Submitted as a part of

Summer Internship Programme 2025,
Physical Research Laboratory (PRL)

on the topic

Evaluating Plasma Parameters from a typical LIBS Plasma

by

Anuprovo Debnath
23MS110
IISER Kolkata

under the guidance of

Dr. Prashant Kumar
Atomic, Molecular and Optical Physics,
Physical Research Laboratory (PRL), Ahmedabad

भौतिक अनुसंधान प्रयोगशाला, अहमदाबाद

Physical Research Laboratory Ahmedabad

SUMMER INTERNSHIP PROJECT REPORT

CERTIFICATE

This is to certify that the work embodies in this summer internship project entitled "**Evaluating plasma parameters from a typical LIBS plasma**" being submitted by "**Mr. Anuprovo Debnath**" of **IISER, Kolkata** for partial fulfillment of the requirement for the award of "Certificate" during the 13th May - 30th June, 2025 of PRL's Summer internship programme 2025 is a record of a bonafide and original piece of work, which was carried out by him under my supervision and guidance in the Atomic, Molecular and Optical Physics Division of PRL, Ahmedabad.

Dr. Prashant Kumar
Scientist/Engineer-SF
PRL, Ahmedabad

Date: 08/10/2025

Abstract

This project focuses on evaluating plasma parameters from a typical Laser-Induced Breakdown Spectroscopy (LIBS) plasma, with particular emphasis to assess plasma parameters. A computational framework was developed using Python to extract key plasma parameters, such as electron temperature (T_e) and electron density (n_e) from spectroscopic data. The NIST Atomic Spectra Database was used to obtain spectral line data for copper (Cu I and Cu II) and aluminium (Al I and Al II), including ionisation energies, statistical weights, and transition probabilities.

The Boltzmann plot method was used to determine the electron temperature while the Saha–Boltzmann approach and Stark broadening analysis of the Hα line were used to estimate the electron density. The methodology was verified on database-based spectra prior to being used on experimental LIBS spectra of copper captured using a Q-switched Nd:YAG laser and ICCD-equipped spectrometer.

The findings show that when precise line identification and database validation are used, calibration-free LIBS (CF-LIBS) can consistently provide plasma parameters. In addition to providing insights into LTE plasmas, this work lays the foundation for future multi-elemental and machine learning-assisted LIBS studies by establishing a computational and experimental framework for plasma diagnostics.

Acknowledgements

I would like to express my heartfelt gratitude to Dr. Prashant Kumar for his guidance that was invaluable, discussions that were insightful and all the wholesome motivation throughout my summer research internship at the Physics Research Laboratory. It was his guidance alone that shaped how I approach every task and interpret all the difficult jargon from Laser-Induced Breakdown Spectroscopy (LIBS). I could not have been more happy to have him as a mentor.

I feel really happy for all my lab members–Dr. Kavil Mehta, Mr. Darshit Parmar, and Ms. Swetapuspa Soumyashree for the support they have given me; for their amazing guidance, and for making my time in the lab both fun and educational. None of the advancements I made would have been possible without them being always ready to help me in any way possible. All these efforts has had tremendous effect on the outcome of the project.

Overall, this internship has been a genuinely enriching experience.

Contents

Chapter 1

Introduction

Laser-Induced Breakdown Spectroscopy (LIBS) is an advanced atomic emission spectroscopy technique that enables rapid, in-situ, multi-elemental analysis of materials. By focusing an intense, high-energy laser pulse on a sample surface, LIBS creates a microplasma from the ablated material. The plasma emits characteristic light corresponding to the elemental composition of the sample, allowing spectral analysis to identify and quantify the elements present.[1]

The primary objective of this project was to understand the physical foundations of plasma and laser operation, specifically the Nd:YAG laser used in LIBS. A key focus was the evaluation of critical plasma parameters, namely the electron temperature (T_e) and electron number density (n_e), from LIBS emission spectra. These parameters are vital for quantitative analysis and plasma diagnostics.

To accomplish this, I studied fundamental plasma theory[2], the principles of laser operation[3], and the working of LIBS technique. Additionally, a computational tool was implemented to analyse spectral line data from the NIST Atomic Spectra Database[4] to evaluate plasma parameters using the Boltzmann plot and Stark broadening methods.

1.1 Plasma Basics

Plasma, often referred to as the fourth state of matter, results from a gaseous state having undergone some degree of ionization. It thus consists of a significant portion of charged particles (ions and/or electrons) as well as some neutral species. While rarely encountered on Earth, it is estimated that 99.9% of all ordinary matter in the universe is plasma. Thus, plasma is an ionized gas consisting of free electrons, ions, and neutral particles. It is electrically conducting and exhibits collective behaviour due to long-range electromagnetic interactions.[5]

1.1.1 Types of Plasma

There are primarily two types of plasma based on temperature.

1. **Thermal (Hot) Plasma:** Electrons, ions, and neutrals are all at roughly the same temperature: $T_e \approx T_i \approx T_n$. They are typically very hot in the range of thousands to millions of Kelvin ($\sim 1\,\text{eV} - 10\,\text{keV}$). It is strongly ionized, often close to fully ionized and is found in stars (e.g., Sun's core and corona), fusion reactors (tokamaks, inertial confinement), lightning arcs, etc.

2. **Non-Thermal (Cold) Plasma:** Also called non-equilibrium plasma. Electrons are very hot ($T_e \sim 1 - 10eV$) but ions and neutrals remain cold (near room temperature): $T_e \gg T_i \approx T_n$. Average gas temperature may be near room temperature, despite high-energy electrons and are partially ionized as many neutrals remain. They are found in Aurora borealis, plasma TVs, etc.

1.1.2 Properties of Plasma

Key plasma properties include[6,7]:

- **Quasi-neutrality:** On macroscopic scales, the number of positive and negative charges is nearly equal. Despite being made of ions and electrons, the net charge density ≈ 0. This balance causes plasma to behave differently from just a cloud of ions or electrons.
- **Collective behaviour:** Plasma particles interact via long-range electromagnetic forces, not just collisions. This leads to complex waves, oscillations, and instabilities.
- **Conductivity:** Plasmas conduct electricity due to the presence of free electrons and ions.
- **Response to fields:** Plasmas can be influenced and shaped by electric and magnetic fields (unlike gases).

1.1.3 Important Plasma parameters:

1. **Electron Temperature (T_e):** It is a measure of average kinetic energy of electrons and defines how energetic the electrons are (controls ionization and excitation rates).
2. **Electron Number Density (n_e):** It is the number of free electrons per unit volume (typically in cm^{-3}) and determines plasma conductivity and opacity.
3. **Degree of Ionization (α):** It is the ratio of ionized to total particles (ions + neutrals) and can vary from weakly ionized (cold plasma) to fully ionized (fusion plasma).
$$\alpha = \frac{n_i}{n_i + n_n}$$
4. **Plasma Frequency (ω_{pe}):** It is the natural oscillation frequency of electrons against ions and determines how plasma interacts with electromagnetic waves.
$$\omega_{pe} = \sqrt{\frac{n_e e^2}{\epsilon_0 m_e}}$$
5. **Debye Length (λ_D):** It is the distance over which electric potentials are screened in plasma and defines the scale of plasma "collective behaviour".
$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B T_e}{n_e e^2}}$$
6. **Plasma Parameter (N_D):** It represents the number of electrons contained within a Debye sphere.
$$N_D = \frac{4}{3}\pi n_e \lambda_D^3$$

1.1.4 Plasma criteria

A gas can be classified as a plasma if it satisfies certain criteria such that it behaves differently from an ordinary gas. The four main plasma criteria are[6,7]:

1. **Quasi-neutrality:** On macroscopic scales, plasma must contain almost equal numbers of positive and negative charges. This ensures the plasma is overall electrically neutral, though small local charge separations can exist.

$$n_e \approx \sum_i Z_i n_i$$

   where n_e is the electron number density and Z_i and n_i denote the ionic charge and ionic density of ith ionic species, respectively.

2. **Debye Shielding (Existence of a Debye Length):** A plasma must be able to shield out electric fields through charge rearrangement. The characteristic shielding length is the Debye length (λ_D). For plasma to exist, the physical system size (L) must be much larger than λ_D:

$$L \gg \lambda_D$$

3. **Plasma Parameter ($N_D \gg 1$):** It is the number of particles in a Debye sphere. For collective behaviour to dominate over individual particle interactions,

$$N_D \gg 1$$

   This ensures long-range electromagnetic effects are important.

4. **Plasma Frequency vs Collision Frequency:**The plasma must support collective oscillations of electrons (plasma frequency ω_{pe}) that are faster than particle collisions.

$$\omega_{pe} \gg \nu_{coll}$$

   This ensures collective plasma oscillations dominate over collisions.

1.2 Basics of LASER

LASER stands for **L**ight **A**mplification by **S**timulated **E**mission of **R**adiation. It is a device that produces a highly Intense, Monochromatic, Coherent, and Directional beam of light.[8]

1.2.1 Principle of a Laser

The operation of a laser is based on Einstein's concept of stimulated emission (1917)[3]:

1. **Spontaneous emission:** Excited atoms randomly emit photons.
2. **Stimulated emission:** An incoming photon causes an excited atom to emit a photon with the same phase, direction, and energy; this is the basis of laser action.
3. **Result:** A cascade of identical photons gives rise to coherent light amplification.

1.2.2 Essential Components of a Laser

1. **Active medium (gain medium):** It is the material where stimulated emission occurs. It can be a solid (Nd:YAG), gas (He-Ne, CO_2), liquid dye, or semiconductor.
2. **Pump source (excitation source):** This provides external energy to excite atoms into higher states. Examples: Flashlamps, electrical discharge, chemical reaction, diode lasers.
3. **Optical resonator (cavity mirrors):** This consists of two mirrors placed around the gain medium - one fully reflective, the other partially reflective letting a fraction of light escape as the laser beam. This ensures photons bounce back and forth, stimulating further emission.

1.2.3 Steps in Laser Action

1. **Pumping** – Energy is supplied to excite atoms into higher energy levels.
2. **Population inversion** – More atoms in excited state than in ground state leading to a non-equilibrium condition.
3. **Stimulated emission** – Incoming photons trigger emission of identical photons.
4. **Amplification** – Photons bounce inside the cavity, multiplying rapidly.
5. **Output coupling** – A coherent, monochromatic laser beam emerges through the partially transparent mirror.

1.2.4 Types of Laser

Lasers are classified by their gain medium. The four main types are:

1. **Solid-State Lasers:** They consist of solid crystal doped with rare-earth ions (e.g., Nd:YAG: Neodymium-doped Yttrium Aluminium Garnet) with a wavelength range in near-infrared, typically $\sim 0.7 - 3\,\mu m$.

   Example: Nd:YAG $\rightarrow$ 1064 nm (infrared), frequency-doubled to 532 nm (green).

2. **Gas Lasers:** Gain medium is gas atoms, ions, or molecules with wavelength range in Visible to far-infrared, $\sim 0.2 - 10\mu m$ (depends on gas).

   Example: He-Ne laser $\rightarrow$ 632.8 nm (red), CO_2 laser $\rightarrow$ 10.6 µm (IR).

   Applications: Cutting, welding, holography, alignment.

3. **Dye Lasers:** The gain medium is organic dye molecules dissolved in liquid solvent with wavelength range across 400 – 900 nm (visible spectrum).

4. **Semiconductor (Diode) Lasers:** The gain medium is semiconductor p–n junction with a wavelength range of 370 nm (UV) – 1550 nm (IR) depending on semiconductor material.

