بسم الله الرحمن الرحيم

Sudan University of Science & Technology

College of Graduated Studies

كلية الدراسات العليا

Simulation of the Combustion Products of Gasoline Engine Using Laser Induced Fluorescence Technique

محاكاة نواتج الاحتراق لمحركات الجازولين باستخدام تقنية الفلورة المستحثة بالليزر

A Thesis Submitted in Fulfillment of the Requirements for the Degree of Ph.D. in Laser Application in Physics

By:
Mustafa Mohammed Badreldin

Supervisor:
Dr. Younis Ahmed Abuaisha

Co-supervisor
Dr. Khalid Mohamed Haroun

April 2019

بسم الله الرحمن الرحيم

اللَّهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مَثَلُ نُورِهِ كَمِشْكَاةٍ فِيهَا
مِصْبَاحٌ الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ
يُوقَدُ مِنْ شَجَرَةٍ مُبَارَكَةٍ زَيْتُونَةٍ لَا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا
يُضِيءُ وَلَوْ لَمْ تَمْسَسْهُ نَارٌ نُورٌ عَلَى نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَنْ
يَشَاءُ وَيَضْرِبُ اللَّهُ الْأَمْثَالَ لِلنَّاسِ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ

النور 35

Dedication

I dedicate my work to my family, especially to my mother and my father.

Also I dedicate my modest work to my small family my lovely and helpful wife " **Safa Abd Elgadir"** , my little kids "**Lila**", **"Fatima**", "**AbdElbagi**", "**Ekhlas"**, **"Mohamed almoktar**", "**Gwairia"**, and "**Amna**" and to the one's whom come after insha Alla.

Also to everyone who helped me by encourage, support or by anyway in any time during the whole time of this thesis.

Finally, my work is dedicated to the Staff of the Institute of Laser, Sudan University of Science & Technology.

Acknowledgement

First of all Thanks to Allah who give me the power and health to complete this work. Then after I would Like to express my deepest thanks and appreciate to my Esteemed Supervisor Dr. **Younis Ahmed Abuaisha** of **West Kurdfan University** with his continuous support suggestions and helps this work was complete**.** Also I would like to express my deepest thanks and appreciation to my Co-Supervisor **Dr. Khalid Mohammed Haroun** of **Alzaiem Al-Azhari University** with help and discussion, suggestions the work becomes easy to complete.

Also I would like to thank to Dr. Yousif Hassan Alsheikh of Omdurman Ahlia University for his support.

Finally thanks to everyone who help me with any kind even by words.

Abstract

Laser induced fluorescence (LIF) technique have been developed into very powerful and most widely used non- instructive techniques for measurements. This work is aimed to diagnosis the gasoline engine products using laser induce fluorescence technique modeled by python language. The software developed allowed us to diagnosis the gasoline combustion products in terms of fluorescence, quantum efficiency, pressure of the emission, and lifetime. In the simulation of the laser induce fluorescence excitation wavelengths from 200 to 700 nm were used. Gasoline combustion products (CO_x, $NO_{x,}$ H_2O, $N_2$and HC [x=1,2]) have been studied. The suitable wavelength was found to be 226 nm and typically agreed with that of Nd: YAG laser operate at 266 nm wavelength which used experimentally. The samples of the carried results showed that the simulation developed to study the emission of the combustion products such as CO_2 sample from the gasoline engine was confined . The relation of the quantum efficiency of the CO_2 with laser wavelengths at different temperatures showed that the quantum efficiency increased as the excitation wavelength increased, the lifetime decrease as the excitation wavelength increase for carbon dioxide (CO_2), the weight ratio of the CO_2 influences its fluorescence, and relationship between the fluorescence and $\mathbf{P_{emission}}$ of CO_2 as a function of its weight was studied and it was shown to be linear. A similar results was obtained for the rests of the gasoline combustion products. The results obtained were in good agreement with the literature experimental work.

المستخلص

تقنية الفلورة المستحثة بالليزر (LIF) تم تطويرها بصورة واسعة وتستخدم للقياسات غير الاتلافية, يهدف هذا البحث لتشخيص نواتج محركات الجازولين باستخدام برنامج تقنية الفلورة المستحثة بالليزر المصمم بلغة بايثون. حيث يسمح البرنامج المصمم لتشخيص نواتج احتراق الجازولين من حيث الفلورة ، الكفاءة الكمية ، ضغط الانبعاث ، وعمر النصف . حيث استخدم لإثارة نواتج الاحتراق ليزر طوله ألموجي يقع في المدى بين 200 نانومتر إلى 700 نانومتر لإحداث الفلورة.

تمت دراسة نواتج احتراق الجازولين (CO_x، NO_x، H_2O، N_2 و CH). ووجد أن الطول ألموجي 266 نانومتر هو الطول الموجي المناسب لإحداث الإثارة ووجد أن هذه النتيجة تتفق مع النتائج التي أجريت تجريبياً.

