Armstrong for being my major advisors and some of my earliest professors at W&M. Moreover, I want to thank Marc Sher for always bringing a smile to my face and introducing me to Saskia and the wonders of fusion during my sophomore year.

Above all else I could not have achieved this without the guidance and support of Saskia Mordijck. Thank you for being my north star over the past two years, both academically and personally. You have shown me that I don't need to change myself

to fit the definition of a physicist, but rather I can change the definition of a physicist to fit me.

Lastly I am grateful for all of my friends, who have supported me through this journey. Many of whom have never taken a physics class, but have always listened and supported my ambitions. I want to give special thanks to Tyler Yeatts who has listened to me discuss my research more than any Business major reasonably should. Thank you.

# List of Figures

1.1	A graph of the average binding energy per nucleon vs the mass number for each element [5]. . . . .	2
1.2	A plot comparing numerous fusion reactors and their associated fusion power gain ratios in relation to their triple product parameters. [10] .	4
2.1	Poloidal cross section of SPARC. The plasma separatrix is red, the central solenoid and poloidal field coils are blue, and the divertor and first limiting surfaces are in black. [9] . . . . .	7
2.2	Previous KN1D simulations of the edge neutral transport of DII-D (left) and Alcator C-MOD (right). The graphs from top to bottom: a) computed ion density, b) resulting neutral density, c) local gradient scale lengths and d) profile shapes for ionization [20]. . . . .	10
2.3	Graph of an interpolation used in the subprogram <b>create_kinetic_h_mesh</b> . 14	
2.4	A comparison of IDL and Python neutral density profiles using Alcator C-MOD Data. The IDL results are on the left and the python results on the right, with atomic and molecular hydrogen on top and bottom respectively. . . . .	15
3.1	Simulated profiles for electron temperature (left), ion temperature (center) and electron density (right) [24]. . . . .	18
3.2	A synthetic ion temperature profile (left) and ion density profile (right). 20	



4.1	The neutral density profiles of molecular (left) and atomic (right) hydrogen based on the CMOD data sets. The profiles are labeled based on how much the initial electron density was multiplied by. Meaning the lowest density input, was the initial data multiplied by 1, and the highest was multiplied by 5. . . . .	23
4.2	The neutral density profiles of molecular (left) and atomic (right) hydrogen based on the SPARC synthetic data sets with edge temperatures ranging from 2 keV to 10 keV and an edge density of $2 \times 10^{20}$ ( $\text{m}^{-3}$ ). The profiles are labeled based on how much the initial input temperature was scaled by. Meaning the lowest temperature input was 2 keV scaled by 1, and the highest was 2 keV scaled by 5 i.e. 10 keV.	24
4.3	A comparison of the atomic hydrogen neutral density profiles with charge exchange and without. . . . .	25
4.4	The neutral density profiles of molecular (left) and atomic (right) hydrogen based on the SPARC synthetic data sets with edge electron densities ranging from $0.6 \times 10^{20}$ to $3 \times 10^{20} \text{ m}^{-3}$ and an edge electron temperature of 7 keV. The profiles are labeled based on how much the initial input density was scaled by. Meaning the lowest temperature input was $0.6 \times 10^{20}$ scaled by 1, and the highest was $0.6 \times 10^{20}$ scaled by 5, i.e. $3 \times 10^{20}$ scaled by 5. . . . .	26
4.5	A comparison of the atomic hydrogen neutral density profiles for an edge electron density of $3 \times 10^{20} \text{ m}^{-3}$ with and without charge exchange.	27

# List of Tables

A.1	A table demonstrating some translations between IDL and Python, where $m$ and $n$ are the dimensions of the chosen array. Furthermore, $A$ is the array of interest and $a$ is simply a specified integer. . . . .	30
-----	--	----

# Chapter 1

## Introduction

As the world population continues to increase, an abundance of global challenges will emerge, requiring a multitude of advancements. For context, in the next 50 years the human population is projected to reach a peak of 9.73 billion people [1]. The growing population paired with an unprecedented reliance on technology worldwide, will certainly place strain on the global energy grid. From 1971 to 2019 there was already a 260% increase in energy consumption [2]. To sustain this level of growth and avoid potential energy shortages, an expansion of the current infrastructure is necessary.

An increase in energy usage is not the sole issue calling for an upheaval of the energy grid; climate concerns are also intensifying. CO<sub>2</sub> emissions have caused the Earth's temperature to increase by 2°F since 1880, and fossil fuels are the leading contributor. While fossil fuels comprise 6.6% less of the energy supply, we are burning more fossil fuels today than in 1971 [2]. If consumption and supply patterns continue to follow this trend, catastrophic climate effects will occur. To combat the associated problems with both increasing energy demand and climate change, there needs to be a significant increase in clean energy sources.

Taking the above concerns into consideration, nuclear fusion energy has the potential to be a sustainable substitute for fossil fuels' role in the energy supply chain.

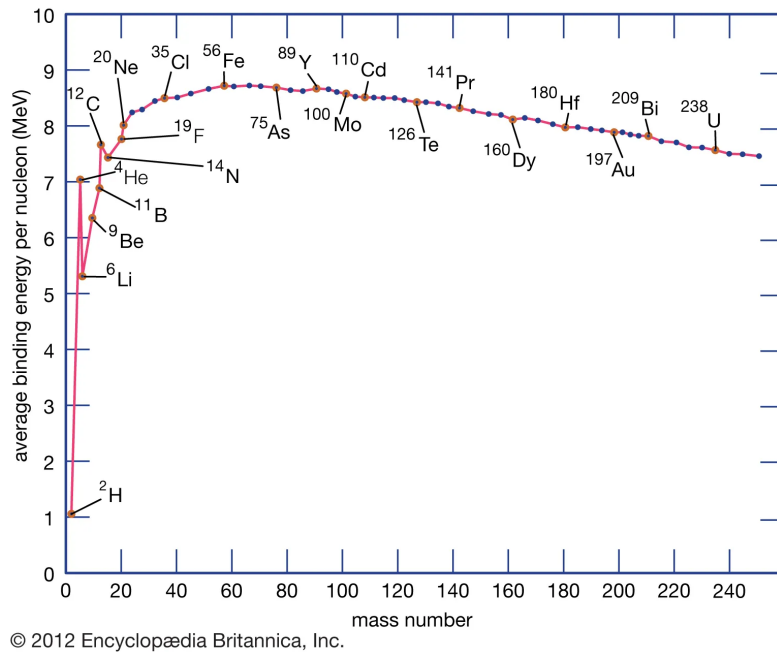


Figure 1.1: A graph of the average binding energy per nucleon vs the mass number for each element [5].

Nuclear fusion is the process that fuels stars, where two atoms collide, causing their nuclei to combine and form a single, heavier nucleus [4]. Heavier elements have a higher binding energy, which is the energy required to split an atom's nucleus into only protons and neutrons [5]. The heavier nucleus produced by a fusion reaction requires less energy to exist. Due to conservation of energy, the excess energy is then released. The binding energy curve is shown in Fig.1.1. Elements to the right of  $^{56}\text{Fe}$  release energy during nuclear fusion. In contrast, elements to the right of  $^{56}\text{Fe}$  release energy when their nuclei split into two. The splitting of an atom's nucleus is called nuclear fission, and it is the reaction that powers current-day nuclear power plants.

One of the most promising combinations of elements for fusion energy is deuterium and tritium, two different hydrogen isotopes. A single fusion reaction between these two atoms produces approximately 17.59 MeV of energy. Nonetheless, to reach a

fusion reaction, there still must be a significant input of energy as well, and generally, temperatures in the order of  $10^8$  K are required [4]. At this temperature, any substance will be a plasma, which is colloquially referred to as the fourth state of matter. A plasma is an ionized gas, meaning its atoms have had at least one electron stripped away, leaving a positively charged ion and a negatively charged free electron. Furthermore, for an ionized gas to be considered a plasma, it must meet several other specifications, including quasi-neutrality and collective behavior [6].

Atoms colliding within a plasma have a higher chance of scattering than going through a fusion reaction. Therefore, it is crucial to confine the plasma to increase the likelihood of successful reactions, allowing fusion to occur at a sufficient rate [4]. Space plasmas, which comprise stars, are confined by gravitational fields. However, this method is unattainable on Earth. That being said, we can achieve similar confinement using magnetic fields. Magnetic fields exert a force called the Lorentz Force on moving charged particles. The Lorentz Force inhibits charged particles from ever crossing a magnetic field line as shown in Eq.1.1 [7].

$$\mathbf{F} = q[\mathbf{E} + \mathbf{v} \times \mathbf{B}] \tag{1.1}$$

Where  $\mathbf{F}$  represents the force,  $q$  represents the charge,  $\mathbf{E}$  is the electric field,  $\mathbf{v}$  is the particle velocity, and  $\mathbf{B}$  is the magnetic field. Since plasma is composed of charged particles, closed magnetic fields will confine the plasma to the region within them. This concept is the foundation for several current nuclear fusion reactor designs, two of which are Tokamaks and Stellarators. For the purposes of this research, we will focus on the tokamak, a toroidal machine with magnetic coils surrounding it. These coils create toroidal and poloidal fields to confine the plasma.