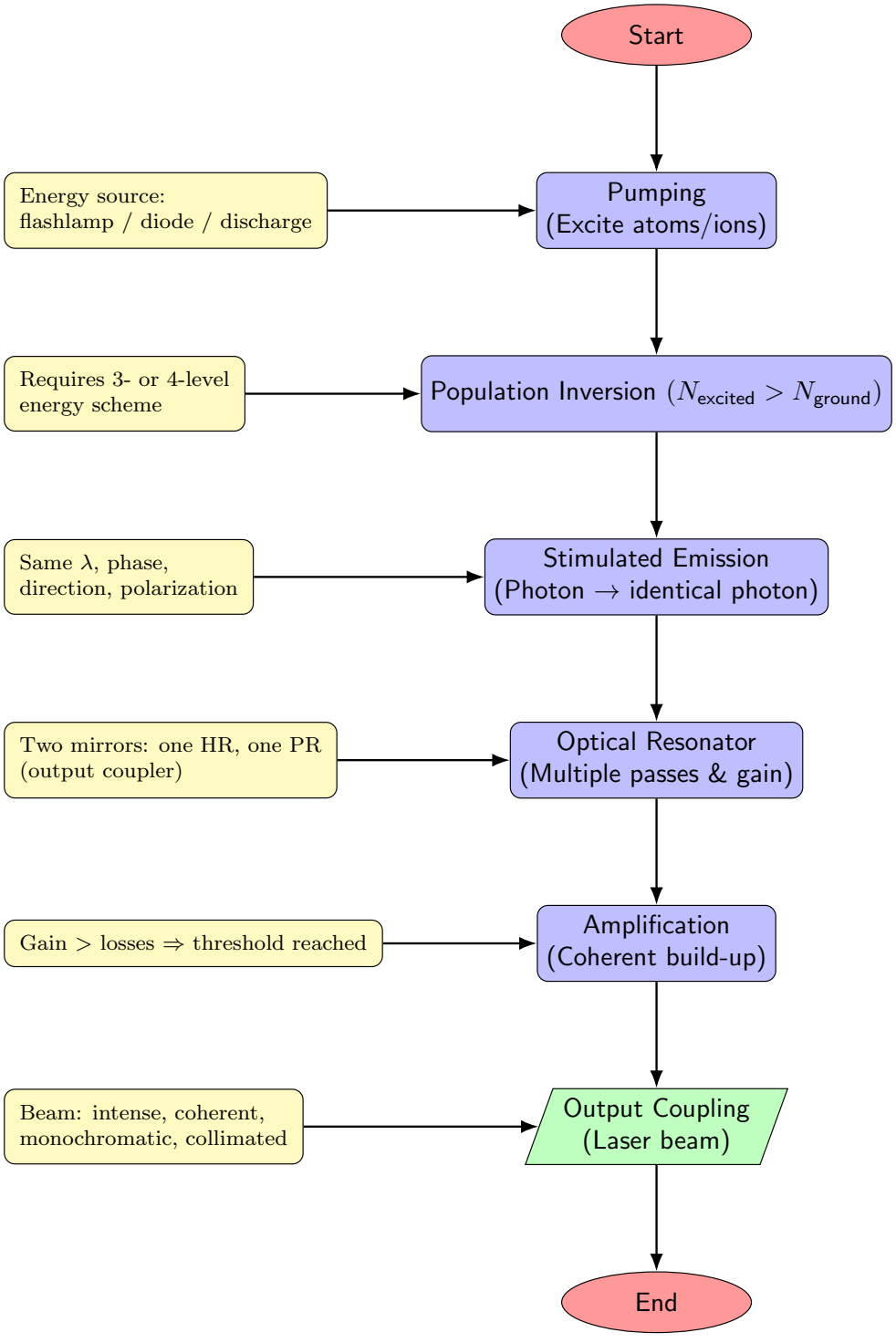


Figure 1.1: Steps in LASER Action

1.3 Working of Nd:YAG Laser

1.3.1 Active Medium (Gain Medium)

The gain medium consists of Nd:YAG = Neodymium-doped Yttrium Aluminium Garnet ($Y_3Al_5O_{12}$). Here the Nd^{3+} ions act as the lasing species. Common lasing transition is 1064 nm (infrared), but frequency-doubled outputs at 532 nm (green) are also common[9].

1.3.2 Pumping (Excitation Source)

Optical pumping is done by flashlamps (older systems), or laser diodes (modern systems). The pump light excites Nd^{3+} ions from the ground state to higher energy levels.

1.3.3 Energy Level Scheme

The Nd^{3+} ions in YAG have a four-level laser system:

a) *Ground state*: Ions start here.

b) *Excited state*: Pump photons excite Nd^{3+} ions to a high energy level.

c) *Metastable state*: Ions quickly decay non-radiatively into this long-lived state.

d) *Lower laser level*: Ions drop to this level when they emit a photon at 1064 nm.

1.3.4 Stimulated Emission

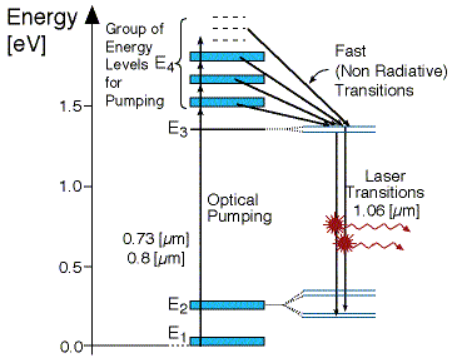
A photon of 1064 nm triggers an excited Nd^{3+} ion to release another photon of the same energy *(same wavelength)*, phase *(coherent)* and direction *(collimated)*.

1.3.5 Optical Resonator

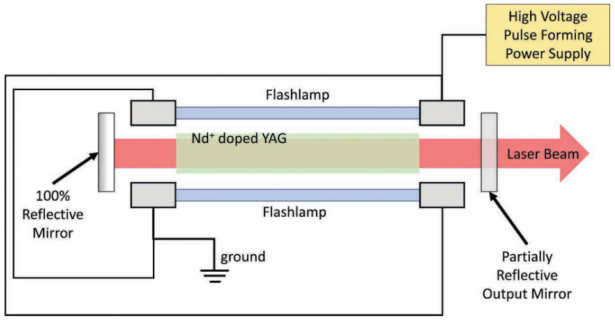
Two mirrors are placed at both ends of the Nd:YAG crystal: one fully reflective, another partially reflective (output coupler). Photons bounce back and forth, stimulating more emission and thus amplifying light.

1.3.6 Output

A highly intense, coherent, monochromatic beam at 1064 nm is emitted through the partially reflective mirror.



(a) Energy level diagram*



(b) Schematic diagram†

*Rami Arieli: "The Laser Adventure" Section 6.2.2 page 2

†What is a Nd:YAG Laser? - SZLASER

1.4 Basics of LIBS

Laser-induced breakdown spectroscopy (LIBS) is a type of atomic emission spectroscopy which uses a highly energetic laser pulse as its excitation source. A high-power, short-pulsed laser is used to ablate a small amount of material from the sample to form a plasma, which atomizes and excites samples. The plasma emits light and by analysing this light with a spectrometer, we can determine various properties of the material (like composition) as well as that of the plasma. The formation of the plasma only begins when the focused laser achieves a certain threshold for optical breakdown, which generally depends on the environment and the target material[10–12].

1.4.1 How LIBS works

1. **Laser Ablation:** A focused, high-energy laser pulse (commonly Nd:YAG at 1064 nm) hits the sample. The intense energy rapidly heats, melts, and vaporizes a micro-volume of the material.
2. **Plasma Formation:** The vaporized material becomes ionized, forming a hot plasma ($T_e \sim 1 - 2$keV) containing electrons, ions, and neutrals.
3. **Plasma Emission:** As the plasma cools and relaxes, atoms and ions emit characteristic photons (emission lines). These emission lines are unique "fingerprints" of the elements present.
4. **Spectral Detection:** A spectrometer collects the emitted light and disperses it into its wavelengths. A detector (CCD/ICCD) records the spectrum.
5. **Analysis:** The spectrum is compared with known databases (e.g., NIST atomic spectra database). From the line intensities and ratios, elemental identification (qualitative analysis), concentration estimation (quantitative analysis, e.g., calibration-free LIBS) and plasma parameters (e.g., T_e, n_e) can be be extracted.

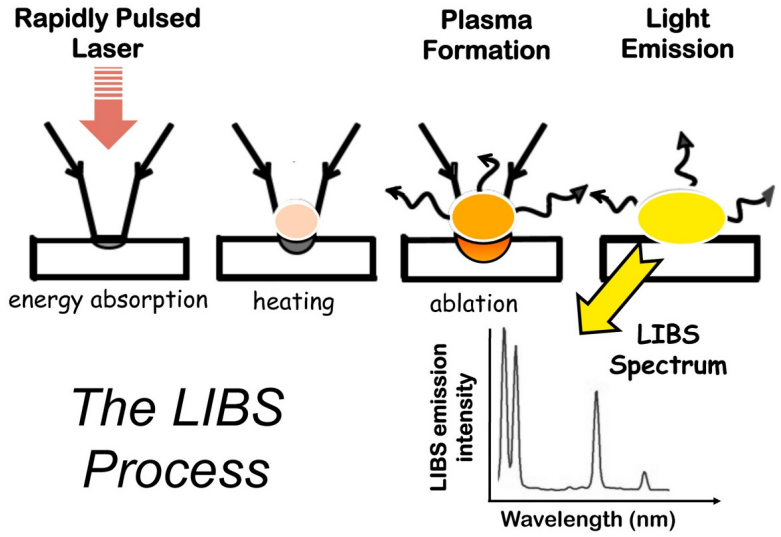


Figure 1.3: Steps in LIBS *

*Laser-Induced Breakdown Spectroscopy – A geochemical tool for the 21st century

1.4.2 Key Plasma Parameters in LIBS

Plasma parameters are essential for calibration-free LIBS (CF-LIBS) because they provide correct plasma conditions and turn intensities into concentrations[13].

1. **Electron Temperature (T_e):** It gives the average kinetic energy of electrons in plasma governing the excitation of electrons. In LIBS plasma, it ranges around $T_e \sim 1 - 2eV(10,000 - 20,000K)$

   ***Calculation**: Boltzmann Plot Method*

   The T_e can be measured based on the intensity of emission lines from the same species (all neutral or all singly ionized) that arise from different upper energy levels by using the Boltzmann plot. The equation is given by[14]:

   $$\ln\left(\frac{I_\lambda^{ki}\,\lambda_{ki}}{g_k\,A_{ki}}\right) = -\frac{E_k}{k_B\,T_e} + \ln\left(\frac{C_s\,F}{U_s(T)}\right) \tag{1.1}$$

   where:

   - I_λ^{ki} – peak line intensity measured at λ
   - λ_{ki} – wavelength of transition
   - g_k – k-level (upper level) degeneracy
   - A_{ki} – transition probability for the given line
   - E_k – upper energy level
   - k_B – is the Boltzmann constant
   - C_s – is the concentration of the emitting atomic species
   - F – experimental parameter that remains constant during the experiment
   - $U_s(T)$ – is the partition function for the emitting species at T_e

   Plotting the left-hand side on y-axis versus E_k on x-axis gives a straight line whose slope $= \dfrac{-1}{k_B T_e}$.

2. **Electron Number Density (n_e):** It gives the number of free electrons per unit volume in the plasma (cm^{-3}) and controls plasma conductivity, opacity, and line broadening. In LIBS plasma, it typical range around $n_e \sim 10^{16} - 10^{18}$ cm^{-3}.

   ***Calculation**: Saha Equation*

   The Saha ionization equation relates the ionization state of a gas in thermal equilibrium to the temperature and pressure. It is given by[15]:

   $$\frac{n_e n_{M_0}(\mathrm{II})}{n_{M_0}(\mathrm{I})} = 6.0 \times 10^{21} \frac{g_0^{\mathrm{II}} T^{3/2}}{g_0^{\mathrm{I}}} \exp\left(-\frac{E_{M_0}}{T}\right)$$

   where

   - n_e – is the electronic density ($\mathrm{e/cm}^3$)
   - n_{M_0}**(I) and** n_{M_0}**(II)** – are, respectively, the population of the ground state of the neutral atomic species M(I) and that of the single ionized species M(II)
   - g – values are the corresponding degeneration values

- T – is the plasma temperature as obtained by the Boltzmann plots
- E_{M_0} – is the ionization potential (in eV) of the species M(I) in its ground state.

Plasma parameter n_e can be obtained by rearranging this equation for an element whose concentration is known for both the ionization states:

$$n_e = 6.0 \times 10^{21} \frac{g_0^{\mathrm{II}} T^{3/2}}{g_0^{\mathrm{I}}} \exp\left(-\frac{E_{M_0}}{T}\right) \frac{n_{M_0}(\mathrm{I})}{n_{M_0}(\mathrm{II})} \tag{1.2}$$

***Calculation**: Stark Broadening Method*

Stark broadening is the broadening of spectral lines due to the electric fields produced by nearby charged particles – mainly electrons and ions – in a plasma. When an emission spectrum is recorded an isolated spectral line should be chosen and then the line profile is fitted[16]. Thus, the obtained Full Width at Half Maximum (**FWHM**)) is also called the Stark width, $\Delta\lambda_{\mathrm{Stark}}$

$$\Delta\lambda_{\mathrm{Stark}} = 2\omega \left(\frac{n_e}{10^{16}\ \mathrm{cm}^{-3}}\right)$$

where

- $\Delta\lambda_{\mathbf{Stark}}$ – measured line width (FWHM)
- ω – Stark width at $n_e = 10^{16}$ cm^{-3}, specific to the line and temperature.
- n_e – electron number density (cm^{-3})

Rearrange to get n_e:

$$n_e = \frac{\Delta\lambda_{\mathrm{Stark}}}{2\omega} \times 10^{16}\ \mathrm{cm}^{-3} \tag{1.3}$$

A hydrogen or neutral metal line with known Stark broadening data is usually used.

Importance of Plasma Parameters:

T_e and n_e is important for understanding the state of plasma and ensures that the plasma is close to local thermodynamic equilibrium (LTE) - a key assumption for Calibration–Free LIBS. They allow correction for:

- Self-absorption
- Matrix effects
- Line intensity variations

Elemental concentration can be calculated by comparing line intensities with atomic data from National Institute of Standards and Technology (NIST) after finding n_e and T_e values[17].