أوضحت النتائج المتحصل عليها لعينات نواتج الاحتراق مثل عينة CO_2 من محرك الجازولين كانت متوافقة مع النتائج التجريبية، و أوضح البحث أن العلاقة بين الكفاءة الكمية لثاني أكسيد الكربون مع الطول الموجي لليزر عند درجات الحرارة المختلفة تزداد بزيادة الطول الموجي ، ويقل عمرها النصفي مع زيادة طول موجة الليزر التي أحدثت الإثارة ، كما أن نسبة الوزن في ثاني أكسيد الكربونCO_2 تؤثر على فلورتها ، وقد تمت دراسة العلاقة بين الفلورة و ضغط الانبعاث لـ (CO_2) كدالة لوزنها وظهر أنها علاقة خطية.

تم الحصول على نتائج مماثلة لباقي نواتج احتراق الجازولين. النتائج التي تم الحصول عليها كانت في اتفاق جيد مع الدراسات السابقة.

CONTENTS

Chapter three
Modeling of Laser Induced Fluorescence

Chapter four
Results and Discussion

LIST OF FIGURES

LIST OF TABLES

Chapter One

Introduction

1.1 Introduction:

To everybody involved in transportation technology one of the greatest challenges for the future is the development of new propulsion systems which are free of the negative environmental effects of today’s concepts. One of the proposed solutions is the development of internal combustion engines for the use with gaseous fuels as a clean alternative. The highest potential is attributed to a configuration wherein hydrogen is injected directly into the combustion chamber, giving a whole new set of possibilities for the optimization of the combustion process (Wolfram Kirchweger et al., 2007).

There are a number of laser related techniques to study combustion processes, namely, laser induced fluorescence (LIF), Raman Spectroscopy, and Rayleigh scattering. Use of lasers for combustion research has mainly focused on gasoline (petrol) and diesel engines (Subramaniam, 2015). Laser-based detection of combustion species has been proven an important tool for combustion research . The tasks of providing measurement techniques for each the analysis of jet patterns at the injector nozzle, the measurement of the fuel distribution at ignition point, and the visualization of the flame front propagation could be fulfilled by adapting the LIF-technique for the particular conditions inside a combustion engine with high-pressure gas injection (Wolfram Kirchweger et al., 2007). There are a number of laser related techniques to study combustion processes, namely, laser induced fluorescence (LIF), Raman Spectroscopy, and Rayleigh scattering. The recording of fluorescence radiation of molecules excited by ultraviolet (UV) optical radiation is common practice in the chemical analysis of organic components. The advent of powerful lasers providing radiation at UV wavelengths allows this technique also to be used in engines. Furthermore, the narrowband tuning capabilities of excimer lasers have opened the path to LIF of specific molecular species present in the combustion process. The analysis of transfer paths and transfer probabilities gives a clear indication of how the fluorescence signal is influenced by the irradiating field (spectral irradiance), the molecular properties (population density, absorption, transfer and emission probabilities, dissociation) and the molecular environment (energy transfer concurrent to fluorescence radiation, collisional quenching). For molecules such as H_2, H_2O, NH_3, and CO that do not have accessible excitation frequencies in the UV/visible region but have resonances in the vacuum ultraviolet (VUV) region, i.e., wavelengths between 100 and 200 nm, two-photon LIF can be used to reach the desired energy level of the molecule.

The advantages of using optical techniques, such as lasers are that it provides information on the IC engine characteristics with a high degree of temporal and spatial resolution. The cost involved may be higher and the IC engines design has to be modified in order to make the laser beam access to the flame. LIF technique is a well established one to detect atoms or molecules in specific quantum states. When an atom or a molecule is excited by an incoming laser beam, the atom or the molecule goes into the excited state .The excited atom stays there for a certain period of time called as the ‘lifetime’ of the excited state and then it has to return to the ground state or the equilibrium state by giving away the extra energy that it had obtained while going up. Besides other processes of emission of energy in the form of a light photon, an atom or a molecule can undergo spontaneous emission and reaches a state other than the ground state giving a light photon and which is called as ‘fluorescence’. The energy is quantized and transitions can occur only for fixed amounts of it. In the case of a molecule undergoing a transition, vibrational and rotational transitions or relaxations causes absorption and fluorescence. The fluorescence lifetime is for one tenth of a nanosecond and the fluorescence rate, that is the number of photons emitted per unit volume per second, is dependent on the number density, saturation intensity, number of atoms or molecules that were there in the ground state, laser parameters, and other molecular constants (McCulloch et al, 2005.) Due to its advantages and high detection accuracy LIF are a subject of a heavy researches. Many parameters that influence the detection of the LIF signal of the combustion products are important in investigation of the gasoline combustion product.