One tokamak in particular is the framework for the following thesis — Commonwealth Fusion Systems' flagship tokamak, SPARC. Commonwealth Fusion Systems is

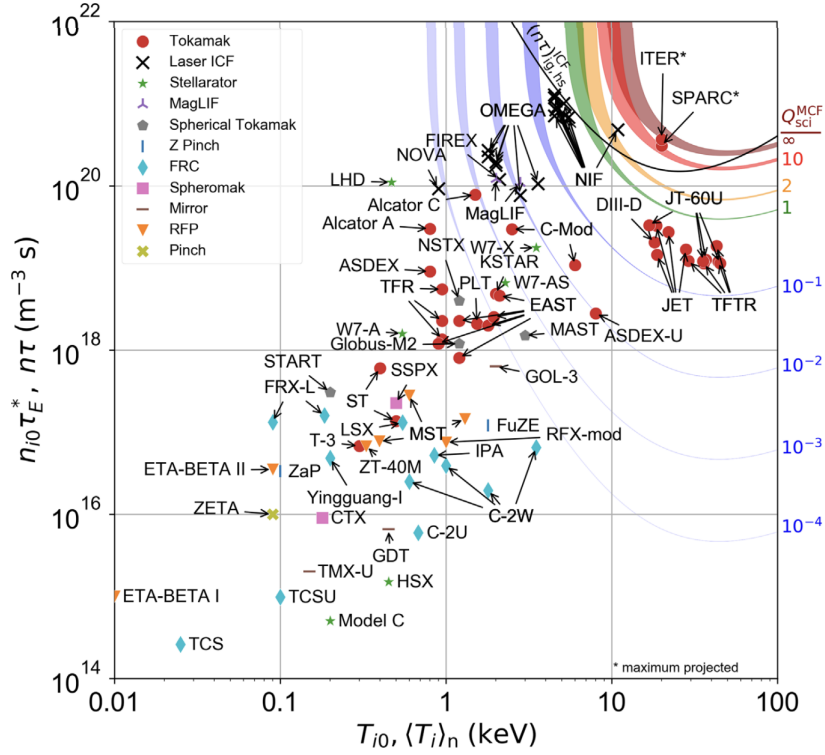


Figure 1.2: A plot comparing numerous fusion reactors and their associated fusion power gain ratios in relation to their triple product parameters. [10]

one of the several fusion startup companies taking advantage of the gap in sustainable energy sources. They are headquartered in Devens, MA, where SPARC is currently under construction and is expected to be operational in 2025 [8]. Once complete, SPARC is expected to reach a fusion power gain of  $Q \approx 10$ , where  $Q$  is the ratio of energy output to energy input [9]. Reaching  $Q = 1$  means a machine has reached an energy break-even point, and SPARC's expectation would be a breakthrough in magnetic confinement fusion energy. In Fig.1.2, we can see SPARC's fusion power gain compared to many other fusion systems, demonstrating the expected operational regime. Preparing adequate predictive analysis is imperative as the first run of SPARC approaches.

The primary objective of this thesis is to conduct predictive analysis regarding

the opacity of SPARC. More specifically, we aim to determine if neutral particles are predominantly ionizing outside of the separatrix. This is achieved using neutral density profiles produced via synthetic SPARC data as inputs to KN1DPy, a 1D kinetic neutral transport code. If the profiles are largely centralized outside of the separatrix, neutral recycling will play a weaker role in fueling thus CFS must rely on alternative fueling practices to power SPARC. Prior to accomplishing these results, KN1DPy was developed by translating the original version from IDL into Python. This translation allowed for increased access and ease of use. Furthermore, the structure and development of KN1D form a substantial section of the following work.

The current chapter aims to lay the groundwork necessary to follow the subsequent thesis, organized as follows. Chapter 2 will offer the theory behind neutral particles and how they fuel the plasma core. This then leads into a discussion of opacity and the associated neutral reactions which can affect said opacity. The second half of the chapter then gives an in-depth overview of KN1DPy, including its structure and the process of translating KN1D from IDL into Python. Chapter 3 will then explain the synthetic data used as inputs for KN1DPy and the various parameters we chose to change, allowing us to analyze the opacity. This is followed by Chapter 4, which aims to discuss the simulated neutral density profiles and their implications for the anticipated opacity of SPARC. Lastly, in Chapter 5, we will summarize the results and their implications, along with an outline of how future work could follow.

# Chapter 2

## Plasma Opacity and its Affects on Neutral Recycling

### 2.1 Tokamak Physics

In current-day tokamaks, neutral particles play a significant role in ensuring the efficiency of the machines [11]. Specifically, neutrals go through a cyclical process known as neutral recycling, where plasma shifts away from the core and recombines at the edge. This is due to a drop in temperature at the edge of the confined region because of heat exhaust. The recombination results in “cold” neutrals, which are approximately 3 eV and free to move through the tokamak regardless of magnetic flux lines. As these neutrals move through the tokamak, they collide with the plasma and other neutrals, resulting in their ionization. This cycle allows for consistent plasma density in the core and, thus, consistent fusion reactions.

For neutral recycling to contribute to the plasma density in the core, neutral particles must ionize inside the separatrix. The separatrix is the last closed magnetic flux surface displayed in red in Fig.2.1. Therefore, magnetic fields confine all charged particles inside the separatrix, while those outside are unconfined. This asserts that neutrals ionizing outside the separatrix can not contribute to the plasma core and will divert toward the wall instead. The region outside of the separatrix is described



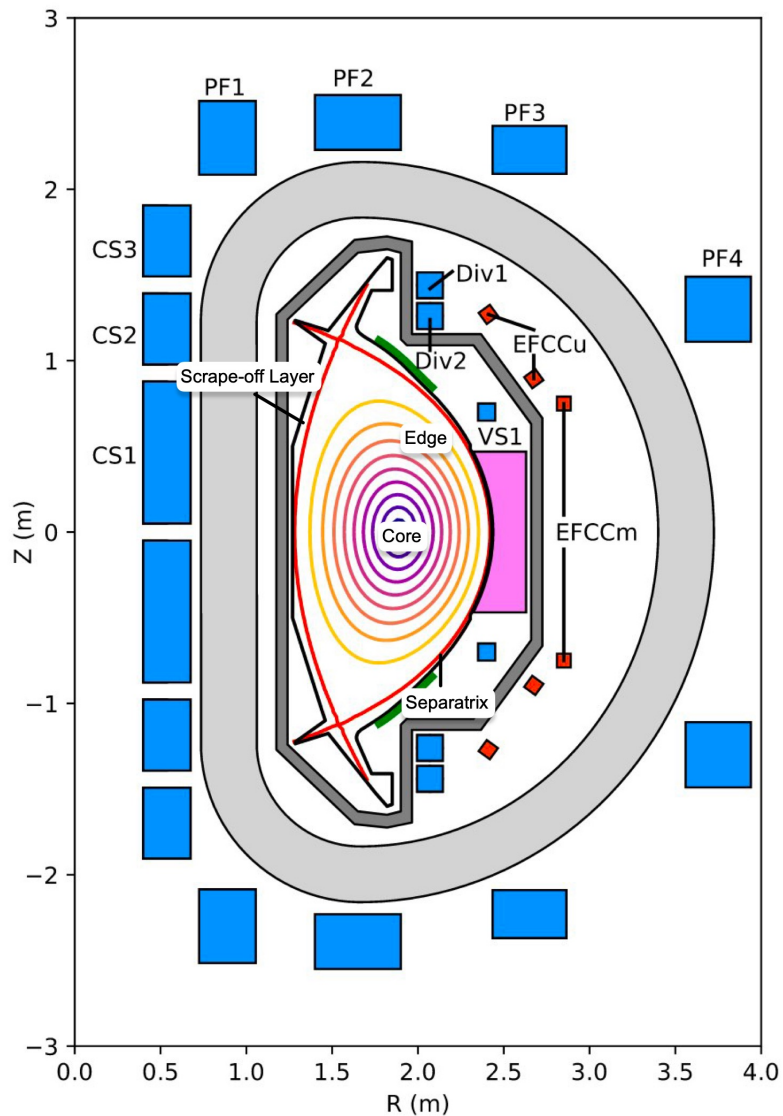


Figure 2.1: Poloidal cross section of SPARC. The plasma separatrix is red, the central solenoid and poloidal field coils are blue, and the divertor and first limiting surfaces are in black. [9]

as the scrape-off layer because charged particles in this region are not confined and are left to get “scraped off” towards the divertor. The ability for neutrals to cross the magnetic field lines prior to ionization is described as the opacity of the plasma. The concept of opacity is derived from optics, where it refers to the extent to which electromagnetic radiation photons cannot pass through a material [12]. In the context

of plasma physics, a similar concept is used where the substance is plasma, and instead of photons, neutral particles penetrate. In plasma physics, opacity depends on the plasma temperature and electron density, and the following empirical equation can characterize it,

$$\eta = \sigma n_e = \frac{1}{\lambda_d} \quad (2.1)$$

Here,  $\sigma$  represents the ionization cross section,  $n_e$  is the electron density, and  $\lambda$  is the neutral penetration depth [13]. Since ionization cross section depends on temperature, Eq.2.1 demonstrates that as temperature and electron density increase in the plasma, opacity also increases. As a result neutrals in tokamaks with higher electron densities and temperatures will have a shorter penetration depth. If the neutral penetration is reduced, then neutral recycling will have a weaker impact on fueling the plasma core.

Another factor that impacts neutral penetration depth is the collisional reaction called charge exchange. Charge exchange occurs when a neutral atom and an ion collide, and an electron is exchanged between the two atoms as expressed below [14].



While the net charge of this reaction stays constant, the energy of the final neutral dramatically increases. Charge exchange reactions can increase a “cold neutral” from 3 eV to often 100 eV or higher. As one can imagine, neutrals at this energy can penetrate further into the plasma core than the “cold neutrals” passively moving inwards. Therefore, charge exchange can affect the opacity in ways Eq.2.1 does not consider.

As previously mentioned, the next generation of tokamaks, including SPARC, will have higher power gain, meaning higher energy output. To achieve higher energy output, these systems maximize the fusion triple product,

$$n \times T \times \tau_E \quad (2.3)$$

where  $n$  is the plasma density,  $T$  is the plasma temperature, and  $\tau_E$  is the energy confinement time [15]. Increasing any of these three values increases the likelihood of fusion reactions, thus increasing energy output. This relationship is further outlined in Fig.1.2, which plots the fusion power gain in relation to the triple product parameters. While some machines, such as ITER, are increasing confinement time by making a larger tokamak ( $R_0 = 6.02\text{m}$ ), SPARC is taking a different approach [16].