1.4.3 Light Emission Mechanisms

1. **Electron-Ion Recombination:** A photon is released when a free electron recombines with a positive ion:

$$\text{Ion}^+ + e^- \rightarrow \text{Atom}^* \rightarrow \text{Atom} + h\nu$$

Each element has a distinct wavelength ($\lambda = \frac{c}{\nu}$) that is correlated with the energy of the photon that is released.

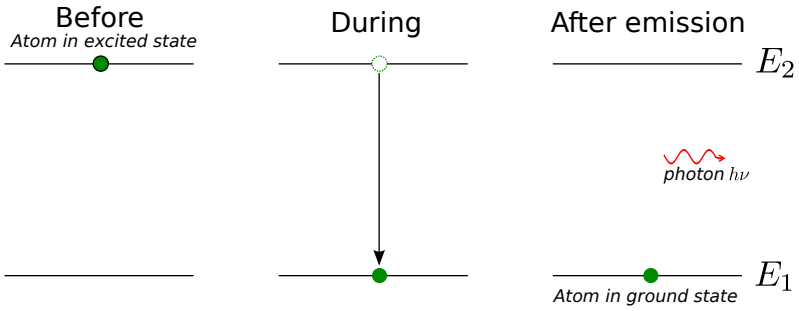


Figure 1.4: Steps in Line Emission [*]

2. **Radiative De-excitation:** Photons can be released when excited atoms or ions spontaneously fall to lower energy levels:

$$\text{Atom}^* \rightarrow \text{Atom} + h\nu$$

Each emitted photon has energy

$$E = h\nu = \frac{hc}{\lambda}$$

As a result, the spectrum exhibits a sharp emission line at wavelength λ.

3. **Bremsstrahlung (Continuum Emission):** In the early plasma life ($\sim$ 0–200 ns), the hot electrons scatter off ions and emit broadband radiation[18]. This is a continuum background and is not element-specific.

[*] Spontaneous Emission from Wikimedia Commons

Chapter 2

Methodology

2.1 NIST Atomic Spectral Database

The NIST Atomic Spectra Database (ASD), a well-known and reliable source for atomic parameters was used to access all of the spectral and atomic data needed for plasma diagnostics in this study. The database provides critically assessed values for spectral lines, transition probabilities, energy levels, and associated constants.

The ASD interface for Laser-Induced Breakdown Spectroscopy (LIBS) provides a convenient and straightforward access to the ASD data of relevance to LIBS applications, allowing the users to compare these synthetic plots with experimental spectra.[19]
It was used to access Line Intensity vs Wavelength data for Aluminium (Al) in `.csv` format with the following parameters:

- Wavelength range: 250 – 700nm
- Min. Rel. Intensity = 0.01
- $T_e = 1.0$ eV
- $n_e = 1.0\text{e+}17$ cm^{-3}
- Resolution = 11000

The generated spectral dataset and subsequent diagnostics only included Al I and Al II species due to two main reasons (1) Under the selected plasma conditions, the Saha–Boltzmann calculations indicate that the population of doubly ionized aluminium (Al III) is very small, and (2) Limiting the analysis to Al I and Al II maintains accuracy for the line-intensity modelling and ionization balance while streamlining the diagnostics.

Further the ASD database for Lines provides access to transition data for atoms and atomic ions. Output is available for wavelengths, relative intensities, radiative transition probabilities and related quantities.[20]
The following parameters were specifically obtained from the database:

1. **Observed Wavelengths (λ):** Spectral line positions in nanometres are used to locate distinctive Al I and Al II transitions in the LIBS spectra.
2. **Probabilities of transition (A_{ki}):** Einstein coefficients for spontaneous emission are crucial for relating excited state populations to measured line intensities.
3. **Upper energy levels (E_k) as well as their statistical weights (g_k):** These are important for using the Saha-Boltzmann equation and creating Boltzmann plots.

2.2 Computational approach

To evaluate the plasma parameters from the aluminium LIBS spectra, a Python program was developed. The algorithm processes spectral data obtained from the NIST Atomic Spectra Database and extracts the electron temperature (T_e) and electron number density (n_e) using established spectroscopic methods. The workflow consists of the following steps[21]:

1. **Data Import and Preprocessing:** LIBS spectral data (wavelength vs. intensity) for aluminium were extracted from `.csv` files. Additionally, distinct files with atomic line information (wavelengths, energy levels, and transition probabilities) for Al I and Al II were imported. The data were cleaned, formatted, and missing entries handled to ensure reliable matching with observed spectral peaks.

2. **Peak Identification and Line Integration:** The `scipy.signal` library was used to implement numerical peak-finding algorithms in order to locate the spectral peaks. After identifying the local minima on either side of each peak, `scipy.integrate` was used to compute the integrated line area using Simpson's rule. This resulted in a collection of peak intensities linked to their respective wavelengths.

3. **Line Assignment using NIST Data:** Detected peaks in the LIBS spectrum were matched with the closest tabulated transitions from the NIST database, within a defined tolerance. For each matched line, the associated transition probability (A_{ki}), statistical weight (g_k), and upper-level energy (E_k) were recorded.

4. **Boltzmann Plot for Electron Temperature:** Using the matched line intensities, the quantity $\ln\left(\frac{I}{gA}\right)$ was computed for each transition and plotted against the upper-level energy E_k. As given by Equation (1.1), a linear regression was performed, with the slope corresponding to $\frac{-1}{k_B T_e}$. This yielded temperature estimates for Al I and Al II, from which the average electron temperature was calculated.

5. **Electron Density from Saha–Boltzmann Relation:** Equation (1.2), which is the Saha–Boltzmann equation, was used to combine pairs of neutral (Al I) and ionic (Al II) lines. The relative line intensities gave estimates of the electron number density based on the calculated electron temperature and the known aluminium ionization energy. To minimize statistical error, several line pairs were examined, and an average n_e and its standard deviation were found.

6. **Output and Visualization:** For T_e and n_e, the program generates tabulated results, Boltzmann plots with fitted lines, and spectral plots with identified peaks. Both quantitative comparison with anticipated plasma parameters and visual confirmation of line identification were made possible by these outputs.

Because of its extensive ecosystem of visualisation and numerical libraries, open-source nature, and widespread use in scientific computing, Python was chosen as the programming language for this project. The `NumPy`, `pandas`, and `SciPy` libraries, in particular, provide efficient numerical operations and data analysis tools, and `matplotlib` enables the clear graphic display of diagnostic plots and spectral data. Python 3.11 was used for all calculations because it ensures reproducibility and makes it easy to expand the developed code for additional work. The full code is included in Appendix A for reference.

Algorithm 1 Computation of Plasma Parameters from LIBS Data

Input: LIBS spectral data (wavelength vs. intensity), NIST line data (Al I, Al II)
Output: Electron temperature T_e, electron density n_e

procedure PREPROCESSING
    Load spectral data from CSV file
    Load NIST line data for Al I and Al II
    Clean missing or invalid entries
end procedure

procedure PEAKDETECTION
    Identify peaks using numerical peak-finding
    for each detected peak **do**
        Locate local minima on either side
        Compute integrated line area using Simpson's rule
    end for
end procedure

procedure LINEASSIGNMENT
    for each detected peak **do**
        Match wavelength with closest NIST transition (within tolerance)
        Record transition probability A_{ki}, energy E_k, degeneracy g_k
    end for
end procedure

procedure TEMPERATUREESTIMATION
    Compute $\ln(I/gA)$ for each assigned line
    Perform linear regression against E_u
    Determine slope $\Rightarrow -1/(k_B T_e)$
    Store electron temperature T_e
end procedure

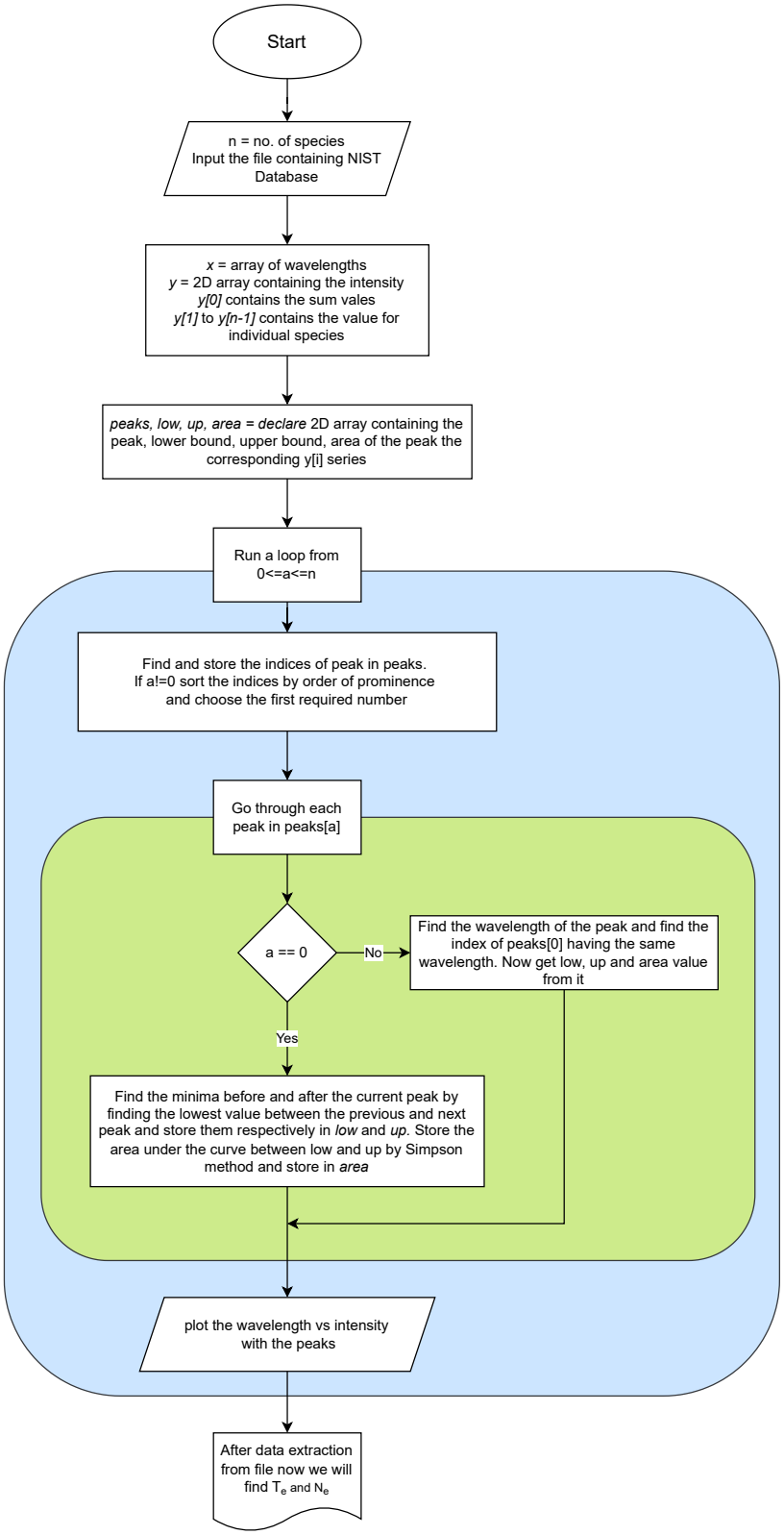
procedure DENSITYESTIMATION
    for pairs of Al I and Al II lines **do**
        Apply Saha–Boltzmann equation
        Estimate n_e from intensity ratios
    end for
    Compute average and standard deviation of n_e
end procedure

procedure VISUALIZATION
    Plot spectrum with identified peaks
    Plot Boltzmann diagram with fitted line
    Display computed T_e and n_e
end procedure