1.2 Research objectives:

The objectives of this work are:

- To model the laser induced fluorescence technique (LIF) for gasoline combustion analysis using Python language.
- To use the modeled LIF software to analyze the gasoline combustion products in terms of quantum efficiency, fluorescence, life time, $P_{emission}$, and fluorescence intensity.
- To study the effect of temperature on the quantum efficiency, and on the life time.

1.3 Problem statement:

The lack of optical diagnosis techniques such as the powerful laser induced fluorescence (LIF) of the gasoline combustion products in country like Sudan requires searching to use an alternative methods one of the possible way is to use the computer software's and languages to model and study the gasoline combustion products in order to understand the physics and the problems and to suggest solutions.

1.4 Methodology:

The methodology of this research uses the computational and modeling method through the use of the Python language of the governing equations of the gasoline combustion products physical properties, the governing fluorescence process equation, the temperature, the emission pressure, the life time, the fluorescence intensity, and the vacuum chamber specifications (length or thickness), and then to build a simulation software that simulates the laser induced fluorescence detection method and to study the gasoline combustion products in terms of (types and weights of the combustion products, fluorescence, quantum efficiency, temperature, and fluorescence intensity). And to discuss the results obtained from the LIF simulation software.

1.5 Research outlines:

Chapter one is a preface of this work including objectives and the problem statement. Introduction to spectroscopy, laser induced fluorescence, and theoretical backgrounds related to the research topic are presented in chapter two.

In chapter three, brief description of the Python language, and the important equations used to build the laser induced fluorescence simulation software program, and the flow chart are given.

Finally, chapter four presents results and discussion of the laser induced fluorescence obtained for CO_x, NO_x, N_2, HC, and H_2O (where x=1,2) gasoline combustion products, conclusion and recommendation for future work.

Chapter two

Theoretical Background

2.1 Introduction:

This chapter describes the basic theoretical concepts of spectroscopy, the principle of the laser induced fluorescence and the literature review.

2.2 Spectroscopy:

Spectroscopy is the study of the absorption and emission of light and other radiation by matter, as related to the dependence of these processes on the wavelength of the radiation. More recently, the definition has been expanded to include the study of the interactions between particles such as electrons, protons, and ions, as well as their interaction with other particles as a function of their collision energy. Spectroscopic analysis has been crucial in the development of the most fundamental theories in physics, including quantum mechanics, the special and general theories of relativity, and quantum electrodynamics. Spectroscopy, as applied to high-energy collisions, has been a key tool in developing scientific understanding not only of the electromagnetic force but also of the strong and weak nuclear forces [8]. Spectroscopic techniques have been applied in virtually all technical fields of science and technology. Radio-frequency spectroscopy of nuclei in a magnetic field has been employed in a medical technique called magnetic resonance imaging (MRI) to visualize the internal soft tissue of the body with unprecedented resolution. Microwave spectroscopy was used to discover the so-called three-degree blackbody radiation, the remnant of the big bang (*i.e.,* the primeval explosion) from which the universe is thought to have originated. The internal structure of the proton and neutron and the state of the early universe up to the first thousandth of a second of its existence is being unraveled with spectroscopic techniques utilizing high-energy particle accelerators. The constituents of distant stars, intergalactic molecules, and even the primordial abundance of the elements before the formation of the first stars can be determined by optical, radio, and X-ray spectroscopy.

Optical spectroscopy is used routinely to identify the chemical composition of matter and to determine its physical structure. Spectroscopic techniques are extremely sensitive. Single atoms and even different isotopes of the same atom can be detected among 10^{20} or more atoms of a different species. Trace amounts of pollutants or contaminants are often detected most effectively by spectroscopic techniques. Certain types of microwave, optical, and gamma-ray spectroscopy are capable of measuring infinitesimal frequency shifts in narrow spectroscopic lines.

Frequency shifts as small as one part in 10^{15} of the frequency being measured can be observed with ultrahigh resolution laser techniques. Because of this sensitivity, the most accurate physical measurements have been frequency measurements [9].

2.2.1 Types of Interacting Materials:

Spectroscopic studies are designed so that the radiant energy interacts with specific types of matter. Spectroscopy is used in physical and analytical chemistry because atoms and molecules have unique spectra. As a result, these spectra can be used to detect, identify and quantify information about the atoms and molecules [8, 10]

A. Atoms

Atomic spectroscopy was the first application of spectroscopy developed. Atomic absorption spectroscopy (AAS) and atomic emission spectroscopy (AES) involve visible and ultraviolet light. These absorptions and emissions, often referred to as atomic spectral lines, are due to electronic transitions of outer shell electrons as they rise and fall from one electron orbit to another.