The CFS team uses REBCO high-temperature superconductors to create a high magnetic field ( $B_0 = 12.2$  T). The high magnetic field allows for a more compact design with major radius  $R_0 = 1.85$  m and minor radius  $a = 0.57$  m [9]. As a result, SPARC will also have higher temperature and electron density, thus making it more opaque to neutrals [11]. As explained above, more opaque plasma results in neutrals ionizing outside of the separatrix; thus, neutral recycling will have a smaller effect on fueling. We can assess the opacity of SPARC using neutral density profiles. The density profile gives insight into where neutrals are concentrated, and low density inside the separatrix would mean low penetration depth. Resulting in a decrease in neutral recycling's contribution to fueling. To provide said neutral density profiles for SPARC, we will use the Python version of KN1D.

## 2.2 KN1D

KN1D, *Kinetic Neutral 1D Transport Code*, is a 1-dimensional spatial, 2-dimensional velocity kinetic neutral recycling algorithm for atomic and molecular hydrogen in an ionizing plasma [17]. B. Labombard developed the program in the early 2000s, using IDL, *Interactive Data Language*. Since its release, KN1D has played a strong role in tokamak research. One example of said use, among many, can be seen in Fig.2.2. KN1D solves for the molecular and atomic hydrogen distribution functions,  $f_{H_2}$  and  $f_n$ . These functions describe the probability of various positions and velocities for

H and H<sub>2</sub> within the plasma. Since KN1D only considers the x spatial dimension, it is assumed that  $f_{H_2}$  and  $f_n$  are rotationally symmetrical. Therefore, they can be described in terms of the velocity in the x direction across the minor radius from the wall to core,  $v_x$ , and the radial velocity,  $v_r$ . Using the distribution functions, KN1D then produces temperature and density profiles for neutral particles and ions, along with the molecular hydrogen dissociation rate profiles. The numerical process

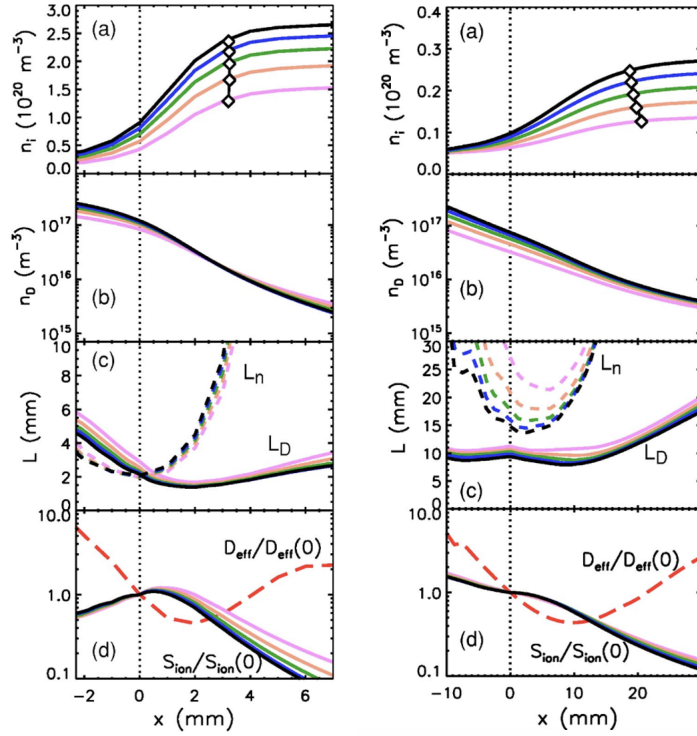


Figure 2.2: Previous KN1D simulations of the edge neutral transport of DII-D (left) and Alcator C-MOD (right). The graphs from top to bottom: a) computed ion density, b) resulting neutral density, c) local gradient scale lengths and d) profile shapes for ionization [20].

begins with the overhead program **KN1D**, which requires inputs for 1-D geometric dimensions (limiter, scrape-off layer, core), plasma profiles (density, ion and electron temperature), and the molecular neutral pressure at the wall. Synthetic examples of the plasma profiles are shown in Fig.3.2. This top-level program also sets up the

numerical grid and the boundary conditions. Furthermore, **KN1D** iterates through the two main subprograms **Kinetic\_H2** and **Kinetic\_H**, which solve for  $f_{H_2}$  and  $f_n$  respectively.

To find the two distribution functions each program applies Boltzmann equation, a partial differential equation commonly used to model statistical systems that are not in equilibrium [18].

$$v_x \frac{\partial f}{\partial x} = \left[ \frac{\partial f}{\partial t} \right]_c + S \quad (2.4)$$

Where  $(\partial f / \partial t)_c$  accounts for the various collisions each species can experience, and  $S$  is the source term. For molecular hydrogen, the source term comprises the source distribution function from the wall, 6 electron impact ionization reactions, and the molecular hydrogen dissociation rate. The wall source distribution function is assumed to be Maxwellian, and the reaction rates are calculated via Sawada [19]. As for the Boltzmann equation involving  $f_n$ , the source term is still comprised of the molecular hydrogen dissociation rate but is not directly affected by the wall source distribution function. Moreover, it uses 10 electron impact ionization reactions rather than 6. The collisional term for both species consists of the collisional charge exchange operator and three elastic collisions.

The two subprograms then solve the Boltzmann equation via the method of successive generations [21]. This method begins by breaking the complete Boltzmann Equation into more minor constituents that correlate to each population of neutrals. The complete equation can then be represented in the following form,

$$\sum_{j=1}^N v_x \frac{\partial f}{\partial x} = s_{wallj-1} + \beta_{cxj-1} - \alpha_c f_{j-1} + \omega_{Mj-1} \quad (2.5)$$

Here,  $s_{wall}$  correlates to the side wall source distribution,  $\beta_{cx}$  is the charge exchange source,  $\alpha_c$  is the charge exchange collisional frequency, and  $\omega_M$  is the elastic momentum transfer frequency. These terms are calculated via direct integration, except

that of  $\omega_M$ . The integration to calculate the momentum transfer frequency would take more computational power than the rest of KN1D combined. As a result, KN1D uses the BGK model to approximate the frequency [22].

The program then solves for the individual distribution functions via integration. The numerical procedure begins with “0th generation”, which refers to the Boltzmann equation for the first population of neutrals. This population of neutrals correlates to the portion of the total distribution function unaffected by ionization, charge exchange, or elastic scattering. Therefore, the Boltzmann equation for the “0th generation” is as follows,

$$v_x \frac{\partial f_0}{\partial x} = S_{H_2} \hat{f}_w - \alpha_c f_0. \quad (2.6)$$

Where  $\hat{f}_w$  is a normalized Maxwellian distribution of neutral particles at the wall temperature. From here, **Kinetic\_H** and **Kinetic\_H2** iterate through the subsequent generations, which include the ionization, charge exchange, and elastic scattering terms. As they iterate through each generation, the various portions of the distribution function along the numerical grid are calculated. Once every generation of the distribution function is gathered, their sum equals the total distribution function.

## 2.3 KN1DPy

While more accurate and advanced kinetic neutral models, such as the Monte Carlo Transport code EIRENE, exist, KN1D’s 1-dimensional nature offers distinct benefits [23]. Due to the simplicity of the 1-dimensional Boltzmann equation, KN1D requires less computational power and has a shorter runtime compared to other codes. These qualities make KN1D more efficient for predictive analysis. With shorter runtimes, users can adjust the input parameters and receive almost immediate feedback on how those parameters affect the system.

Furthermore, in the context of opacity, KN1D provides deeper insight than the empirical equation (Eq. 2.1) because it accounts for charge exchange. As previously mentioned, charge exchange results in high-energy neutrals that can penetrate further into the plasma compared to their “cold” counterparts. Thus, KN1D can offer a more comprehensive understanding while still minimizing computational time and power.

However, a significant drawback of KN1D is that it was written in IDL, an array-based language which has declined in use since its development in the 1970’s. Not only is IDL difficult to couple with current major languages and programs, but it is also not an open-source language. In response to the difficulties of IDL, we developed a Python version of KN1D, titled KN1DPy, to produce simulated neutral density profiles for SPARC. This Python version is a one-to-one recode of the IDL version, thus keeping the same structure. Nonetheless, there is potential for updated atomic physics later down the line.

As previously stated, IDL is an array-based language, whereas Python is an object-oriented language. While the most simple functions are shared between the two, their core functionality varies. To circumvent some of the difficulties caused by these differences, KN1DPy employs several Python libraries. The main library used to carry out array operations is NumPy, an open-source scientific computing package. While NumPy recreates the advanced array manipulation that is standard in IDL, the two languages use opposite indexing methods. NumPy uses row-column indexing, whereas IDL uses column-row indexing. Since array operations power a significant portion of KN1D, this was a crucial distinction when writing KN1DPy. Furthermore, KN1DPy uses SciPy, another open-source package for mathematics and science. SciPy facilitates the series of interpolations required for the computational method. Appendix B provides a more in-depth look at the common translations.

While understanding the conceptual differences between the two languages is cru-

cial, application is often not as simple. Consequently, KN1DPy underwent significant debugging throughout its development. The primary debugging technique was a top-down approach. This involved running the overhead **KN1D** program until an error occurred. In the event of an error, we followed the Python traceback to the file that required debugging. This approach bypassed running each file to check for errors, a practical reality since KN1D consists of over 40 separate files.