Return: Average T_e and n_e

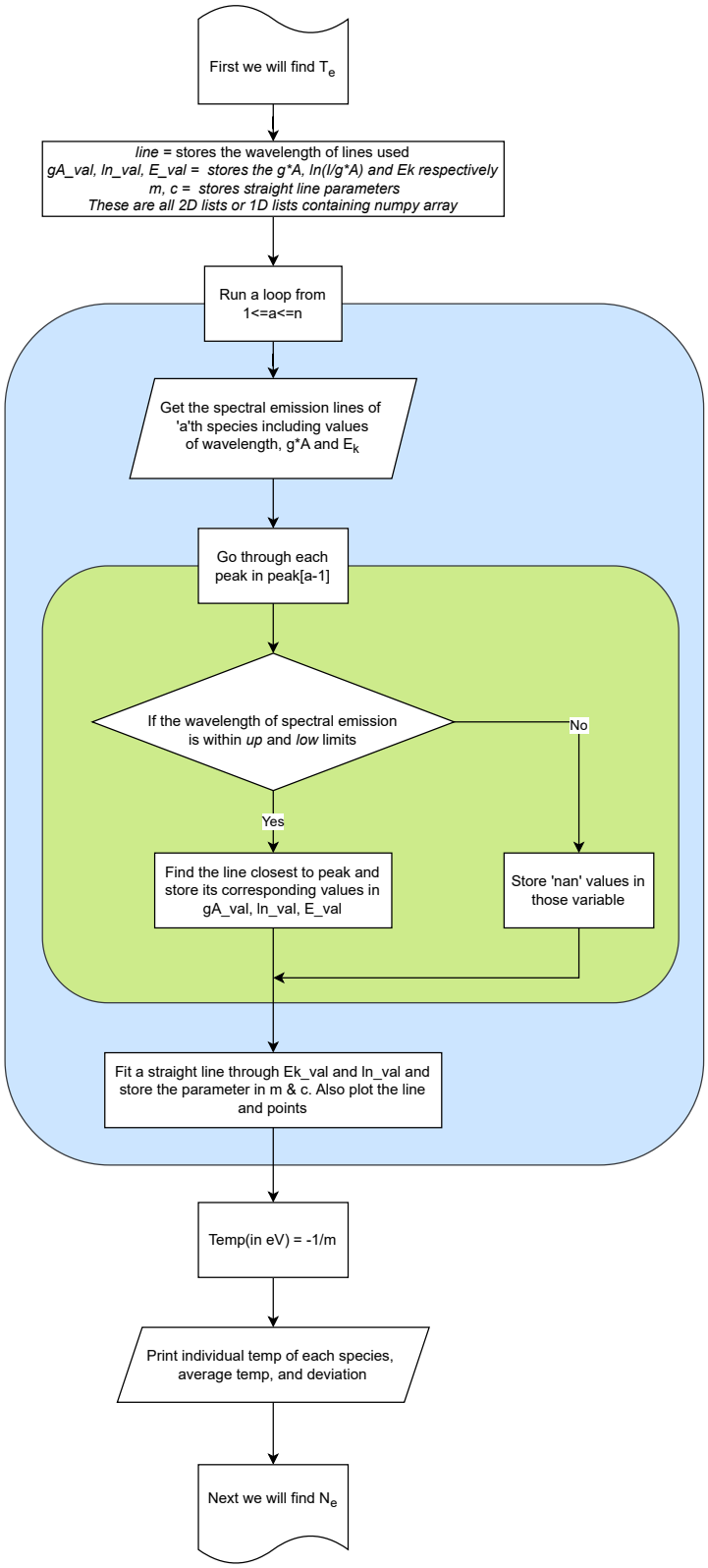
The above algorithm is described in detail in the following flowcharts.

2.2.1 Data Preprocessing



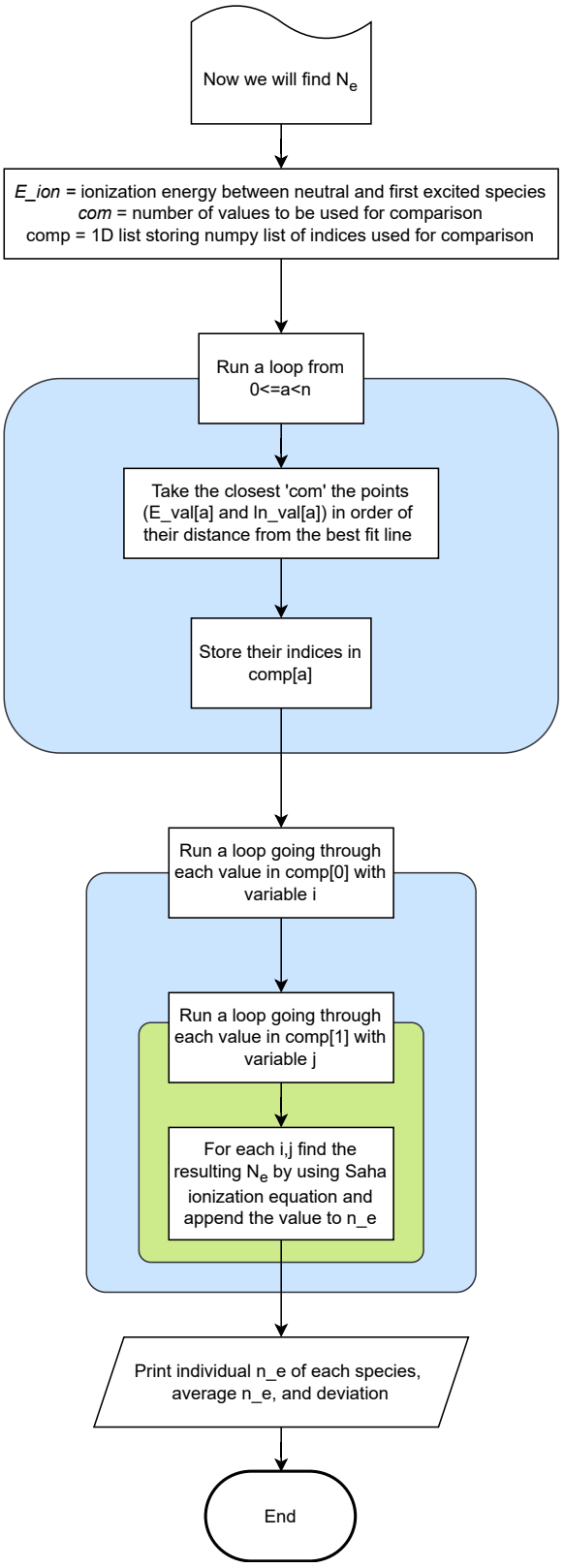
Flowchart 2.1: Detailed steps in data preprocessing: peak identification and line integration

2.2.2 Finding T_e using Boltzmann Plot



Flowchart 2.2: Detailed steps in line assignment and using Boltzmann plot

2.2.3 Finding n_e using Saha Equation

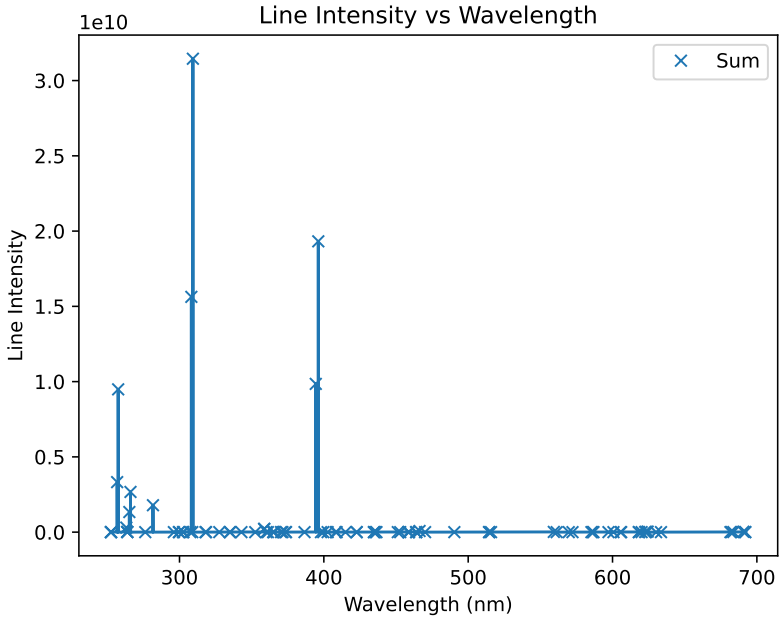


Flowchart 2.3: Detailed steps in finding electron number density

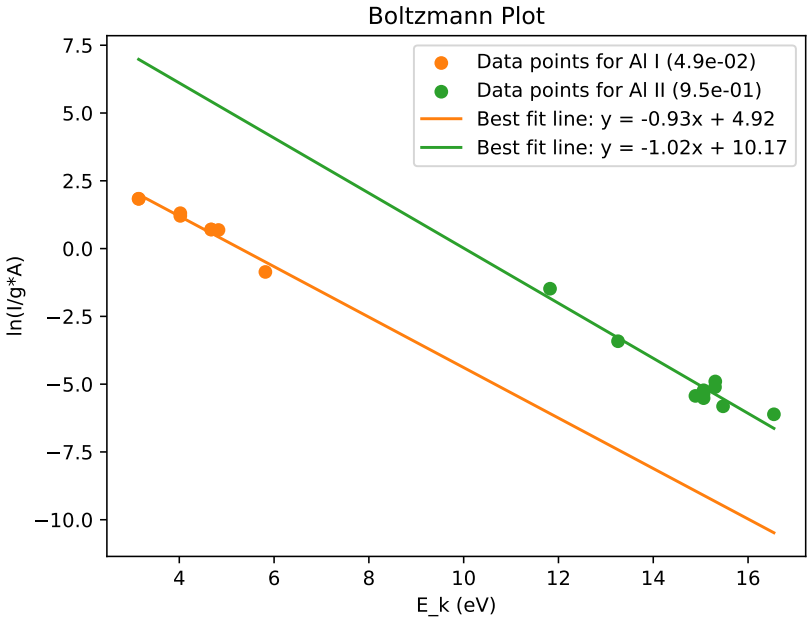
2.3 Testing and Validation

To verify the correctness of the developed algorithm, it was first applied to the LIBS spectrum of aluminium obtained from the NIST Atomic Spectra Database. The simulated plasma conditions were $T_e = 1$ eV and $n_e = 1 \times 10^{17}$ cm^{-3}. This step makes sure that the computational approach was able to reproduce these known plasma parameters.

Output:



Plot 2.1: Wavelength vs Line Intensity of Aluminium with Peaks



Plot 2.2: Boltzmann Plot for Al I and Al II

```
Temperatures (eV): 1.0743803598096184, 0.9847913656377654
Average Temperature = 1.029585862723692 eV
Standard Deviation = 0.04479449708592653 eV

Electron number density: 5.2078487248735315e+17, 3.64898223707846e+17,
5.6360740998101485e+17, 3.949026817702858e+17
Average N_e = 4.6104829698662496e+17
Deviation % = 18.05167983353625
```

Validation

The algorithm yielded an average electron temperature values of $T_e = 1.03 \pm 0.04$ eV. This result is in excellent agreement with the prescribed plasma temperature of 1.0 eV from the NIST dataset, confirming that the Boltzmann plot method was implemented correctly.

For the electron density, four independent estimates were obtained from pairs of Al I and Al II lines:

- 5.21×10^{17} cm^{-3}
- 3.65×10^{17} cm^{-3}
- 5.64×10^{17} cm^{-3}
- 3.95×10^{17} cm^{-3}

The average value was $n_e = 4.61 \times 10^{17}$ cm^{-3} with a deviation of approximately 18%. This estimate is of the correct order of magnitude even though it exceeds the nominal input density of 1.0×10^{17} cm^{-3} and the deviation within the values is also quite small. The sensitivity of the Saha–Boltzmann relation to slight changes in input parameters, the small number of spectral lines used, and uncertainties in line intensity integration causes these disparity.

Overall, the validation confirms that the developed code can accurately reproduce the plasma temperature and estimate the electron density within a reasonable range. These findings support the application of the computational method to actual experimental LIBS spectra in the following Chapter (Chapter 3) and show how reliable it is.

Chapter 3

Experimental Procedures and Application

While the computational approach was verified on aluminium spectra from the NIST database, the experimental LIBS measurements in this work has been carried out on copper samples. On the copper surface, plasma was created using a pulsed Nd:YAG laser. A spectrometer–detector system was used to gather and examine the radiation that was released. The experimental setup, data processing procedures used on the raw copper spectra, code modifications made to account for Cu I and Cu II transitions, and the resulting plasma parameters are covered in this chapter.

3.1 Experimental Setup

The experimental LIBS measurements were performed on copper samples using Nd:YAG laser with a fundamental wavelength of 1064 nm. The laser produced pulses with adjustable energies between 30 and 175 mJ per pulse, with a few Hz of repetition rate. A calibrated energy sensor was used to confirm the actual laser output energy before each run, and measurements were made at 30, 50, 75, 100, 125, 150, and 175 mJ to guarantee dependability. After the calibration verified steady output, the 100 mJ setting was chosen for the final analysis done here.

A lens was used to focus the laser beam onto a polished copper target, creating a power density high enough to ablate the material and produce a plasma plume. Prior to data collection, 25 “cleaning” laser shots were directed at each location to reduce the impact of surface contamination. Five consecutive shots were then taken, and the emission spectra were gathered from each one using a gate delay of 2000 ns. To increase the signal-to-noise ratio, the five spectra were averaged. For every energy level, the process was carried out five times on the same copper sample at different sites.

A lens was used to collect the emitted plasma radiation at a 90° angle to the laser axis, and it was then coupled into an optical fibre that was attached to a spectrometer. A Mechelle 22.8 system with a DH334T-18U-03 CCD detector served as the spectrometer. High wavelength measurement accuracy was ensured by calibrating the system prior to data acquisition, and only a very small deviation was discovered (calibration data in Appendix C). A gate delay of 2 µs, an exposure time of 10 µs, and a gate width of 4 µs was used for the current measurements. The experimental arrangement is illustrated schematically in Figure 3.2.