Atoms also have distinct x-ray spectra that are attributable to the excitation of inner shell electrons to excited states. Atoms of different elements have distinct spectra and therefore atomic spectroscopy allows for the identification and quantization of a sample's elemental composition. New elements were discovered by observing their emission spectra. Atomic absorption lines are observed in the solar spectrum and referred to as Fraunhofer lines after their discoverer. A comprehensive explanation of the hydrogen spectrum was an early success of quantum mechanics and explained the Lamb shift observed in the hydrogen spectrum led to the development of quantum electrodynamics [8, 10]. Modern implementations of atomic spectroscopy for studying visible and ultraviolet transitions include flame emission spectroscopy, inductively coupled plasma atomic emission spectroscopy, glow discharge spectroscopy, microwave induced plasma spectroscopy, laser spectroscopy and spark or arc emission spectroscopy. Techniques for studying x-ray spectra include X-ray spectroscopy and X-ray fluorescence (XRF).

B. Molecules

The combination of atoms into molecules leads to the creation of unique types of energetic states and therefore unique spectra of the transitions between these states. Molecular spectra can be obtained due to electron spin states (electron paramagnetic resonance), molecular rotations, molecular vibration and electronic states. Rotations are collective motions of the atomic nuclei and typically lead to spectra in the microwave and millimeter-wave spectral regions; rotational spectroscopy and microwave spectroscopy are synonymous. Vibrations are relative motions of the atomic nuclei and are studied by both infrared and Raman spectroscopy. Electronic excitations are studied using visible and ultraviolet spectroscopy as well as fluorescence spectroscopy.

Studies in molecular spectroscopy led to the development of the first maser and contributed to the subsequent development of the laser [11].

C. Crystals and extended materials

The combination of atoms or molecules into crystals or other extended forms leads to the creation of additional energetic states. These states are numerous and therefore have a high density of states. This high density often makes the spectra weaker and less distinct, i.e., broader. For instance, blackbody radiation is due to the thermal motions of atoms and molecules within a material. Acoustic and mechanical responses are due to collective motions as well. Pure crystals, though, can have distinct spectral transitions and the crystal arrangement also has an effect on the observed molecular spectra. The regular lattice structure of crystals also scatters x-rays, electrons or neutrons allowing for crystallographic studies.

D. Nuclei

Nuclei also have distinct energy states that are widely separated and lead to gamma ray spectra. Distinct nuclear spin states can have their energy separated by a magnetic field, and this allows for NMR spectroscopy [9].

2.2.2 Absorption spectroscopy:

Atoms and molecules can absorb light/photons over a large range of wavelengths, in the range from the UV $(\lambda \leq 400nm, h\nu \geq 6.2eV)$, through the visible $(\lambda = 400 - 700nm, h\nu \cong 1.6 - 6.2eV)$, to the IR $(\lambda \geq 700nm, h\nu \leq 1.6eV)$. It should be noted that, by and large, IR absorption is restricted to molecules, which exhibit rotational and vibrational level energy structures with a narrow level spacing for allowed dipole transitions, of the order of fractions of an electron volt

$$(< 10^{-3} - 10^{-1} eV)$$

The energy level spacing between the ground state and excite states in atoms is usually of the order >> $1eV$ .The consequence of absorption is that for a beam of light passing through an absorbing medium. The radiation is attenuated. The general principle of light attenuation on passage through a (absorbing) medium is shown in Figure 2.1. This attenuation can be described quantitatively by using the so-called Beer-Lambert law.

2.2.3 The Beer-Lambert law:

The Beer-Lambert law (or Beer’s law) is the linear relationship between absorbance A and number density N of an absorbing species. The Beer-Lambert law is frequently written in the simple form:

$$A = \sigma NL \qquad (2.1)$$

where A (dimensionless) is the measured absorbance, $\sigma(cm^2)$ is a frequency-dependent (or wavelength- dependent) absorption coefficient, $L(cm)$ is the path length, and $N(cm^{-3})$ is the analyte particle density, and $\alpha(cm^{-1})$ is the attenuation coefficient. Experimental measurements are usually made in terms of transmittance T, which is defined as:

$$T = \frac{I_L}{I_O} \qquad (2.2)$$

Where I_L is the light intensity after it has passed through a sample and I_0 is the initial light intensity. The relation between A and T is:

$$A = -1nT = -1n\left(\frac{I_L}{I_0}\right) \qquad (2.3)$$

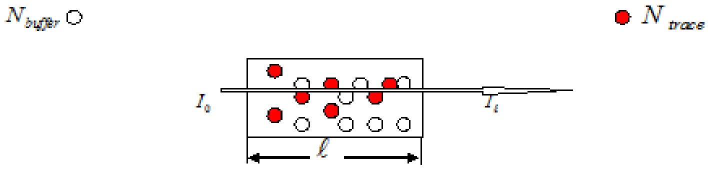


Figure 2.1: Principle of attenuation of light beam when passing through a medium containing absorbing particles (Helmut H. Telle, Angel Gonzalez Urena, Robert J. Donovan. 2007. Laser Chemistry)