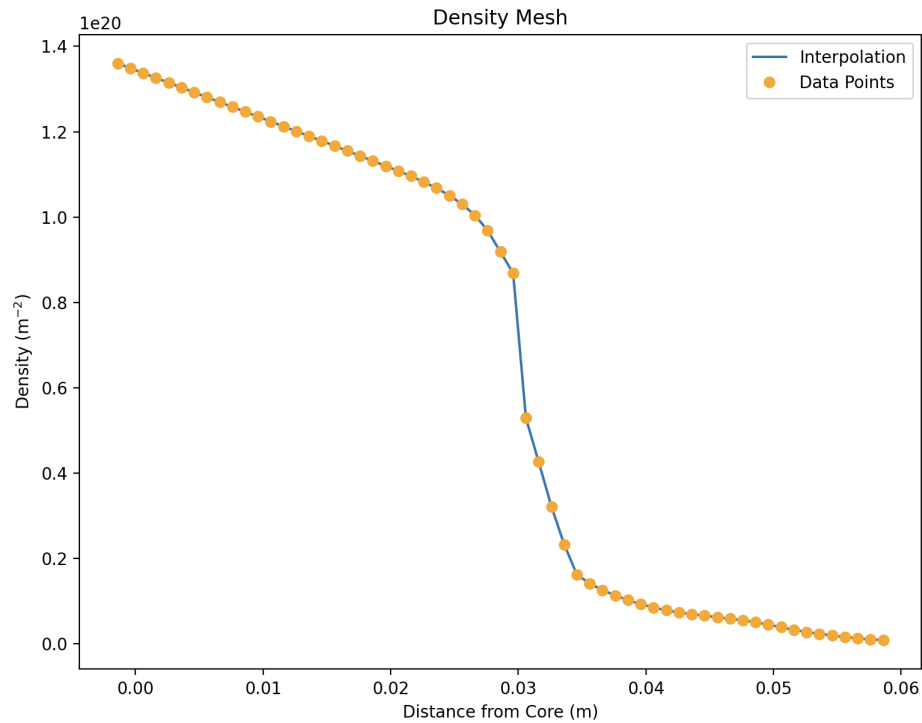


Figure 2.3: Graph of an interpolation used in the subprogram **create\_kinetic\_h\_mesh**.

Suppressing compilation and runtime errors is only a fraction of the required steps to ensure a working program; verifying the accuracy of the physics is also vital. One can verify a portion of the functionality by graphing the interpolation of internal files. Interpolation occurs frequently in the numerical process; however, for simplicity, the following discussion will focus on interpolation within **create\_kinetic\_h\_mesh**. This



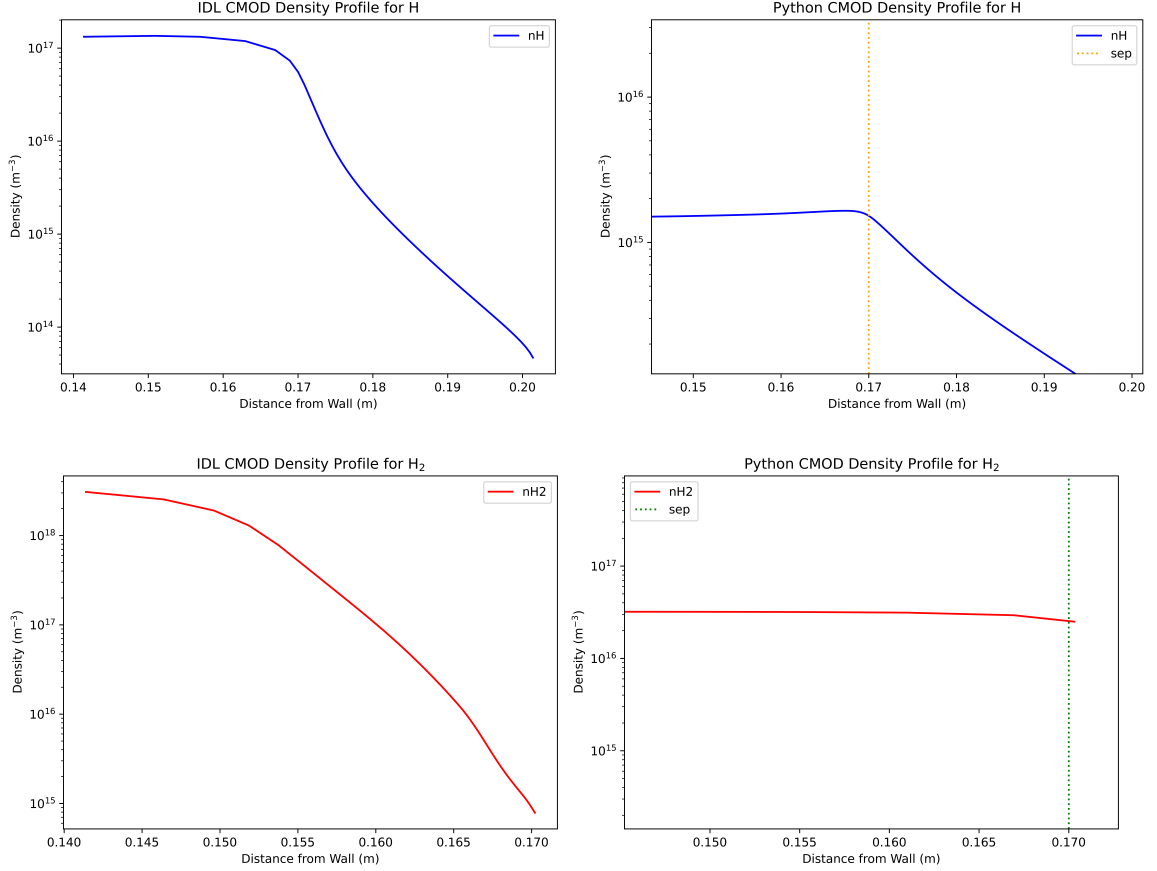


Figure 2.4: A comparison of IDL and Python neutral density profiles using Alcator C-MOD Data. The IDL results are on the left and the python results on the right, with atomic and molecular hydrogen on top and bottom respectively.

function is the second subprogram called within **KN1D**, and it defines the velocity and positional meshes for atomic hydrogen, which are later used in the method of successive generations. To calculate said meshes, **create\_kinetic\_h\_mesh** interpolates over the plasma profiles given as inputs. As seen in Fig. 2.3, the interpolated mesh closely aligns with the data inputs. The agreement between the data and the interpolation further verifies the accuracy of KN1DPy.

We commonly verified KN1DPy via direct comparison to the IDL outputs since the original KN1D has already been extensively verified and validated. We used data from the retired Alcator C-Mod Tokamak as inputs for both KN1D and KN1DPy

to compare. During this process, it was found that the KN1DPy iterations after the “1st generation” were not operating correctly, causing nonphysical results for the atomic hydrogen neutral density profiles. That being said, the “0 generation”, which does not include charge exchange or elastic collisions in the calculation, and the “1st generation”, which does, did yield meaningful results. Unfortunately, we were limited to only accessing the total IDL results. Therefore, we compared the sum of the Python “0th” and “1st generations” to the sum of every IDL generation. While this could be better, it provides a general understanding of the accuracy and performance of KN1DPy. As shown in Fig. 2.4, the IDL and Python outputs share similar shapes and widths. However, the Python outputs are smaller by two orders of magnitude. This discrepancy is expected since the KN1D outputs account for more atomic reactions, and KN1DPy does not.

Moreover, there appear to be some errors occurring in the scrape-off layer, as shown by the discrepancy of the atomic hydrogen profile left of the separatrix,  $x = 0.17$  m. The lack of calculations for elastic collisions further compounds these errors. Lastly, there is an issue in how KN1DPy determines the resolution of  $f_{H_2}$ ; there are far fewer points in this graph, and we are not seeing as much of the profile.

While we cannot currently include the later generations in the calculation, and there seem to be errors in the scrape-off layer and resolution of molecular hydrogen, we can still gather information on how these profiles change as electron temperature and densities increase. Depending on how the slope and maximums change, we can extrapolate useful insights into the role of charge exchange and the opacity of SPARC. Keeping in mind that we cannot discuss the specific values of neutral densities at any given point due to the aforementioned errors. More specifically, we can observe the neutral particle ionization trends and how the high electron temperatures and densities of SPARC could affect the neutral penetration depth compared to profiles

with low electron temperatures and densities.

# Chapter 3

## SPARC Predictive Analysis

Currently, the SPARC team uses code to produce models for ICRH heating, turbulent transport, pedestal structure, edge profiles, magnetohydrodynamics (MHD) stability, and ripple loss of fast alphas [9]. However, they lack code for predictive neutral density profiles. Using KN1DPy, we can produce neutral density profiles and, in turn, assess the opacity of SPARC. A majority of the KN1DPy inputs are geometric

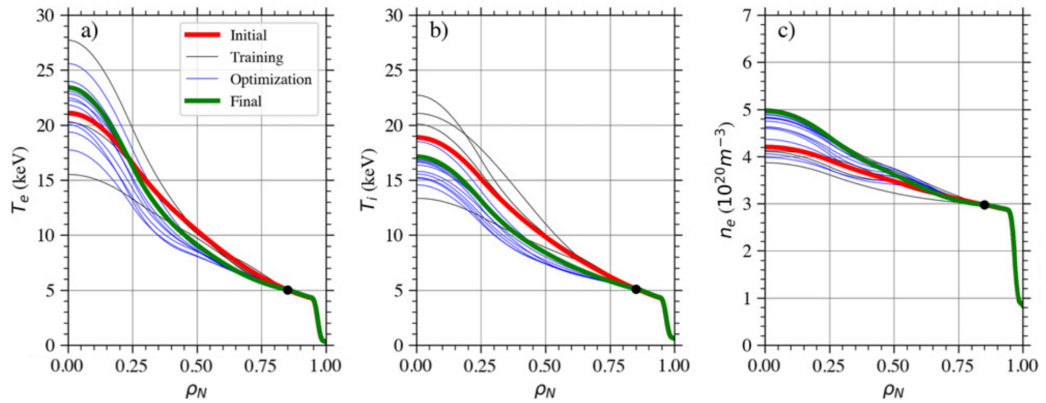


Figure 3.1: Simulated profiles for electron temperature (left), ion temperature (center) and electron density (right) [24].

parameters that have already been published for SPARC. These include  $x$ -values over the minor radius  $a = 0.57m$ , the separatrix coordinate, the wall coordinate, and the limiter. Since SPARC uses a divertor instead of a limiter, the limiter coordinate is treated as equal to the wall coordinate. Furthermore, SPARC has not published data

regarding their diagnostic probe geometry and pressure. Therefore, this portion of the geometry is borrowed from Alcator C-Mod. With that said, as more information is published, we could revise this in the future. Aside from the geometric parameters, the principle inputs are the background profiles for electron density ( $n_e$ ), ion temperature ( $T_i$ ), and electron temperature ( $T_e$ ), where the last two are usually equal. As this is a predictive analysis, we do not have experimental profiles; rather, the plasma profile inputs will be synthetic approximations.