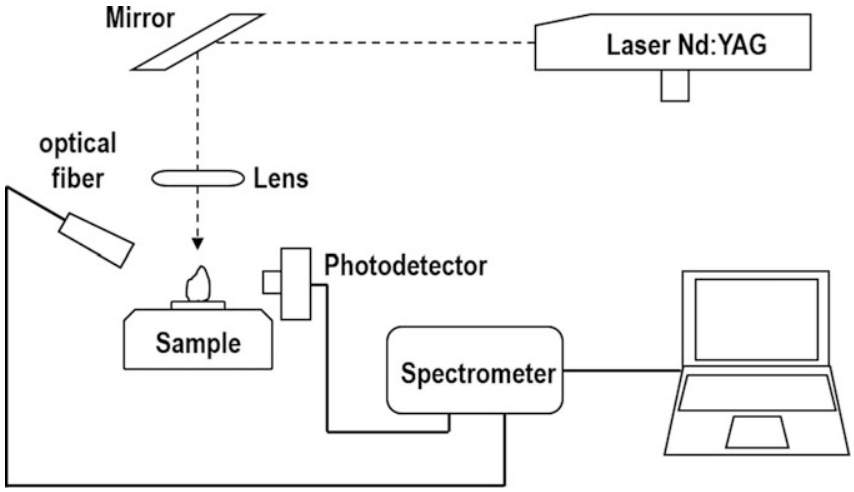


Figure 3.1: Schematic diagram of the experimental LIBS setup *

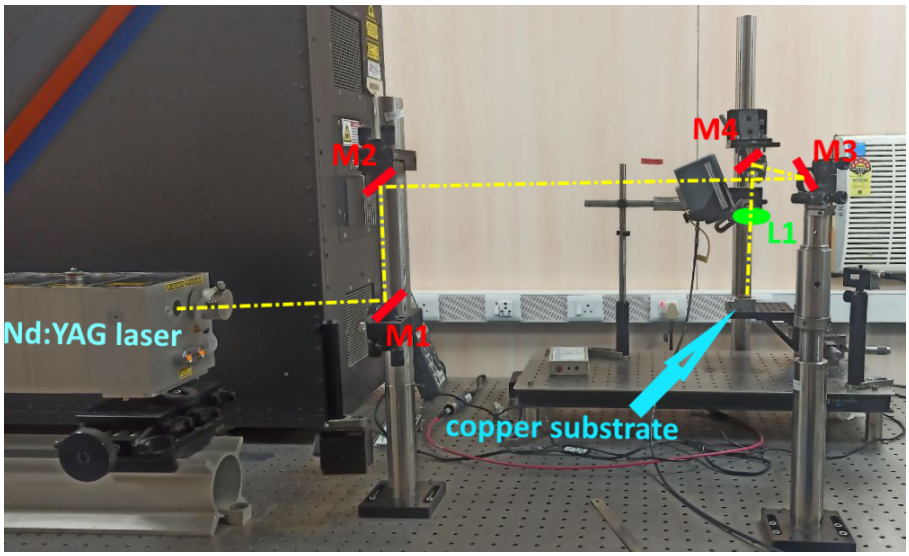


Figure 3.2: Optical path in setup for LIBS experimental setup.
M1–M4: Mirrors; L1: Focussing lens

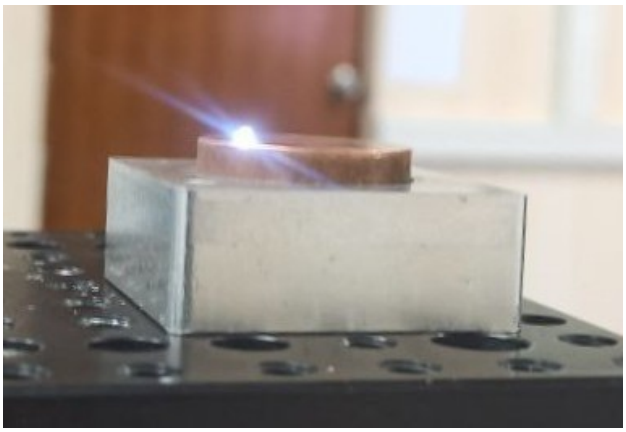


Figure 3.3: Plasma formation on copper sample

*The Technique of LIBS for Determination of Heavy Metals in the Receiving Body of Water

3.2 Data Analysis and Plasma Diagnostics

There were two main diagnostic procedures carried out:

1. **Electron temperature** (T_e) via the Boltzmann plot method, and
2. **Electron number density** (n_e) using both the Saha–Boltzmann relation and Stark broadening analysis.

Algorithm 2 Plasma Diagnostics Workflow for Copper LIBS Data

Input: Five Cu LIBS spectra ($S_1, S_2, \ldots, S_5$)
Output: Electron temperature (T_e) and electron density (n_e)

procedure Data Averaging
    Load spectra $S_1, \ldots, S_5$
    Average to obtain mean spectrum S_{avg}
end procedure

procedure Peak Detection and Line Assignment
    Detect peaks in S_{avg}
    Match observed peaks with NIST Cu I and Cu II transitions
    Extract line parameters: I, g_k, A_{ki}, E_k
end procedure

procedure Temperature Extraction
    for each transition **do**
        Compute $\ln(I/g_k A_{ki})$
    end for
    Fit $\ln(I/g_k A_{ki})$ vs E_k with linear regression
    Extract slope m
    Compute $T_e = -1/m$
end procedure

procedure Electron Density (Saha–Boltzmann)
    Select pairs of Cu I and Cu II transitions
    for each matched pair **do**
        Compute n_e using

$$n_e \propto \frac{I^{\text{Cu II}}/(g_k A_{ki})}{I^{\text{Cu I}}/(g_k A_{ki})} \exp\left(\frac{E_k^{\text{Cu I}} - E_k^{\text{Cu II}} - E_{ion}}{T_e}\right) T_e^{3/2}$$

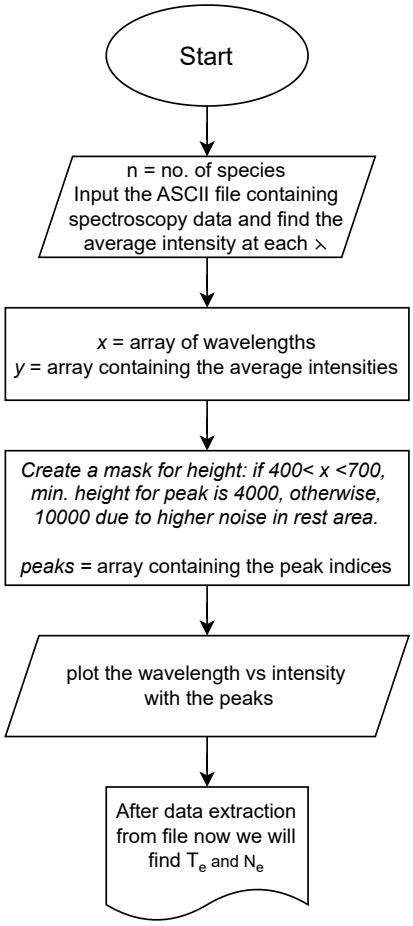
    end for
    Average results to obtain $n_e^{(Saha)}$
end procedure

procedure Electron Density (Stark Broadening)
    Isolate Hα line region (652.5–662.5 nm)
    Fit line profile with Lorentzian profile
    Extract Lorentzian width Γ
    Convert Γ to $n_e^{(Stark)}$ using tabulated coefficients
end procedure

Return: T_e, $n_e^{(Saha)}$, $n_e^{(Stark)}$

3.2.1 Data Preprocessing and Peak Detection

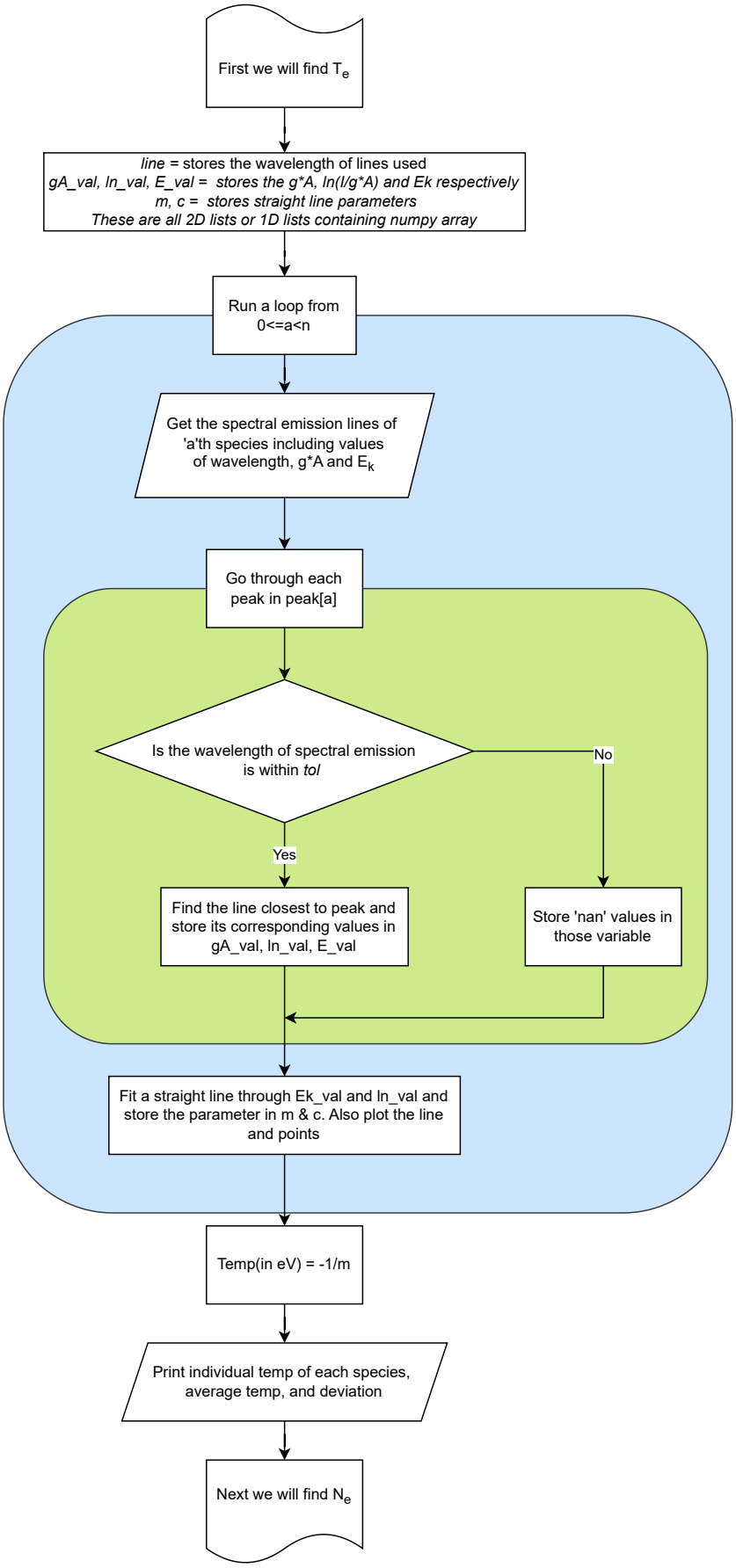
The five recorded spectra (`CuLIBS1.asc` to `CuLIBS5.asc`) were imported and combined by taking the mean intensity at each wavelength. Peaks were identified in the averaged spectrum using the `scipy.signal.find_peaks` routine. To ensure robust detection, a variable threshold was applied: peaks in the 400–700 nm range were detected above a lower threshold, while a higher cut-off was applied outside this range to suppress spurious noise. The resulting peak list provided the observed emission lines of the copper plasma.



Flowchart 3.1: Detailed steps in data preprocessing and peak detection

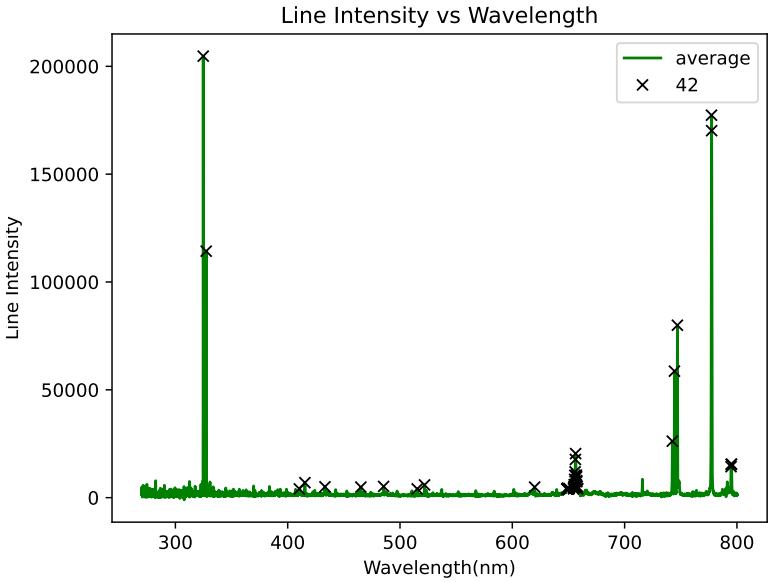
3.2.2 Line Assignment and Boltzmann Plot for T_e

The NIST Atomic Spectra Database's tabulated Cu I and Cu II transitions were compared to the observed copper peaks. The excitation energy (E_k), statistical weight (g_k), and transition probability (A_{ki}) were noted for every match. The code computed $\ln\left(\frac{I\lambda}{g_k A_{ki}}\right)$ after using Equation (1.1) to find the electron temperature, where I is the measured peak line intensity at wavelength λ. The excitation energy E_k (in eV) was plotted against these values. $T_e = -\frac{1}{m}$ was derived from the slope m of the Boltzmann plot, which was obtained through linear regression. The standard deviation of the results gave an estimate of the uncertainty, and separate fits for Cu I and Cu II were averaged. In this case, the flowchart is quite similar to Flowchart 2.2.