2.2.4 Derivation of the Beer-Lambert law:

The Beer-Lambert law can be derived from an approximation for the absorption coefficient for a molecule by approximating the molecule by an opaque disk whose cross-sectional area σ represents the effective area seen by a photon of frequency ν. If the light is far from resonance with an atomic or molecular transition, then the area is approximately zero, and the area is a maximum ν if is close to resonance. Notes that the formulation term of $\sigma(\nu)$ or $\sigma(\lambda)$ are equivalent; one only need to convert light frequencies into light wavelengths, exploiting the relation $\nu\lambda = c$.

Note also that in order to obtain the total absorption cross-section S of a line one needs to integrate over the full line profile, which yields:

$$S = \int_0^\infty \sigma(\nu)d\nu \qquad (2.4)$$

Assume that the photons travel in the z -direction. Taking an infinitesimal slab dz of sample, one can derive the attenuation of light in this slab using the sketch in figure 2.2, I_0 is the intensity entering the sample at $z = 0, I_z$ is the intensity entering the infinitesimal slab at z, dI is the intensity absorbed in the slab, and I is the intensity of light leaving the sample (of total length λ) .The total opaque area Q on the limited by a number of chemical and instrumental factors. Causes of non-linearity include:

- deviations in absorption coefficient at high molar concentrations (>0.01 M) due to interactions between molecules in close proximity (aggregate formation);

- scattering of light due to particulates in the sample;
- fluorescence or phosphorescence of the sample;
- Stray light.

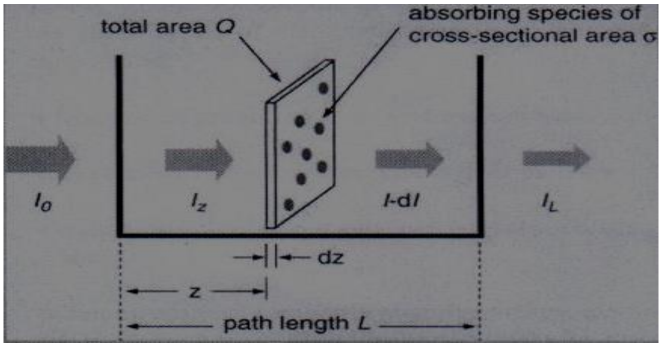


Figure 2.2: Attenuation of a laser light beam on passage through an absorbing medium (Helmut H. Telle, Angel Gonzalez Urena, Robert J. Donovan. 2007. Laser Chemistry)

Slab due to the absorbers is $\sigma NQdz$; consequently, the fraction of photons absorbed is $\sigma NQdz/Q$ and hence:

$$\frac{dI}{I_z} = -\sigma Ndz \qquad (2.5)$$

Integrating this equation from z=0 to z = L yields

$$1n(I_L) - 1n(I_0) = -\sigma NL \qquad (2.6)$$

Or

$$-1n(I_L/I_0) = \sigma NL \equiv A \qquad (2.7)$$

Or in exponential representation

$$I(L) = I_0 \exp(-\sigma NL) \qquad (2.8)$$

Occasionally, one also finds that the product of the absorption coefficient and the particle number density is abbreviated as

$$\alpha(cm^{-1}) = \sigma N \tag{2.9}$$

This is known as the attenuation coefficient.

Note that in analytical chemistry one customary uses the molar absorbtivity $\varepsilon(M^{-1}cm^{-1})$, $\sigma(cm^2)$ instead of absorption cross-section Also, the absorbance A is often expressed in terms of the decade rather than the natural logarithm (recall that In x=2.303log x) . Exploiting the well-known relation between number densities and concentrations (moles/liter), and that mole incorporates 6.023 x 10^{23} particles (also known as the Avogadro number), one finds for the relation between σ and ε

$$\varepsilon = \frac{6.023\times10^{23}/10^{3}}{2.303}\sigma = 2.61\times10^{20}\sigma \tag{2.10}$$

Table (2.1) shows a typical cross sections and molar absorbtivity values of molecules and that of the possible transitions

Table 2.1: Typical cross-sections and molar absorbtivity values are as follows:

Interacting	$\sigma(cm^2)$	$\varepsilon(M^{-1}cm^{-1})$
Molecules	$10^{-12} to 10^{-15}$	$3\times10^{8} to 3\times10^{5}$
Electronic transitions	$10^{-14} to 10^{-17}$	$3\times10^{6} to 3\times10^{3}$
Vibrational transitions	$10^{-19} to 10^{-21}$	$3\times10^{0} to 3\times10^{-2}$
Raman scattering	$\sim10^{-29}$	$\sim3\times10^{-9}$

2.2.5 Emission Spectra

The discrete bright (dark) lines in the emission (absorption) spectrum can be explained by treating light as a photon that is emitted (absorbed) by an atom, as shown in Figure (2.3), For this lab we are going to concentrate on emission spectra. In the quantum model of the atom, electrons exist only in specific energy states. The photon emitted from an atom when an electron "falls" from an excited energy state to a lower state is limited to the difference between those states, so only specific energies of light are emitted.