In P. Rodriguez-Fernandez *et al*, CFS published simulated plasma profiles, shown in Fig. 3.1, that were simulated using TRANSP, an advanced transport code [24]. While each profile offers immense detail, there are few, and they do not cover a large range of possible temperature and densities at the edge. In response, KN1DPy will use simplified profiles based on the modified hyperbolic tangent function. This will allow us to better assess each parameter’s role in the plasma’s opacity.

When fitting data for plasma temperature and density in a tokamak, it is standard to use a modified hyperbolic tangent function [25]. This practice dates back to the 80s when the modified tanh approximation was empirically derived [26]. The modified hyperbolic tangent is shown in Eq. 3.1. It pairs the hyperbolic tangent function with the Heaviside function, resulting in the signature pedestal structure at the edge and a steep slope at the core.

$$\begin{aligned}
 Y &= A \tanh(2 \times (x_{SYM} - x)/W) + B \\
 Y_- &= Y + M(x_{KNEE} - x), \quad x < x_{KNEE}
 \end{aligned}
 \tag{3.1}$$

Here,  $A$  is simply the scaling parameter,  $x_{SYM}$  is the center of the pedestal structure, and  $W$  is the width of the pedestal structure. Moreover,  $B$  is the offset,  $M$  is the slope, and  $x_{KNEE}$  is the position at which the steep gradient of the pedestal rounds out [25][27].

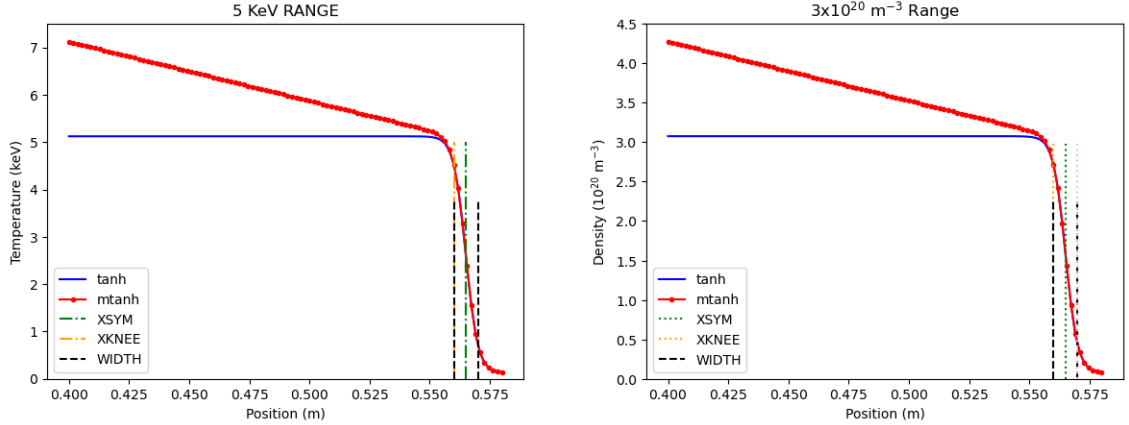


Figure 3.2: A synthetic ion temperature profile (left) and ion density profile (right).

Using Eq. 3.1, five temperature profiles ranging from an edge temperature of 2 keV to 10 keV were produced. In conjunction with the five temperature profiles, five density profiles ranging from  $0.6 \times 10^{20} \text{ m}^{-3}$  to  $3 \times 10^{20} \text{ m}^{-3}$  at the edge were also calculated. These ranges span the temperature and densities similar to those measured in C-MOD, progressing through the expected values of SPARC, all while using the SPARC geometry. We can then input each temperature profile into KN1DPy while the density profile is unchanged and vice versa. The resulting neutral density profiles, sweeping across both temperature and electron density produced, allow us to understand the impact of each parameter. The profiles in the upper half of that range are of particular interest because SPARC is expected to have edge measurements close to 10 keV and  $3 \times 10^{20} \text{ m}^{-3}$  [9]. Allowing us to develop expectations for SPARC’s opacity.

After obtaining initial simulation results, information regarding the opacity is determined based on the neutral penetration depth and the profile gradients. If the neutral density is concentrated outside the separatrix, i.e.,  $x > 0.57$ , or if the profiles have steep gradients at higher temperatures and densities, then plasma in SPARC will likely be more opaque to neutrals. Therefore, neutral recycling will play

a weaker role. Furthermore, we can compare the profiles with charge exchange to those without, demonstrating the degree to which charge exchange can affect neutral penetration depth and opacity. This is especially informative since Eq. 2.1 does not account for this reaction. If the profile has a less steep slope than those with charge exchange, one can infer that charge exchange does play a significant role in neutral penetration depth. This could imply that SPARC will be less opaque than Eq. 2.1 predicts.

# Chapter 4

## KN1DPy SPARC Profiles

Due to complications regarding iterating through the generations and the associated elastic collisions, we, unfortunately, cannot observe the complete effect of charge exchange on opacity. Furthermore, the results show that the program is not accurately calculating the neutral density profile in the scrape-off layer. Therefore, we cannot determine whether more neutrals are ionizing outside the separatrix at higher temperatures and electron densities. However, we can observe how the neutral density profiles will change as electron density and temperature increase. Additionally, we can gain insight into how charge exchange affects the neutral penetration depth by comparing neutral penetration depths at high electron temperatures and densities to those at low electron temperatures and densities.

Before examining theoretical data for SPARC, we examined the impact of increasing these parameters on profiles, which we have shown are within the physical expectations, keeping in mind that there are some errors. Consequently, we multiplied the electron density profile input from Alcator C-Mod by factors of 1 through 5. In Fig.4.1, one can see that increasing the electron density profile did cause a steeper gradient for molecular and atomic hydrogen compared to that of Fig.2.4. This fits the theory outlined in Eq.2.1, which states that opacity will increase as electron density increases.



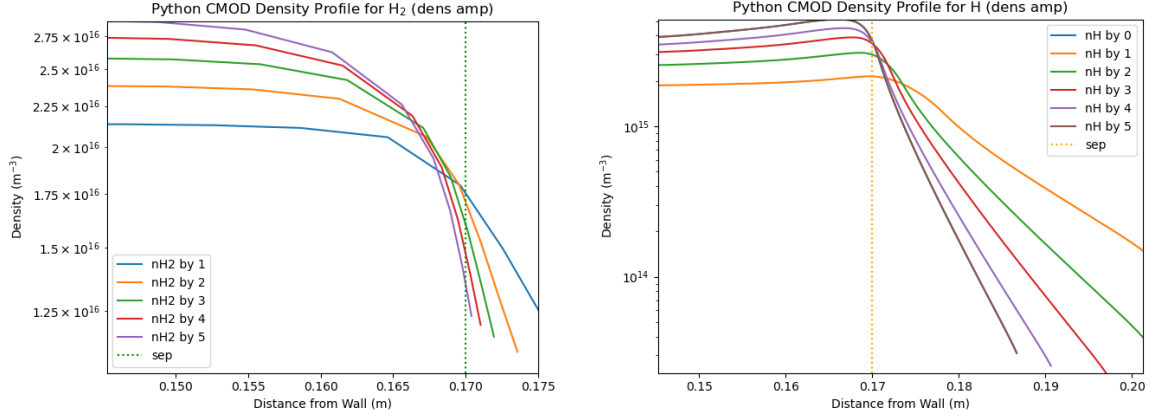


Figure 4.1: The neutral density profiles of molecular (left) and atomic (right) hydrogen based on the CMOD data sets. The profiles are labeled based on how much the initial electron density was multiplied by. Meaning the lowest density input, was the initial data multiplied by 1, and the highest was multiplied by 5.

After observing the effect of increasing the electron density profile for the Alcator C-Mod data, we proceeded with the synthetic data inputs. Before discussing the final results of these runs, it is essential to consider the mesh adjustments made to KN1DPy for the SPARC data. The initial molecular hydrogen neutral density profiles only included 3 points, as the `create_kinetic_h2_mesh` function decided. The lack of resolution inhibits meaningful extrapolation because few trends can be identified with a small data set. In response, we increased the upper cutoff for the x coordinate of molecular hydrogen to the maximum value, which still allows the program to run at a reasonable timescale. This resolution error is likely because the meshes could have been calculated based on C-MOD geometry, something we did not realize until significantly later.

Once we made the above observation, this minor adjustment allowed for superior results. Fig.4.2, displays the neutral density profiles for atomic and molecular hydrogen across a range of electron temperatures while the electron density stayed constant with an edge value of  $2 \times 10^{20} \text{ m}^{-3}$ . Here, we see that at the highest temperature, the purple plot, the neutral density at the separatrix is lowest. This aligns with ex-

pections since atomic collisions and ionization are more likely to occur at higher electron temperatures, resulting in lower neutral density. Furthermore, the gradient of the atomic hydrogen profile at high electron temperature is less steep, as evident by its intersection with the other profiles. Based on the previous explanation, one may expect that higher electron temperatures will result in more ionization in total. However, this gentle gradient is caused by charge exchange. As detailed in Section 2, charge exchange results in higher energy neutrals, allowing them to penetrate further into the plasma prior to ionization and, thus, a smaller gradient. So, the opacity does not align with the predictions outlined in 2.1 at high electron and ion temperatures.