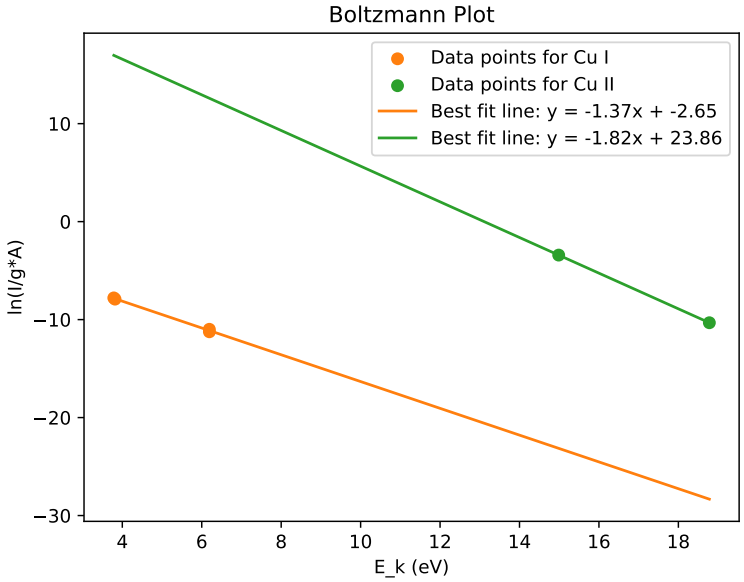
Flowchart 3.2: Detailed steps in line assignment and using Boltzmann plot

Output:



Plot 3.1: Wavelength vs Intensity of Copper with Peaks

```
Cu I  line; Observed: 324.77386 Actual: 324.7536 gA: 558000000.0 Ek 30783.697
Cu I  line; Observed: 327.43427 Actual: 327.3952 gA: 275200000.0 Ek 30535.324
Cu I  line; Observed: 515.33295 Actual: 515.3238 gA: 240000000.0 Ek 49935.195
Cu I  line; Observed: 521.82684 Actual: 521.8202 gA: 450000000.0 Ek 49942.051
Cu II line; Observed: 657.69287 Actual: 657.70816 gA: 160000000.0 Ek 151470.113
Cu II line; Observed: 777.35187 Actual: 777.31992 gA: 5400000.0 Ek 120876.0141
```

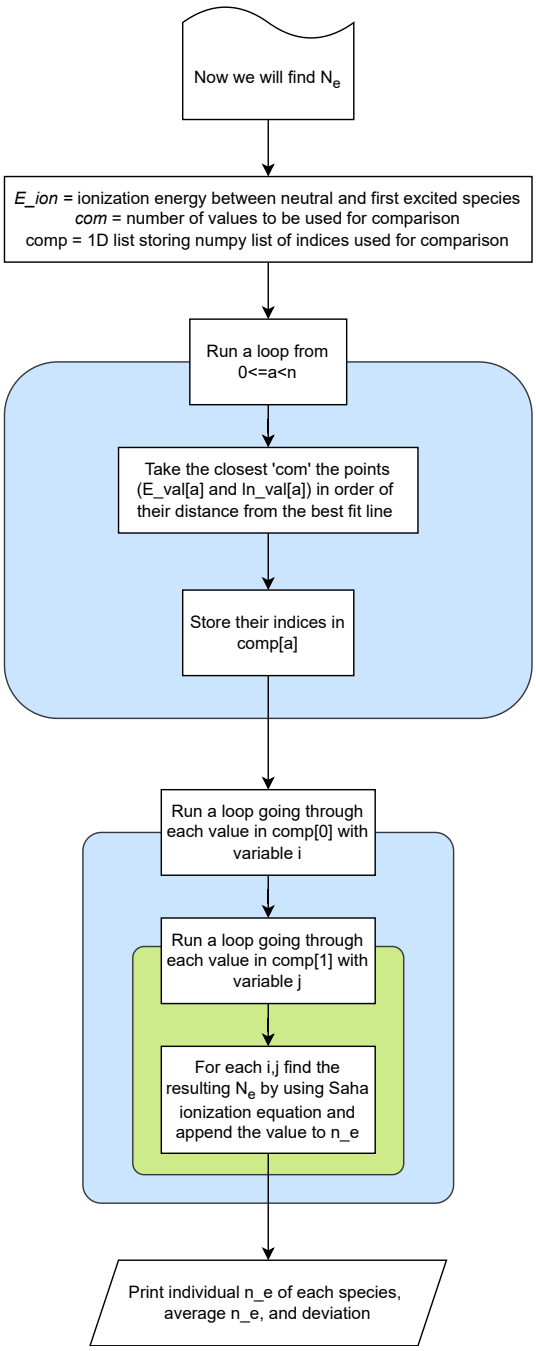


Plot 3.2: Boltzmann Plot for Cu I and Cu II

```
Temperatures (eV): 0.7314209047901338, 0.549414442963459
Average Temperature = 0.6404176738767964 eV
Standard Deviation = 0.0910032309133374 eV
```

3.2.3 Calculating (n_e) from Saha Ionisation Equation

The Saha ionisation equation was first used to estimate the electron number density using the ratio of neutral to singly ionised line intensities, T_e, the ionisation energy and the partition functions of the two ionisation stages to estimate the electron number density. Closely matched Cu I and Cu II transitions were chosen and the Equation (1.2) was used with their intensities and transition probabilities. At the extracted temperature, partition functions $U(T)$ were obtained from the NIST database. This method yielded multiple n_e values, which were averaged to obtain a representative electron density along with its standard deviation. Here, the flowchart is quite similar to Flowchart 2.3.



Flowchart 3.3: Detailed steps in finding electron number density

Output:

```
Electron number density: 53629.15403759502, 35748851440.10834,
114095.80049609604, 76055531642.65619
Average N_e = 27951137701.929764
Deviation % = 112.24656778882145
```

3.2.4 Calculating (n_e) from Stark Broadening

The initial attempt to compute n_e from the Saha–Boltzmann relation gave highly inconsistent results, with values spanning several orders of magnitude (e.g., 5.36×10^4 to 7.61×10^{10} cm^{-3}). With a deviation of more than 100%, the average electron density was $\langle n_e \rangle = 2.80 \times 10^{10}$ cm^{-3}. This significant scatter shows how susceptible the method is to line misidentification; specifically, a peak close to 777 nm was mistakenly attributed to Cu II when it actually corresponds to an oxygen line. The NIST LIBS database at the approximate plasma conditions can be used to confirm this.

As a result, only Cu I transitions were used to determine the electron temperature, whereas Stark broadening of the hydrogen Hα line at 656.3nm was used to determine the electron density[22]. The experimental profile of this line was isolated in the range 652.5–662.5 nm and fitted with a Lorentzian function

$$L(\lambda) = I_0 \frac{(\Gamma/2)^2}{(\lambda - \lambda_0)^2 + (\Gamma/2)^2},$$

where

- I_0 is the peak intensity
- λ_0 is the line centre
- Γ is the full width at half maximum (FWHM).

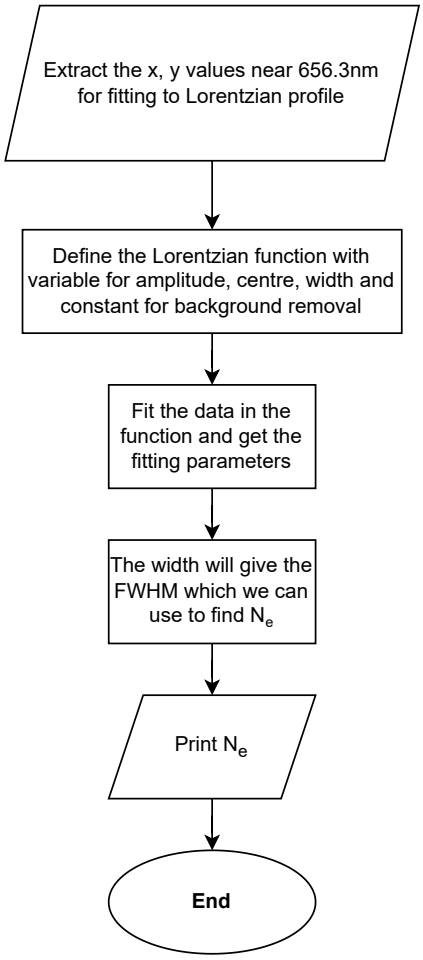
The extracted FWHM is directly related to the electron density through the Stark broadening relation given in Equation (1.3):

$$n_e = \frac{\Delta\lambda_{\mathrm{FWHM}}}{2\omega} \times 10^{16} \text{ cm}^{-3}$$

where ω is the tabulated Stark broadening parameter for Hα at $n_e = 10^{16}$ cm^{-3}.

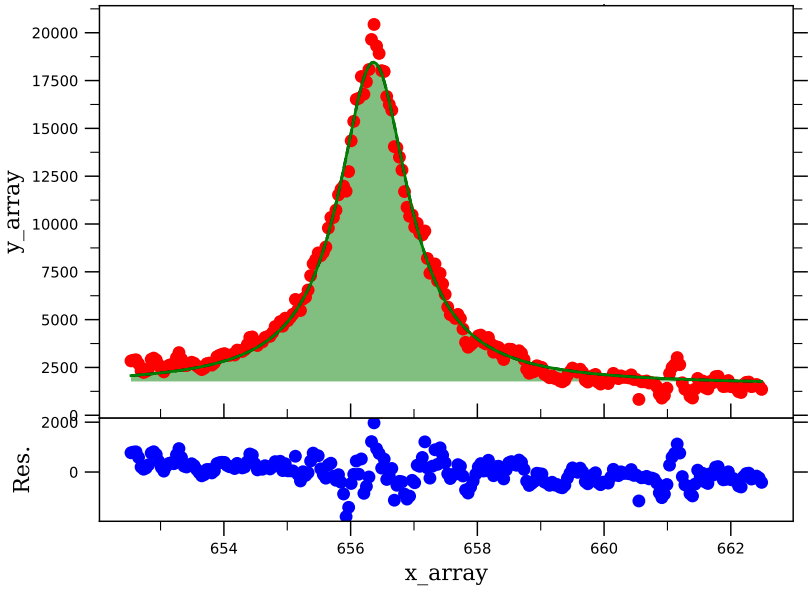
This method avoids ambiguities due to line misidentification and yields a more reliable estimate of n_e. Non-linear least squares (`scipy.optimize.curve_fit`) was used for the fitting. The extracted Lorentzian width Γ was related to n_e through the Stark broadening coefficient reported in the literature. A residual analysis was performed to verify the fit quality, and the coefficient of determination (R^2) was calculated to ensure robustness.

The fitting routine was adapted with reference to publicly available examples by Emily Ripka on her blog[23].



Flowchart 3.4: Detailed steps in finding electron number density

Output:



Plot 3.3: Wavelength vs Intensity and residual plot for Hα line

```
-------------Peak-------------
constant = 1567.36 (+/-) 47.68
amplitude = 16897.60 (+/-) 135.28
center = 656.36 (+/-) 0.01
width = 0.67 (+/-) 0.01
area = 1196455.09
R^2 =  0.9882411555103434
N_e =  4.526728033734376e+16
```

Results

In this chapter, the experimental procedure for obtaining LIBS spectra from a copper target was described, including laser operation, spectrometer calibration, and data acquisition parameters. The raw spectra were averaged, peaks were detected, and spectral lines were assigned using the NIST database. The electron temperature T_e was determined from Cu I transitions via the Boltzmann plot method, while the electron density n_e was extracted from Stark broadening of the Hα line.

The Saha–Boltzmann approach for n_e was attempted but yielded inconsistent results due to line misidentification, highlighting the importance of careful spectral validation. The use of Stark broadening provided a more reliable estimate of n_e, while Cu I transitions alone were found to give consistent T_e.

$$T_e = 0.7314209047901338 \text{ eV}$$
$$n_e = 4.526728033734376 \times 10^{16} \text{ cm}^{-3}$$

Together, these results demonstrate the feasibility of computational plasma diagnostics on real experimental data. The next chapter will present the overall conclusions, key findings, and possible directions for future work.