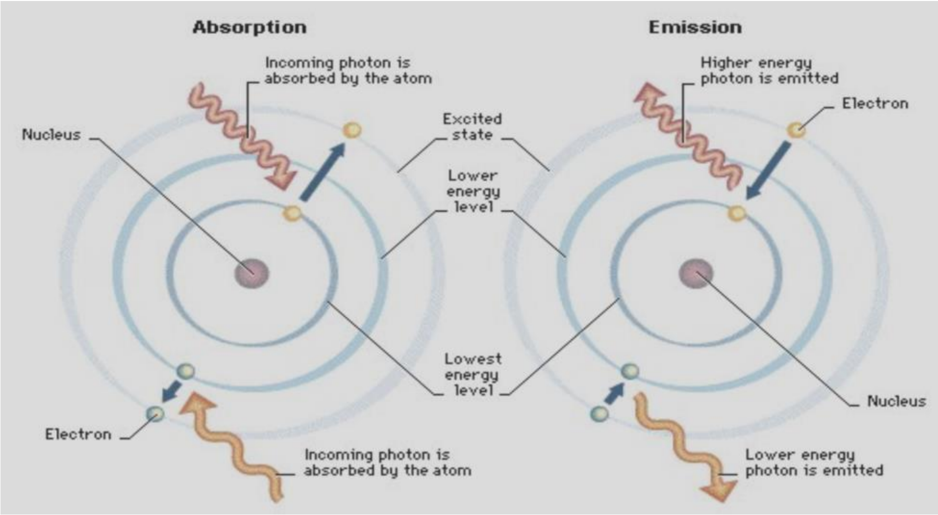

Absorption
Incoming photon is absorbed by the atom
Nucleus
Excited state
Lower energy level
Lowest energy level
Electron
Incoming photon is absorbed by the atom
Emission
Higher energy photon is emitted
Electron
Nucleus
Lower energy photon is emitted

Figure 2.3: Model of an atom showing absorption and emission of photons. (Helmut H. Telle, Angel Gonzalez Urena, Robert J. Donovan. 2007. Laser Chemistry)

The periodic table of the elements is also explained by the atomic model, where every element is uniquely identified by the number of protons in its nucleus. Quantum mechanics explains the energy states of the electrons in an atom. Every element in the periodic table has a distinct set of electron energy levels, so for a given element only photons of specific energies can be emitted. Therefore, when you are measuring the emission spectrum of an element, only certain wavelengths of light are allowed and the "pattern" that is produced is unique for that substance. Notice the differences in the emission spectra for hydrogen (H), helium (He) and mercury (Hg) that are shown in Figure (2.4). The energy of an emitted photon and its wavelength are related.

For example, the color of a laser pointer (e.g. red or green) is determined by the energy of the emitted light. The energy of a photon is described by the equation:

$$E = \frac{hc}{\lambda} \tag{2.11}$$

where λ is the wavelength in meters, h is the Planck constant (h ≈ 6.63 × 10^{-34} J s), and c is the speed of light (c ≈ 3.00 × 10^{8} m/s). When dealing with visible light, wavelengths are usually given in nanometers (nm) so it is convenient to convert the quantity hc into units of electron volts (eV) now be written as:

$$E = \frac{hc}{\lambda} = \frac{1240 \text{ eV nm}}{\lambda} \tag{2.12}$$

where λ is the wavelength in nm and the energy of the photon is in eV.

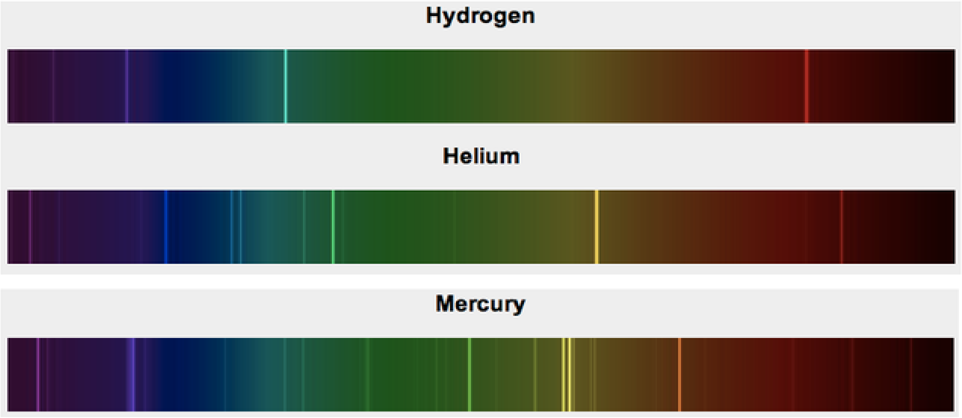


Figure 2.4: Emission spectra for hydrogen, helium and mercury (Helmut H. Telle, Angel Gonzalez Urena, Robert J. Donovan. 2007. Laser Chemistry)