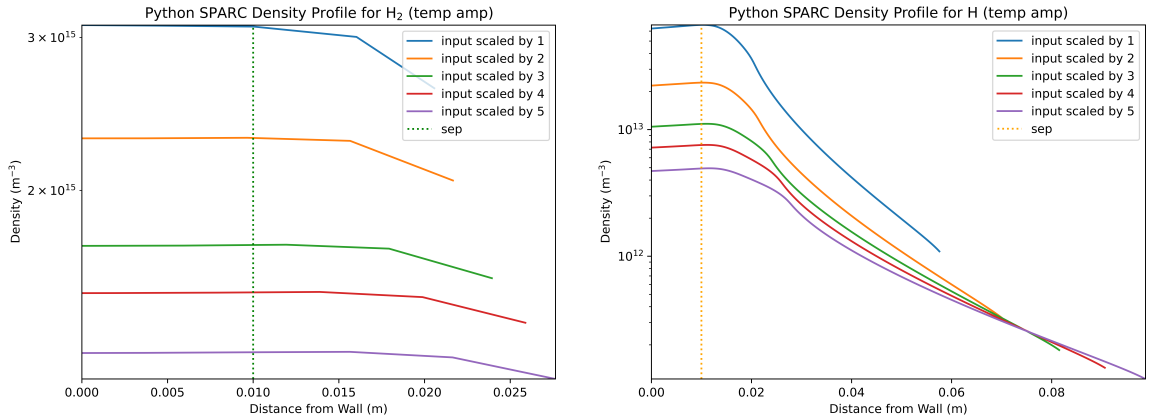


Figure 4.2: The neutral density profiles of molecular (left) and atomic (right) hydrogen based on the SPARC synthetic data sets with edge temperatures ranging from 2 keV to 10 keV and an edge density of  $2 \times 10^{20} \text{ (m}^{-3}\text{)}$ . The profiles are labeled based on how much the initial input temperature was scaled by. Meaning the lowest temperature input was 2 keV scaled by 1, and the highest was 2 keV scaled by 5 i.e. 10 keV.

The effects of charge exchange are further observed in Fig. 4.3, which compares the atomic hydrogen neutral density profile after the “0th generation” to after the “1st generation”. Since the “0th generation” does not include charge exchange but the “1st generation” does, this demonstrates the effect of charge exchange on neutral penetration depth. The “0th generation”, represented via the blue line, has a steeper gradient than that of the “1st generation”, represented via the orange line. Once

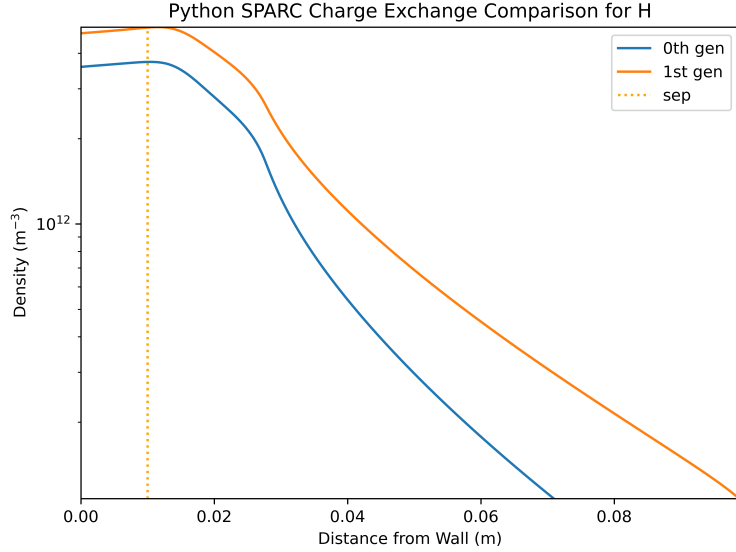


Figure 4.3: A comparison of the atomic hydrogen neutral density profiles with charge exchange and without.

again, the change in gradient is because charge exchange results in higher energy neutrals, which can penetrate further before ionization.

While the temperature sweeps demonstrate a strong influence from charge exchange, this trend is not translated to the electron density sweeps. In Fig.4.2, we see the neutral density profiles for atomic and molecular hydrogen across a range of electron densities while the electron temperature stayed constant with an edge value of 2 keV. In this graph, the profile with high electron density also has the highest neutral density. High ion and electron densities require a larger source, i.e., larger neutral densities. Furthermore, this profile has a steeper gradient than the others, which is the opposite result shown in the temperature sweeps, Fig.4.4. In conjunction with a steeper gradient, we can also see the neutral density global maximum occurring closer to the separatrix. This implies that the plasma will be more opaque at higher electron densities, so more neutrals will ionize outside the separatrix. These results demonstrate that high electron densities have a more substantial affect on the opacity than charge exchange. Therefore, the opacity of SPARC will align closely with the

expectations outlined in 2.1.

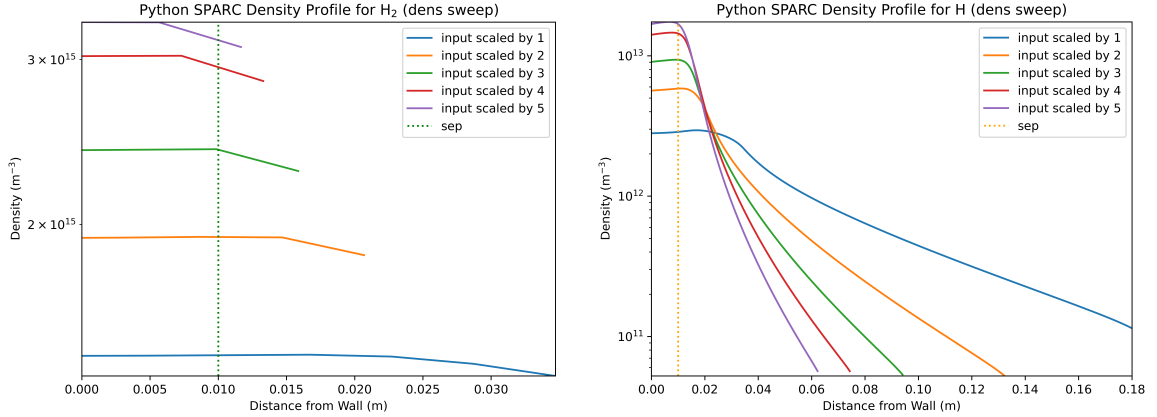


Figure 4.4: The neutral density profiles of molecular (left) and atomic (right) hydrogen based on the SPARC synthetic data sets with edge electron densities ranging from  $0.6 \times 10^{20}$  to  $3 \times 10^{20} \text{ m}^{-3}$  and an edge electron temperature of 7 keV. The profiles are labeled based on how much the initial input density was scaled by. Meaning the lowest temperature input was  $0.6 \times 10^{20}$  scaled by 1, and the highest was  $0.6 \times 10^{20}$  scaled by 5, i.e.  $3 \times 10^{20}$  scaled by 5.

While the effects of the high electron densities appear stronger than those of charge exchange, the gradients are still less steep than they could be. Fig.4.5 compares the atomic hydrogen neutral density profile after the “0th generation” to after the “1st generation”. The “0th generation”, represented via the blue line, once again has a steeper gradient than the “1st generation”, represented via the orange line. As before, the change in gradient is because charge exchange results in higher energy neutrals, which can penetrate further before ionization. So, while the high-temperature plasma causes steeper gradients overall, charge exchange somewhat diminishes that effect.

Due to the aforementioned setbacks regarding the scrape-off layer, the calculations of later generations, and elastic collision calculations, the results are somewhat incomplete. Still, the neutral density profiles outlined above corroborate the expectation that increasing the electron density of the plasma makes it more opaque to neutrals. At the same time, charge exchange can decrease the opacity at high temperatures, but soon, high electron densities overpower its effect. A higher concentration of neutrals

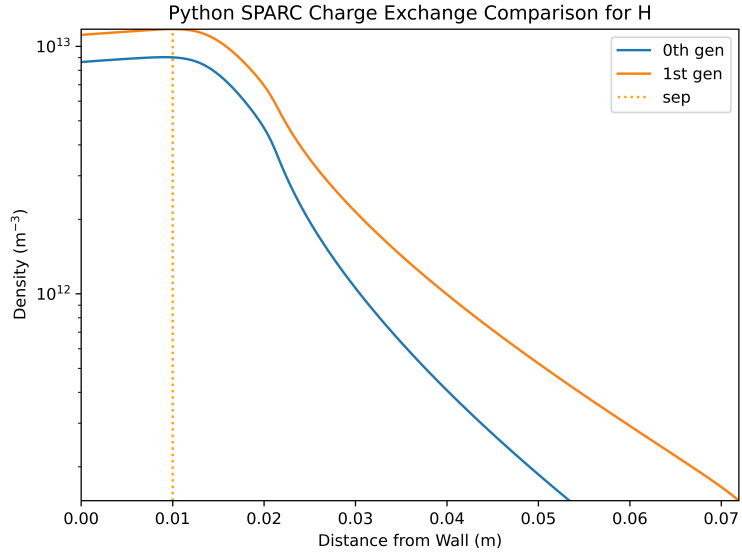


Figure 4.5: A comparison of the atomic hydrogen neutral density profiles for an edge electron density of  $3 \times 10^{20} \text{ m}^{-3}$  with and without charge exchange.

ionizing outside of the separatrix neutral recycling will likely have a weaker role in fueling the plasma core. As a result, we can expect that the SPARC team and other future tokamaks will have to rely on alternative fueling methods such as gas puffing and pellet injection [28].