Chapter 4

Conclusions and Future Work

4.1 Summary of Work

This project's objective was to use both computational and experimental methods to assess plasma parameters from a typical LIBS plasma. The study started by going over the basic ideas of laser operation (Nd:YAG), plasma physics, and Laser-Induced Breakdown Spectroscopy. A Python-based program was created to carry out calibration-free diagnostics of plasma parameters using the NIST Atomic Spectra Database. The technique was applied to experimental spectra of copper acquired using a pulsed Nd:YAG laser after being verified on NIST tabulated data for Al I and Al II.

4.2 Key Findings

- The Boltzmann plot method successfully provided electron temperature (T_e) values from Cu I transitions, with consistent results between different lines.
- The Saha–Boltzmann approach for electron density (n_e) gave unreliable results due to spectral line misidentification (notably, an O line near 777 nm was incorrectly treated as Cu II). This demonstrated the sensitivity of the method to line assignment and the need for rigorous line validation.
- Stark broadening analysis of the Hα line provided a more robust method for determining n_e. The Lorentzian fit yielded FWHM values that could be directly related to n_e using tabulated Stark parameters.
- The developed computational framework is flexible and can be adapted to both database-derived and experimental LIBS spectra, providing a foundation for calibration-free plasma diagnostics.

4.3 Limitations

- The analysis assumed local thermodynamic equilibrium (LTE), which may not hold at all plasma conditions.
- Only Cu I lines were used for temperature extraction, as Cu II lines were prone to misidentification and interference.
- Self-absorption, matrix effects, and instrumental broadening were not explicitly accounted for, which may affect the accuracy of extracted parameters.

4.4 Future Scope

- Extend the code to include multi-element plasmas, enabling diagnostics in more complex samples.
- Extend the code to fit Gaussian for every peak to determine plasma parameters.
- Incorporate corrections for self-absorption and instrumental broadening to improve accuracy.
- Explore the use of Al III and Cu II lines once experimental conditions and databases allow for reliable identification.
- Integrate machine learning approaches for faster line recognition and automated plasma parameter extraction.
- Perform systematic studies of the effect of laser energy, gate delay, and ambient atmosphere on extracted plasma parameters.

4.5 Concluding Remarks

This project demonstrated the successful implementation of a calibration-free LIBS diagnostic framework that integrates both theoretical plasma models and experimental data. By combining Boltzmann analysis for T_e and Stark broadening for n_e, a practical and reliable approach to plasma diagnostics was established. While limitations remain, the methodology provides a solid basis for further refinement and application in real-world LIBS studies.

Appendix A

Python Code for Aluminium LIBS Plasma Diagnostics

The source code is divided into three main parts:

1. Data Preprocessing – Peak Identification and Line Integration
2. Finding T_e using Boltzmann Plot
3. Finding n_e using Saha Equation

A.1 Data Preprocessing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sps
import scipy.integrate as spi
import copy

n = 2 #No. of species
use = [10,10]

# Data loading from file
file_path = 'C:/Users/a/Documents/PRL/Code/Al (T=1.0).csv'
data = pd.read_csv(file_path, comment='#').fillna(value = 0.0)

# Feature selection
x_ = data.columns[0]  # 0: Wavelength
x = data[x_].values

y_ = data.columns[1:]  # 1: sum, 2: Al 1, 3: Al 2, 4: Al 3

y = data[y_].values
y = np.transpose(y)

#Find peaks and area
peaks = []
area = [[] for i in range(n+1)]
low, up = copy.deepcopy(area), copy.deepcopy(area)

```

```
for a in range(n+1):
    peak, properties = sps.find_peaks(y[a], height=0)

    if (a != 0):
        prop = sps.peak_prominences(y[a], peak)
        indices = np.argsort(prop[0])[-1:-1-use[a-1]:-1]
        peak = peak[indices]

    peaks.append(peak)

    for i, p in enumerate(peak):

        if (a != 0):
            ind = np.where(peaks[0] == p)[0][0]
            area[a].append(area[0][ind])
            low[a].append(low[0][ind])
            up[a].append(up[0][ind])
            continue

        l = 0 if (i == 0) else peak[i-1]
        u = y.shape[1]  if (i == np.size(peak) - 1) else peak[i+1]
        m1, m2 = p, p

        #finding area uptil minima before peak
        for j in range(l,p):
            if (y[a,j] < y[a,m1]):
                m1 = j

        #finding area uptil minima after peak
        for j in range(p,u):
            if (y[a,j] < y[a,m2]):
                m2 = j

        area[a].append(spi.simpson(y[a,m1:m2+1], x=x[m1:m2+1]))
        low[a].append(x[m1])
        up[a].append(x[m2])

    #ploting with peak
    plt.plot(x, y[a], c='C'+str(a))
    plt.plot(x[peak], y[a][peak],"x", label=y_[a],c='C'+str(a))
    plt.title('Line Intensity vs Wavelength')
    plt.xlabel(x_)
    plt.ylabel('Line Intensity')
    plt.legend()
    if a == 0:
    plt.savefig("Al_Intensity.pdf", format='pdf')
    plt.show()
```

Listing A.1: Source code for data preprocessing

A.2 Finding T_e using Boltzmann Plot

```
nan = np.nan
#Find corresponding lines
line = [[] for _ in range(n)]
gA_val, ln_val, E_val, m, c = [],[],[], np.array([]), np.array([])

lim = np.array([1e4,-1e4])
tol = 0.05  #tolerance

for a in range(1,n+1):

    #reading data from file
    file_path = 'C:/Users/a/Documents/PRL/Code/Al ' +str(a)+ ' (lines).csv'
    data = pd.read_csv(file_path)
    columns = data.columns

    #cleaning data
    data = data.replace(r'[^a-zA-Z0-9/*.\s]','', regex=True)
    data.replace("", nan, inplace=True)
    data = data.dropna(subset=[columns[3]])
    data = data[[data.columns[0], data.columns[3], data.columns[6]]]
    data = data.apply(pd.to_numeric)

    size = (data.shape[0])

    #Find Matching wavelengths
    gA, Ek = [], []
    for i, p in enumerate(peaks[a]):
        ind, diff = -1, abs(data.iloc[0, 0] - x[p])
        for j in range(size):
            if ((abs(data.iloc[j, 0] - x[p]) < diff) and (low[a][i] <= data.iloc[j, 0]) and (data.iloc[j, 0] <= up[a][i])):
                ind, diff = j, abs(data.iloc[j, 0] - x[p])

        if (ind != -1):
            line[a-1].append(data.iloc[ind, 0])
            gA.append(data.iloc[ind, 1])
            Ek.append(data.iloc[ind, 2])
        else:
            gA.append(nan)
            Ek.append(nan)

    gA = np.array(gA)
    ln = np.log(area[a]/gA)
    Ek = np.array(Ek)/8065.543937

    gA_val.append(gA)
    ln_val.append(ln)
    E_val.append(Ek)

    Ek = Ek[~np.isnan(Ek)]
    ln = ln[~np.isnan(ln)]
```

```
51
52      if (Ek.size != 0 and ln.size != 0):
53          coef = np.polyfit(Ek, ln, 1)
54          m = np.append(m, coef[0])  # Slope
55          c = np.append(c, coef[1])  # Intercept
56          lim = np.array([min(lim[0],Ek.min()),max(lim[1],Ek.max())])
57
58      else:
59          m = np.append(m, 0)  # Slope
60          c = np.append(c, 0)  # Intercept
61
62      plt.scatter(Ek, ln, c='C'+str(a), label='Data points for '+y_[a
    ])
63
64
65  # Plot the data and the best fit line
66  for a in range(n):
67      plt.plot(lim, lim*m[a] + c[a],  c='C'+str(a+1), label=f'Best
    fit line: y = {m[a]:.2f}x + {c[a]:.2f}')
68
69  plt.title('Boltzmann Plot')
70  plt.xlabel('E_k (eV)')
71  plt.ylabel('ln(I/g*A)')
72  plt.legend()
73  plt.savefig("Al_Boltzmann.pdf", format='pdf')
74  plt.show()
75
76  #Temperature
77  Temp = -1/m
78  T = np.average(Temp)
79
80  print("Temperatures (eV):", ', '.join(map(str, Temp)))
81  print("Average Temperature =", T, "eV")
82  print("Standard Deviation =", np.std(Temp), "eV")
```

Listing A.2: Source code for finding T_e

A.3 Finding n_e using Saha Equation

```
comp = []
com = 2
E_ion = 5.985769
U = []
#
for a in range(n):
    dis = np.abs(m[a] * E_val[a] - ln_val[a] + c[a])/np.sqrt(m[a]**2 + 1)
    indices = np.argsort(dis)[:com]
    mxm = dis.min()
    comp.append(indices)

n_e = []
for i in comp[0]:
    for j in comp[1]:
        n_e.append((gA_val[1][j] / gA_val[0][i]) * (y[1,peaks[1][i]] / y[2,peaks[2][j]]) * np.exp((E_val[0][i]-E_val[1][j]-E_ion)/T))


n_e = np.array(n_e) * 6.0371021691e21 * np.power(T,1.5)

print("Electron number density:", ', '.join(map(str, n_e)))
print("Average N_e =", np.average(n_e[:80]))
print("Deviation % =", np.std(n_e)/np.average(n_e)*100)
```

Listing A.3: Source code for finding n_e

Appendix B

Python Code for Copper LIBS Plasma Diagnostics

The source code is divided into three main parts:

1. Data Preprocessing – Peak Identification
2. Line Integration and Boltzmann Plot for T_e
3. Calculating (n_e) from Saha Ionisation Equation
4. Calculating (n_e) from Stark Broadening

B.1 Data Preprocessing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sps
import scipy.integrate as spi
import copy

n = 5 #No. of files

# Data loading from file
file_path = 'C:/Users/a/Documents/PRL/Data/Delay_2000ns/100
    mJ_GD2000ns/' #1.asc'

# Feature selection
# x_ = data.columns[0]  # 0: Wavelength
x_ = []
y_ = []

for a in range(1,n+1):

    data = pd.read_csv(file_path+'CuLIBS'+str(a)+'.asc', header=
    None, delimiter='\t', comment='#').fillna(value = 0.0)

    x_.append(data[0].values)
    y_.append(np.transpose(data[1].values))

x = x_[0]
y_ = np.array(y_)
```

```
y = y_.mean(axis=0)

h = np.where((x > 400) & (x < 700), 4000, 10000)

peak, properties = sps.find_peaks(y, height=h)

plt.plot(x, y, label="average",c='green')
plt.plot(x[peak], y[peak],"x", label=str(peak.size), c='black')
plt.title('Line Intensity vs Wavelength')
plt.xlabel('Wavelength(nm)')
plt.ylabel('Line Intensity')
plt.legend()
plt.show()
```

Listing B.1: Source code for data preprocessing

B.2 Line Integration and Boltzmann Plot for T_e

```
n=2 #no. of species

nan = np.nan
#Find corresponding lines
line = [[] for _ in range(n)]
gA_val, ln_val, E_val, m, c = [],[],[], np.array([]), np.array([])

lim = np.array([1e4,-1e4])
tol = 0.05  #tolerance

for a in range(n):

    #reading data from file
    file_path = 'C:/Users/a/Documents/PRL/Code/Cu ' +str(a+1)+ ' (lines).csv'
    data = pd.read_csv(file_path)
    columns = data.columns

    #cleaning data
    data = data.replace(r'[^a-zA-Z0-9/*.\s]','', regex=True)
    data.replace("", nan, inplace=True)
    data = data.dropna(subset=[columns[3]])
    data = data[[data.columns[1], data.columns[3], data.columns[6]]]
    data = data.apply(pd.to_numeric)

    size = (data.shape[0])

    #Find Matching wavelengths
    gA, Ek = [], []
    for i, p in enumerate(peak):
        ind, diff = -1, abs(data.iloc[0, 0] - x[p])
        for j in range(size):
            if ((abs(data.iloc[j, 0] - x[p]) < diff) and (abs(data.iloc[j, 0] - x[p]) < tol)):
                ind, diff = j, abs(data.iloc[j, 0] - x[p])
```

```
        if (ind != -1):
            line[a-1].append(data.iloc[ind, 0])
            gA.append(y[p]/data.iloc[ind, 1])
            Ek.append(data.iloc[ind, 2])
            print('Cu '+('I'*(a+1)) + (' '*(1-a))+ " line; Observed
   :",x[p],"Actual:",data.iloc[ind, 0],"gA:",data.iloc[ind, 1],"Ek"
   ,data.iloc[ind, 2])

    gA = np.array(gA)
    ln = np.log(gA)
    Ek = np.array(Ek)/8065.543937

    gA_val.append(gA)
    ln_val.append(ln)
    E_val.append(Ek)

    Ek = Ek[~np.isnan(Ek)]
    ln = ln[~np.isnan(ln)]

    if (Ek.size != 0 and ln.size != 0):
        coef = np.polyfit(Ek, ln, 1)
        m = np.append(m, coef[0])  # Slope
        c = np.append(c, coef[1])  # Intercept
        lim = np.array([min(lim[0],Ek.min()),max(lim[1],Ek.max())])

    else:
        m = np.append(m, 0)  # Slope
        c = np.append(c, 0)  # Intercept

    plt.scatter(Ek, ln, c='C'+str(a+1), label='Data points for Cu '
   +('I'*(a+1)))

# Plot the data and the best fit line
for a in range(n):
    plt.plot(lim, lim*m[a] + c[a],  c='C'+str(a+1), label=f'Best
   fit line: y = {m[a]:.2f}x + {c[a]:.2f}')

plt.title('Boltzmann Plot')
plt.xlabel('E_k (eV)')
plt.ylabel('ln(I/g*A)')
plt.legend()
plt.show()

#Temperature
Temp = -1/m
T = np.average(Temp)

print("Temperatures (eV):", ', '.join(map(str, Temp)))
print("Average Temperature =", T, "eV")
print("Starndard Deviation =", np.std(Temp), "eV")
```