2.3 Laser Scattering Spectroscopy:

Laser light scattering spectroscopy is based on the evaluation of the frequency shift of coherent light scattered by moving particles. This makes it particularly suitable for use in light guiding systems. It has been used in recent years to study the diffusive motions of macro-molecules in solution, bacterial movement. Laser light scattering spectroscopy is based on the properties of laser light: monochromaticity, coherence, and directionality [15]. Scattering spectroscopy measures the amount of light that a substance scatters at certain wavelengths, incident angles, and polarization angles. One of the most useful applications of light scattering spectroscopy is Raman spectroscopy .

2.4 Phenomenon of fluorescence:

Fluorescence is an emission of light, which occurs from singlet-excited states to the singlet ground state. The electron in the excited orbital is paired of opposite spin to second electron in the ground state orbital. Then it returns to its ground state orbital by emission of photon. Fluorescence lifetime is near 10ns (10^{-9}s). Fluorescence emission spectrum is a plot of the fluorescence intensity versus wavelength (nanometers). Even if the electronic transition promotes the specie to an energy state higher than S_1, the excess energy in condensed system is dissipated away to the surrounding as thermal energy and the species comes to stay in the lowest excited state .

2.4.1 Jablonski Diagram and the lifetime: -

The Jabolnski diagram is a diagram was illustrated the processes, which occur between the absorption and emission. It is sometime used as starting point for discussing light absorption and emission. They exist in a variety of forms, to illustrate various molecular processes, which can occur in excited states . Figure (2.5) is shown a typical of Jabolnski diagram, the singlet ground, first and second electronic state depicted by $S_{0,}$ S_1 and S_2 respectively.

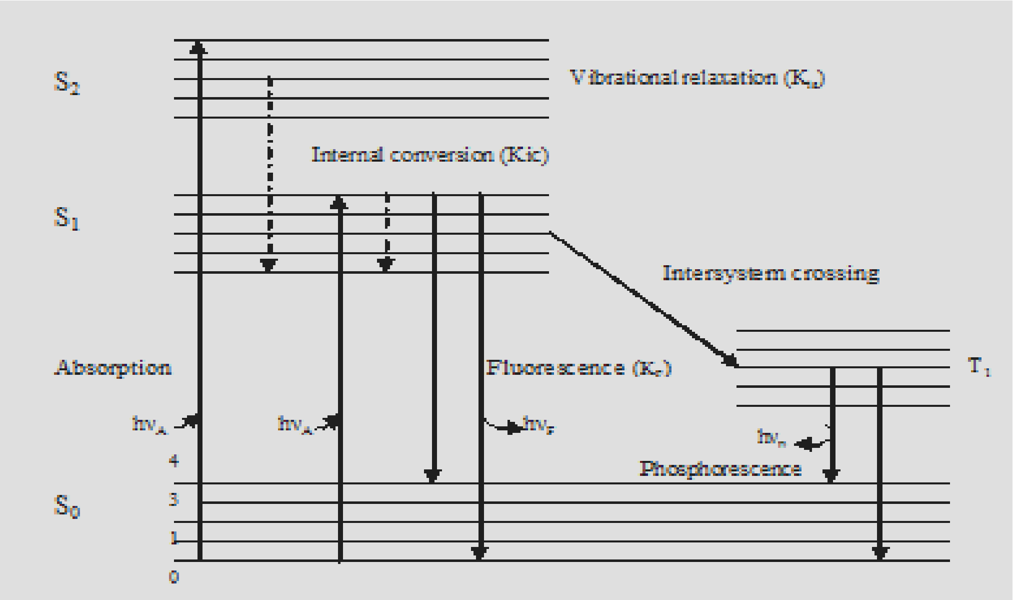


Figure 2.5: Jabolnski diagram used as starting point for discussing light absorption and emission. (*Orazio Svelto*.1998. Principles of Lasers)

All of these electronic energy levels the fluorophoreres can exist in a number of vibrational energy levels, denoted by 0, 1, 2 etc. In this diagram we have excluded a number of interactions, such as quenching, energy transfer, and solvent interactions. The transitions between states are depicted as vertical line to illustrate the instantaneous nature of light absorption. The lifetime of the transitions occur in about 10^{-15}s.