# Chapter 5

## Conclusion

The next generation of tokamaks, including CFS’s SPARC, will have higher temperatures and electron densities, producing more opaque plasmas. Using KN1DPy, we generated neutral density profiles for SPARC, an area of predictive analysis that has yet to be explored. While the functionality to calculate the distribution functions after the “1st generation” and the elastic collisions are not yet ready, the neutral density profiles still showed overarching trends regarding the opacity of SPARC. More specifically, we saw in Fig.4.2 that as temperature increases, with a constant edge electron density of  $2 \times 10^{20} \text{ m}^{-3}$ , the neutral density at the separatrix decreases. This aligns with our expectations since higher temperatures result in more frequent collisions and ionization. Furthermore, the higher temperature ranges had less steep gradients, a sign that charge exchange significantly affects opacity at middle-range densities. On the other hand, Fig.4.4 demonstrated that as the electron density increases, there is higher neutral density at the separatrix and steeper gradients. The increase in neutral densities is expected because the high electron density requires large neutral sources. Not only were the neutral densities higher — their maximum was also closer to the separatrix. This shift, paired with the steep gradients, shows a decrease in the effect of charge exchange. So, the effect of high electron densities, as outlined in Eq.2.1, overpowers the countereffect associated with charge exchange.

Combining these results, we expect the high electron densities and temperatures of SPARC to cause a plasma with higher opacity, corroborating the prediction outlined by Eq.2.1. While charge exchange does play a vital role in extending the neutral penetration depth, it is not stronger than the increased ionization caused by high electron and ion densities. The increased opacity and shortened penetration depth imply that neutral recycling can not support fueling to the same extent that it has in previous machines. Therefore, SPARC and other future tokamaks will need to rely on alternative forms of fueling such as gas puffing and pellet injection [28].

Lastly, we could not gather sufficient data on whether more neutral particles would ionize outside the separatrix due to errors in the scrape-off layer. This requires further investigation in future work. Moreover, the results were limited due to the lack of calculations past the “1st generation”. To expand on the above results, it would be ideal to revise KN1DPy further to allow for this functionality. Once revised, not only would it give more accurate results, but it would also allow for more discussion surrounding the role of charge exchange in determining the opacity of the plasma. Currently, the charge exchange calculations are limited by the number of generations. With the revised code, charge exchange could likely play a more significant role, allowing for an expanded discussion regarding the opacity of SPARC and neutral recycling’s role in fueling the plasma core.

# Appendix A

## IDL to Python

Shown below in Table A.1, are a series of commonly used translations from IDL to Python. For ease of reading the Python libraries NumPy and SciPy are abbreviated to np and sp respectively. Furthermore, IDL differentiates a floating-point array from a double-precision, floating-point array. This is often denoted by a function starting with a “d” for double-precision and “f” otherwise. Python, however, does not make this distinction and so several Python functions are repeated in the table below.

Table A.1: A table demonstrating some translations between IDL and Python, where  $m$  and  $n$  are the dimensions of the chosen array. Furthermore,  $A$  is the array of interest and  $a$  is simply a specified integer.

IDL	Python	Comments
<code>fltarr(<math>n, m</math>)</code>	<code>np.zeros(<math>m, n</math>)</code>	Creates an array full of zero with $m$ rows and $n$ columns
<code>dblarr(<math>n, m</math>)</code>	<code>np.zeros(<math>m, n</math>)</code>	Creates an array full of zero with $n$ rows and $m$ columns
<code>dindgen(<math>n</math>)</code>	<code>np.arange(<math>n</math>)</code>	Creates a 1D array where each element counts up from 0 to $n - 1$
<code>findgen(<math>n</math>)</code>	<code>np.arange(<math>n</math>)</code>	Creates a 1D array where each element counts up from 0 to $n - 1$



<code>lindgen(<math>n</math>)</code>	<code>np.arange(<math>n</math>)</code>	Creates a 1D array where each element counts up from 0 to $n - 1$
<code>reform(<math>a, n, m</math>)</code>	<code>np.shape(<math>a, (m, n)</math>)</code>	Reshapes the array $a$ to be the specified dimensions. In this case $a$ ends with $m$ rows and $n$ columns.
<code>reverse(<math>a</math>)</code>	<code>np.fliplr(<math>a</math>)</code>	Reverses the rows of the array $a$ ; the columns are unaffected.
<code>reverse(<math>a, 2</math>)</code>	<code>np.flipud(<math>a</math>)</code>	Reverses the columns of the array $a$ ; the rows are unaffected.
<code>shift(<math>a</math>)</code>	<code>np.roll(<math>a</math>)</code>	Shifts the elements of array $a$ over one. Elements that would shift off the end of $a$ are put at the beginning.
<code>n_elements(<math>a</math>)</code>	<code>np.size(<math>a</math>)</code>	Returns the number of elements in array $a$ .
<code>total(<math>a</math>)</code>	<code>np.sum(<math>a</math>)</code>	Takes the sum of every element in array $a$ .
<code>where(<math>a &gt; n</math>)</code>	<code>np.argwhere(<math>a &gt; n</math>).T</code>	Finds the indices which satisfy the condition and generates them in an array. In this case the condition is when the element of $a$ is greater than the variable $n$ .
<code>interpol(<math>y, x</math>)</code>	<code>sp.interpolate.interp1d(<math>x, y</math>)</code>	1D interpolation over $x$ and $y$ arrays.

# Appendix B

## Code Sample for Synthetic Data

The following is the Python code that populates the synthetic data for a range of temperatures using a modified hyperbolic tangent function. It then runs KN1DPy using these data sets and plots the results.

```
/* **** */
/* Gwendolyn R. Galleher */

from KN1D import KN1D
import numpy as np
import matplotlib.pyplot as plt
import scipy.io as sio
from scipy.io import readsav
from scipy import interpolate
def read_sav(file_name):

# This function will print the data from an IDL .sav file
# Inputs:
# file name - the name of the .sav file
# Outputs:
# sav_data - a dictionary containing all of the data
    path = '//Users/Gwen/Desktop/Plasma_Physics/kn1d/' + file_name
    sav_data = readsav(path)
    return sav_data
# Pull Cmod Data for the LC and PipeDia parameters
data_file = read_sav('1090904018_950to1050.sav')

keys = data_file.keys() # gets the keys from the dictionary
keys_list = list(keys) # puts the keys into a list
print(keys_list)

values = data_file.values() # gets the values from the dictionary
```

```

values_list = list(values)

LC = values_list[14]
PipeDia = values_list[15]

# generate empty lists to save data
nH2_tot = [[], [], [], [], []]
nH_tot = [[], [], [], [], []]
xH2_tot = [[], [], [], [], []]
xH_tot = [[], [], [], [], []]

# generate text file to save data to
save_data = open("kn1d_sparc_temp_sweep_data.txt", "w")
save_data.write("Data for KN1DPy SPARC Temperature sweep: \n")
save_data.close()

for i in range(1, 6):

    # synthetic data for temperature
    a = i
    b = a + a/20

    CORE = 0.58 - 0.4
    WALL = 0.58 - 0.58
    X_sep = 0.58 - 0.57

    WIDTH = 0.01
    XSYM = X_sep + WIDTH/2
    XKNEE = XSYM + WIDTH/2

    SLOPE = 5*a

    pts = 61
    x = np.linspace(WALL, CORE, pts)
    SP = round(((CORE - XKNEE) / (CORE - WALL) * pts))

    x_ = np.linspace(XKNEE, CORE, SP)
    y = a * np.tanh(-(2 * (XSYM - x) / WIDTH)) + b
    y_ = y - np.append( np.zeros(pts - SP), SLOPE * (XKNEE - x_))

    Ti = y_ * 1000
    Te = y_ * 1000

    # Synthetic data for density
    a = 1
    b = a + a/20

    CORE = 0.58 - 0.4
    WALL = 0.58 - 0.58
    X_sep = 0.58 - 0.57

```

```

WIDTH = 0.01
XSYM = X_sep + WIDTH/2
XKNEE = XSYM + WIDTH/2

SLOPE = 5*a

pts = 61
x = np.linspace(WALL, CORE, pts)
SP = round(((CORE - XKNEE) / (CORE - WALL) * pts))

print(SP)
x_ = np.linspace(XKNEE, CORE, SP)
y = a * np.tanh(-(2 * (XSYM - x) / WIDTH)) + b
y_ = y - np.append( np.zeros(pts - SP), SLOPE * (XKNEE - x_))

ne = y_ * 10**20

xlim = WALL
xsep = X_sep
GaugeH2 = 0.100
mu = 2
vx = np.zeros(61)

print('x', x.size, x)
print('xlim', xlim)
print('xsep', xsep)
print('GaugeH2', GaugeH2)
print('mu', mu)
print('Ti', Ti.size, Ti)
print('Te', Te.size, Te)
print('ne', ne.size, ne)
print('vx', vx.size, vx)
print('LC', LC.size, LC)
print('PipeDia', PipeDia.size, PipeDia)

# KN1D(x, xlimiter, xsep, GaugeH2, mu, Ti, Te, n, vxi, LC, PipeDia, \
#      truncate = 1.0e-3, refine = 0, File = '', NewFile = 0, ReadInput = 0, \
#      error = 0, compute_errors = 0, plot = 0, debug = 0, debreif = 0, pause = 0, \
#      Hplot = 0, Hdebug = 0, Hdebreif = 0, Hpause = 0, \
#      H2plot = 0, H2debug = 0, H2debreif = 0, H2pause = 0)

xH2, nH2, GammaxH2, TH2, qxH2_total, nHP, THP, SH, SP, \
      xH, nH, GammaxH, TH, qxH_total, NetHSource, Sion, QH_total, SideWallH, \
      Lyman, Balmer = KN1D( x, xlim, xsep, GaugeH2, mu, Ti, Te, ne, vx, LC, PipeDia)
# save data to lists
nH2_tot[i - 1] = nH2.tolist()
nH_tot[i - 1] = nH.tolist()
xH2_tot[i - 1] = xH2.tolist()
xH_tot[i - 1] = xH.tolist()

# calculate gradient