Listing B.2: Source code for line integration and finding T_e

B.3 Calculating (n_e) from Saha Ionisation Equation

```
comp = []
com = 2
E_ion =  7.726380

for a in range(n):
    dis = np.abs(m[a] * E_val[a] - ln_val[a] + c[a])/np.sqrt(m[a
   ]**2 + 1)
    indices = np.argsort(dis)[:com]
    mxm = dis.min()
    comp.append(indices)

n_e = []
for i in comp[0]:
    for j in comp[1]:
        n_e.append((gA_val[1][j] / gA_val[0][i]) * (y[peak[i]] / y[
   peak[j]]) * np.exp((E_val[0][i]-E_val[1][j]-E_ion)/T))


n_e = np.array(n_e) * 6.0371021691e21 * np.power(T,1.5)

print("Electron number density:", ', '.join(map(str, n_e)))
print("Average N_e =", np.average(n_e))
print("Deviation % =", np.std(n_e)/np.average(n_e)*100)
```

Listing B.3: Source code for calculating (n_e) from Saha Ionisation Equation

B.4 Calculating (n_e) from Stark Broadening

The Stark broadening analysis routine for Lorentzian peak fitting was adapted from publicly available code examples. In particular, a Jupyter notebook hosted on GitHub[24]. The routines were customised for the copper LIBS analysis carried out in this project.

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import scipy as scipy
from scipy import optimize
from matplotlib.ticker import AutoMinorLocator
from matplotlib import gridspec
import matplotlib.ticker as ticker

const = 1000
ampL1 = 10000
cenL1 = 656.3
widL1 = 1

def _1Lorentzian(x, c, amp, cen, wid):
    return c+amp*wid**2/((x-cen)**2+wid**2)

x_array = x[np.where((652.5<x) & (x<662.5))]
y_array_lorentz = y[np.where((652.5<x) & (x<662.5))]
```

```
popt_lorentz, pcov_lorentz = scipy.optimize.curve_fit(_1Lorentzian,
    x_array, y_array_lorentz, p0=[const, ampL1, cenL1, widL1])

perr_lorentz = np.sqrt(np.diag(pcov_lorentz))

pars_L = popt_lorentz[0:4]
lorentz_peak = _1Lorentzian(x_array, *pars_L)

# this cell prints the fitting parameters with their errors
print ("-------------Peak-------------")
print ("constant = %0.2f (+/-) %0.2f" % (pars_L[0], perr_lorentz
    [0]))
print ("amplitude = %0.2f (+/-) %0.2f" % (pars_L[1], perr_lorentz
    [1]))
print ("center = %0.2f (+/-) %0.2f" % (pars_L[2], perr_lorentz[2]))
print ("width = %0.2f (+/-) %0.2f" % (pars_L[3], perr_lorentz[3]))
print ("area = %0.2f" % np.trapz(lorentz_peak))

residual_lorentz = y_array_lorentz - (_1Lorentzian(x_array, *
    popt_lorentz))

fig = plt.figure()
gs = gridspec.GridSpec(2,1, height_ratios=[1,0.25])
ax1 = fig.add_subplot(gs[0])
ax2 = fig.add_subplot(gs[1])
gs.update(hspace=0)

ax1.plot(x_array, y_array_lorentz, "ro")
ax1.plot(x_array, _1Lorentzian(x_array, *popt_lorentz), 'k--')

# peak 1
ax1.plot(x_array, lorentz_peak, "g")
ax1.fill_between(x_array, lorentz_peak.min(), lorentz_peak,
    facecolor="green", alpha=0.5)

# residual
ax2.plot(x_array, residual_lorentz, "bo")

ax2.set_xlabel("x_array",family="serif",  fontsize=12)
ax1.set_ylabel("y_array",family="serif",  fontsize=12)
ax2.set_ylabel("Res.",family="serif",  fontsize=12)


ax1.xaxis.set_major_locator(ticker.MultipleLocator(20))

ax2.xaxis.set_minor_locator(AutoMinorLocator(2))
ax1.yaxis.set_minor_locator(AutoMinorLocator(2))

ax1.xaxis.set_major_formatter(plt.NullFormatter())

ax1.tick_params(axis='x',which='major', direction="out", top="on",
    right="on", bottom="off", length=8, labelsize=8)
ax1.tick_params(axis='x',which='minor', direction="out", top="on",
    right="on", bottom="off", length=5, labelsize=8)
ax1.tick_params(axis='y',which='major', direction="out", top="on",
    right="on", bottom="off", length=8, labelsize=8)
```

```
68 ax1.tick_params(axis='y',which='minor', direction="out", top="on",
       right="on", bottom="on", length=5, labelsize=8)
69
70 ax2.tick_params(axis='x',which='major', direction="out", top="off",
       right="on", bottom="on", length=8, labelsize=8)
71 ax2.tick_params(axis='x',which='minor', direction="out", top="off",
       right="on", bottom="on", length=5, labelsize=8)
72 ax2.tick_params(axis='y',which='major', direction="out", top="off",
       right="on", bottom="on", length=8, labelsize=8)
73 ax2.tick_params(axis='y',which='minor', direction="out", top="off",
       right="on", bottom="on", length=5, labelsize=8)
74
75 fig.tight_layout()
76
77 plt.show()
78
79 ss_res = np.sum(residual_lorentz**2)
80 ss_tot = np.sum((y_array_lorentz-np.mean(y_array_lorentz))**2)
81
82 print("R^2 = ", 1-(ss_res / ss_tot))
83
84 print("N_e = ",pars_L[3]*5e15/0.074)
```

Listing B.4: Source code for calculating (n_e) from Stark Broadening

Appendix C

Spectrometer Wavelength Calibration Data

A mercury–argon (Hg–Ar) calibration lamp was used to calibrate the spectrometer's wavelength prior to LIBS measurements. These lamps produce a series of distinct, well-tabulated emission lines across the UV–visible spectrum that are frequently utilised in optical spectroscopy as reference standards[25]. The deviations between the known tabulated values and the observed line positions were noted. The findings, which are displayed in Table C.1, confirm to the instrument's extremely low deviations (all within 0.04 nm), which guaranteed high wavelength assignment accuracy throughout the plasma measurements.

Table C.1: Spectrometer wavelength calibration using Hg–Ar lamp lines.

Actual (nm)	**Observed (nm)**	**Deviation (nm)**	**Abs. Dev. (nm)**
253.652	253.653	0.001	0.001
296.728	296.721	-0.007	0.007
302.150	302.137	-0.013	0.013
313.155	313.164	0.009	0.009
365.015	365.002	-0.013	0.013
404.656	404.637	-0.019	0.019
435.833	435.818	-0.015	0.015
546.074	546.052	-0.022	0.022
576.960	576.936	-0.024	0.024
579.066	579.035	-0.031	0.031
696.543	696.511	-0.032	0.032

References

[1] David A. Cremers and Leon J. Radziemski. *Handbook of Laser-Induced Breakdown Spectroscopy*. John Wiley & Sons, 2006. ISBN 9780470092996.

[2] Prof. M.V. Sunil Krishna. Plasma physics and applications, iit roorkee, 2025. URL https://nptel.ac.in/courses/115107447. NPTEL Plasma Course.

[3] Prof. Paramita Deb. The science of light amplification: An extensive laser course, bhabha atomic research centre(barc)–iit bombay, 2025. URL https://nptel.ac.in/courses/115101643. NPTEL LASER Course.

[4] Alexander Kramida, Yuri Ralchenko, Joseph Reader, and NIST ASD Team. NIST Atomic Spectra Database (version 5.12), 2024. URL https://physics.nist.gov/asd. National Institute of Standards and Technology.

[5] Plasma (physics), 2025. URL https://en.wikipedia.org/wiki/Plasma_(physics). Accessed: 2025-06-08.

[6] Francis F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer, 3rd edition, 2016. ISBN 9783319223094.

[7] S. N. Malik. Plasma physics. Study Material, Vardhman Mahaveer Open University, Kota, 2020. URL https://assets.vmou.ac.in/MPH-09.pdf.

[8] Laser, 2025. URL https://en.wikipedia.org/wiki/Laser. Accessed: 2025-06-08.

[9] William T. Silfvast. *Laser Fundamentals*. Cambridge University Press, 2nd edition, 2004. ISBN 9780521833455.

[10] Javier Laserna, José M. Vadillo, and Pablo Purohit. Laser-induced breakdown spectroscopy (libs): Fast, effective, and agile leading edge analytical technology. *Applied Spectroscopy*, 72:35–50, 2018. doi: 10.1177/0003702818791926.

[11] David W. Hahn and Nicoló Omenetto. Laser-induced breakdown spectroscopy (libs), part i: Review of basic diagnostics and plasma—particle interactions: Still-challenging issues within the analytical plasma community. *Applied Spectroscopy*, 64 (12):335A–366A, 2010. doi: 10.1366/000370210793561691.

[12] David W. Hahn and Nicolò Omenetto. Laser-induced breakdown spectroscopy (libs), part ii: Review of instrumental and methodological approaches to material analysis and applications to different fields. *Applied Spectroscopy*, 66(4):347–419, 2012. doi: 10.1366/11-06574.

[13] Maya L. Najarian and Rosemarie C. Chinni. Temperature and electron density determination on laser-induced breakdown spectroscopy (libs) plasmas: A physical chemistry experiment. *Journal of Chemical Education*, 90:244–247, 2013. doi: 10.1021/ed3003385.

[14] A. Ciucci, M. Corsi, V. Palleschi, S. Rastelli, A. Salvetti, and E. Tognoni. New procedure for quantitative elemental analysis by laser-induced plasma spectroscopy. *Applied Spectroscopy*, 53:960–964, 1999. doi: 10.1366/0003702991947612.

[15] Alexander A. Fridman. *Plasma Chemistry*. Cambridge University Press, Cambridge, UK, 2008.

[16] Mohammed Hashim Albashir, Naga Abdalaziz, Hajhamed Diab, and Hashim Gad Elseed. Spectral line broadening mechanisms in plasmas: A theoretical study using doppler broadening models. *International Journal of Scientific Engineering and Science*, 7:122–126, 2023. ISSN 2456-7361.

[17] Dunja Bulajic, M Corsi, Gabriele Cristoforetti, Stefano Legnaioli, Vincenzo Palleschi, A Salvetti, and E Tognoni. A procedure for correcting self-absorption in calibration free-laser induced breakdown spectroscopy. *Spectrochimica Acta Part B: Atomic Spectroscopy*, 56:339–353, 2001. doi: 10.1016/S0584-8547(01)00398-6.

[18] Bremsstrahlung, 2025. URL https://en.wikipedia.org/wiki/Bremsstrahlung. Accessed: 2025-06-08.

[19] National Institute of Standards and Technology. Libs interface help file, 2025. URL https://physics.nist.gov/PhysRefData/ASD/Html/libshelp.html. Accessed: 2025-06-18.

[20] National Institute of Standards and Technology. Spectral lines help file, 2025. URL https://physics.nist.gov/PhysRefData/ASD/Html/lineshelp.html. Accessed: 2025-06-18.

[21] Taznin Hussain, M. A. Gondal, and M. Shamraiz. Determination of plasma temperature and electron density of iron in iron slag samples using laser induced breakdown spectroscopy. In *IOP Conference Series: Materials Science and Engineering*, volume 146, page 012017. IOP Publishing, 2016. doi: 10.1088/1757-899X/146/1/012017. Published under licence by IOP Publishing Ltd.

[22] M. Lemke. Extended vcs stark broadening tables for hydrogen – lyman to brackett series. *Astronomy and Astrophysics Supplement Series*, 122(2):285–292, 1997. doi: 10.1051/aas:1997134.

[23] Emily G. Ripka. Blog post: Peak fitting, 2018. URL http://emilygraceripka.com/blog/16. Accessed: 2025-06-23.

[24] Emily G. Ripka. Peak fitting jupyter notebook, 2018. URL https://github.com/emilyripka/BlogRepo/blob/master/181119_PeakFitting.ipynb. Accessed: 2025-06-23.

[25] Y. C. Sun, C. Huang, G. Xia, S. Q. Jin, and H. B. Lu. Accurate wavelength calibration method for compact ccd spectrometer. *J. Opt. Soc. Am. A*, 34(4):498–505, 2017. doi: 10.1364/JOSAA.34.000498. URL https://opg.optica.org/josaa/abstract.cfm?URI=josaa-34-4-498.