```

```

f_H2 = np.array([xH2, nH2])
f_H = np.array([xH, nH])

grad_H2 = np.gradient(f_H2)
grad_H = np.gradient(f_H)
print(grad_H2)

#normalized gradient
nH2_sep = interpolate.interp1d(xH2, nH2, fill_value='extrapolate')(xsep)
ngrad_H2 = grad_H2/nH2_sep

nH_sep = interpolate.interp1d(xH, nH, fill_value='extrapolate')(xsep)
ngrad_H = grad_H/nH_sep

# save data into txt file
save_data = open("kn1d_sparc_temp_sweep_data.txt", "a")

data = [f"xH2_{i}: {xH2} \n", f"nH2_{i}: {nH2} \n", \
f"xH_{i}: {xH} \n", f"nH_{i}: {nH} \n"]
save_data.writelines(data)

save_data.writelines("Gradient Data \n")

grad_data = [f"grad_H2_{i}: {grad_H2} \n", f"grad_H_{i}: {grad_H} \n", \
f"ngrad_H2_{i}: {ngrad_H2} \n", f"ngrad_H_{i}: {ngrad_H} \n"]
save_data.writelines(grad_data)
save_data.close()

print('nH2', nH2)
print('GammamaxH2', GammamaxH2)
print('TH2', TH2)
print('qxH2_total', qxH2_total)
print('nHP', nHP)
print('THP', THP)
print('SH', SH)
print('SP', SP)
print('xH', xH)
print('nH', nH)
print('GammamaxH', GammamaxH)
print('TH', TH)
print('qxH_total', qxH_total)
print('NetHSource', NetHSource)
print('Sion', Sion)
print('QH_total', QH_total)
print('SideWallH', SideWallH)
print('Lyman', Lyman)
print('Balmer', Balmer)

print('Press any key to continue')
input()
print('continuing')

```

```

nH2_mins = []
nH2_maxs = []
xH2_mins = []
xH2_maxs = []
nH_mins = []
nH_maxs = []
xH_mins = []
xH_maxs = []

for i in range(0, 5):
    nH2_mins.append(min(nH2_tot[i]))
    nH2_maxs.append(max(nH2_tot[i]))
    xH2_mins.append(min(xH2_tot[i]))
    xH2_maxs.append(max(xH2_tot[i]))
    nH_mins.append(min(nH_tot[i]))
    nH_maxs.append(max(nH_tot[i]))
    xH_mins.append(min(xH_tot[i]))
    xH_maxs.append(max(xH_tot[i]))
plt.figure(constrained_layout=True)
for i in range(1, 6):
    plt.plot(xH2_tot[i - 1], nH2_tot[i - 1], linestyle = 'solid', \
            label = f'input scaled by {i}')
plt.plot(np.full(10, 0.01), np.linspace(min(nH2_mins), max(nH2_maxs), 10), \
        linestyle = ':', color = 'green', label = 'sep')
plt.xlabel('Distance from Wall (m)')
plt.ylabel('Density (m-3)')
plt.yscale("log")
plt.ylim(min(nH2_mins), max(nH2_maxs))
plt.xlim(min(xH2_mins), max(xH2_maxs))
plt.title(f'Python SPARC Density Profile for H2 (temp amp)')
plt.legend()
plt.savefig(f'/Users/Gwen/Desktop/KN1DPy-Nick/Plots/nH2_sparc_temp_py_HR.png', dpi = 800)
plt.show()

print('Press any key to continue')
input()
print('continuing')

plt.figure(constrained_layout=True)
for i in range(1, 6):
    plt.plot(xH_tot[i - 1], nH_tot[i - 1], linestyle = 'solid', \
            label = f'input scaled by {i}')
plt.plot(np.full(10, 0.01), np.linspace(min(nH_mins), max(nH_maxs), 10), \
        linestyle = ':', color = 'orange', label = 'sep')
plt.xlabel('Distance from Wall (m)')
plt.ylabel('Density (m-3)')
plt.yscale("log")
plt.ylim(min(nH_mins), max(nH_maxs))
plt.xlim(min(xH_mins), max(xH_maxs))
plt.title(f'Python SPARC Density Profile for H (temp amp)')

```

```
plt.legend()  
plt.savefig(f'/Users/Gwen/Desktop/KN1DPy-Nick/Plots/nH_sparc_temp_py_HR.png', dpi = 800)  
plt.show()
```





# Bibliography

- [1] S.E. Vollset, E. Goren, C.-W. Yuan, J. Cao, A.E. Smith, T. Hsiao, C. Bisignano, G.S. Azhar, E. Castro, J. Chalek, A.J. Dolgert, T. Frank, K. Fukutaki, S.I. Hay, R. Lozano, A.H. Mokdad, V. Nandakumar, M. Pierce, M. Pletcher, T. Robalik, K.M. Steuben, H.Y. Wunrow, B.S. Zlavog, and C.J.L. Murray, *The Lancet* 396, 1285 (2020).
- [2] IEA (2019), *World Energy Balances 2019*, OECD Publishing, Paris, <https://doi.org/10.1787/3a876031-en>.
- [3] *Climate Change: Global Temperature* — NOAA Climate.gov. <http://www.climate.gov/news-features/understanding-climate/climate-change-global-temperature> (accessed 2023-12-04).
- [4] R.A. Serway, C.J. Moses, and C.A. Moyer, *Modern Physics*, 3rd ed (Thomson Brooks/Cole, Belmont, CA, 2005).
- [5] Britannica, T. Editors of Encyclopaedia. nuclear binding energy. *Encyclopedia Britannica*. <https://www.britannica.com/science/nuclear-binding-energy> (2023)
- [6] F.F. Chen, *Introduction to Plasma Physics and Controlled Fusion* (Springer International Publishing, Cham, 2016).
- [7] D.J. Griffiths, *Introduction to Electrodynamics*, Fourth edition (Pearson, Boston, 2013).
- [8] Devens Updates. <https://cfs.energy/devens-campus/updates> (accessed 2024-03-31).
- [9] A.J. Creely, M.J. Greenwald, S.B. Ballinger, D. Brunner, J. Canik, J. Doody, T. Fülöp, D.T. Garnier, R. Granetz, T.K. Gray, C. Holland, N.T. Howard, J.W. Hughes, J.H. Irby, V.A. Izzo, G.J. Kramer, A.Q. Kuang, B. LaBombard, Y. Lin, B. Lipschultz, N.C. Logan, J.D. Lore, E.S. Marmor, K. Montes, R.T. Mumgaard, C. Paz-Soldan, C. Rea, M.L. Reinke, P. Rodriguez-Fernandez, K. Särkimäki, F. Sciortino, S.D. Scott, A. Snicker, P.B. Snyder, B.N. Sorbom, R. Sweeney, R.A. Tinguely, E.A. Tolman, M. Umansky, O. Vallhagen, J. Varje, D.G. Whyte, J.C. Wright, S.J. Wukitch, J. Zhu, and the SPARC Team, *J. Plasma Phys.* 86, 865860502 (2020).
- [10] S.E. Wurzel and S.C. Hsu, *Physics of Plasmas* 29, 062103 (2022).
- [11] S. Mordijck, *Nucl. Fusion* 60, 082006 (2020).

- [12] M.F. Modest, Radiative Heat Transfer (Elsevier Science & Technology, San Diego, UNITED STATES, 2003).
- [13] R. Reksoatmodjo, S. Mordijck, J.W. Hughes, J.D. Lore, and X. Bonnin, Nuclear Materials and Energy 27, 100971 (2021).
- [14] B.H. Bransden, Rep. Prog. Phys. 35, 949 (1972).
- [15] A.E. Costley, Nucl. Fusion 56, 066003 (2016).
- [16] I.A.E. Agency, Summary of the ITER Final Design Report (International Atomic Energy Agency, 2001), p. 1.
- [17] B. LaBombard, KN1D: A 1-D Space, 2-D Velocity, Kinetic transport algorithm for atomic and molecular hydrogen in an ionizing plasma, MIT Plasma Science and Fusion Center Report PSFC/RR-01-3; Research Report PSFC/RR-01-3
- [18] C. Cerignani, The Boltzmann Equation and Its Applications, Springer-Verlag (1988)
- [19] Sawada, K. and Fujimoto, T., Journal of Applied Physics 78 (1995) 2913.
- [20] J.W. Hughes, B. LaBombard, D.A. Mossessian, A.E. Hubbard, J. Terry, T. Biewer, and T.A.C.-M. Team, Physics of Plasmas 13, 056103 (2006).
- [21] G.W. Stuart and R.W. Woodruff, Nuclear Science and Engineering (1958).
- [22] J.R. Haack, C.D. Hauck, and M.S. Murillo, J Stat Phys 168, 826 (2017).
- [23] D. Reiter, M. Baelmans, and P. Börner, Fusion Science and Technology 47, 172 (2005).
- [24] P. Rodriguez-Fernandez, N.T. Howard, and J. Candy, Nucl. Fusion 62, 076036 (2022).
- [25] R.J. Groebner and T.H. Osborne, Scaling Studies of the H-Mode Pedestal (General Atomics, San Diego, CA (United States), 1998).
- [26] F. Wagner and K. Lackner, Physics of Plasma-Wall Interactions in Controlled Fusion (edited by D.E. Post and R. Behrisch, published by Plenum Press, New York in cooperation with NATO Scientific Affairs Division), Series B, Physics Vol. 131, 931 (1986).
- [27] R.J. Groebner, D.R. Baker, K.H. Burrell, T.N. Carlstrom, J.R. Ferron, P. Gohil, L.L. Lao, T.H. Osborne, D.M. Thomas, W.P. West, J.A. Boedo, R.A. Moyer, G.R. McKee, R.D. Deranian, E.J. Doyle, C.L. Rettig, T.L. Rhodes, and J.C. Rost, Nucl. Fusion 41, 1789 (2001).
- [28] Refuelling. <https://www.ipp.mpg.de/14881/brennstoff> (accessed 2024-05-10).