The large energy difference between the S_0 and S_1 excited state is too large for thermal population of S_0 , S_1, for this reason we used light and not heat to induce fluorescence .

The internal conversion is process which of a few rare exception, molecules in condensed phase rapidly relax to the lowest vibrational level of S_1, generally occurs in 10^{-12}s or less. The lifetimes of the fluorescence are typically near 10^{-8} s. The internal conversion is generally complete prior to emission .

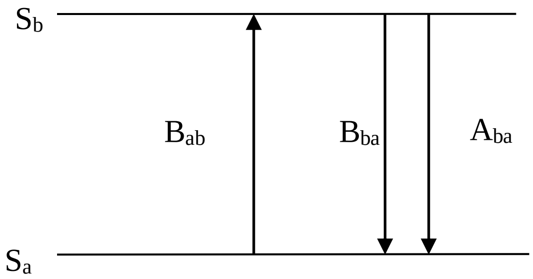
Hence fluorescence emission generally results from thermally equilibrated excited state, that is the lowest –energy vibrational state S_1.

Return to the ground state typically occurs to a higher excited vibrational ground-state level this occurs in 10^{-12}s. Molecules in the S_1 state can also undergo a spin conversion to the first triplet T_1.

Emission from T_1 is termed phosphorescence and is generally shifted to longer wavelength (lower energy). The inter system crossing is the conversion of S_1 to T_1. Transition from T1 to the singlet ground state is forbidden, and as a result rate constants for triplet emission are several orders of magnitude smaller than those for fluorescence.

2.4.2 Basic principles of fluorescence:

Consider species in the two energy levels S_b and S_a, as shown in the diagram below:



The light of radiation density I (v) induces the transition from S_a to S_b at rate B_{ab} per species. The radiation-induced process S_b to S_a occurs at exactly the same rate; thus .

$$B_{ab} = B_{ba} \qquad (2.13)$$

If the system originally contains n$_a$ species in state S$_a$ and n$_b$ in state S$_b$, the net rates of conversion are n$_a$B$_{ab}$I(υ) and n$_b$B$_{ba}$I(υ), in the equilibrium.

$$n_a B_{ab} I(\nu) = n_a B_{ba} I(\nu) \tag{2.14}$$

or $n_b = n_a$, independent of radiation density. Without light virtually all species should be in the ground state, S$_a$. At low light intensities, one hardly expects a major perturbation of the equilibrium values of n$_a$ and n$_b$ these values are derivable from statistical mechanics. Simple Boltzmann factor applies [12].

$$\frac{n_a}{n_b} = \exp\{-E_a - E_b / KT\} = e^{h\nu/KT} \tag{2.15}$$

where:

h≡ Planck's constant

T≡ temperature

K≡ Boltzmann's constant

E≡ energy

The rate of this process (A$_{ba}$) should be independent of I (ν). When spontaneous emission is induced and the rates of Intercoversion of S$_a$ and S$_b$ are set equal at equilibrium.

$$n_a / n_b = [B_{ba} I(\nu) + A_{ba}] / B_{ab}\, I(\nu) = (1 + A_{ba}) / B_{ab} I(\nu) \tag{2.16}$$

To do this we must first insert for I(ν) the radiation density expected for black body at temperature T .

$$\mathrm{I}(\nu) = 8\pi h \nu^3 / c^3 \left(e^{h\upsilon/_{KT}} - 1 \right) \tag{2.17}$$

Put equation (2.14) in equation (2.13)

$$n_a / n_b = 1 + A_{ba} / B_{ab} I(\nu) = 1 + A_{ba} / B_{ab} I(8\pi h\nu^3 / c^3 (e^{h\nu/KT} - 1)) \qquad (2.18)$$

When this value is set equal to $e^{h\nu/KT}$

$$1 + A_{ba}(e^{h\nu/KT} - 1)/8\pi h\nu^3 c^{-3} B_{ab} = e^{h\nu/KT} \qquad (2.19)$$

$$8\pi h\nu^3 c^{-3} B_{ab} + A_{ba}(e^{h\nu/KT} - 1) = e^{h\nu/KT}(8\pi h\nu^3 c^{-3}) B_{ab} \qquad (2.20)$$

$$A_{ba}(e^{h\nu/KT} - 1) = (e^{h\nu/KT} - 1)(8\pi h\nu^3 c^{-3}) B_{ab} \qquad (2.21)$$

$$A_{ba} = 8\pi\nu^3 c^{-3} B_{ab} \qquad (2.22)$$

This expression depends of the frequency. This means that at short wave lengths A_{ba} much larger than B_{ab}. We showed that .

$$B_{ba} = (2\pi / 3\hbar^2) D_{ab} \qquad (2.23)$$

Thus we can obtain

$$A_{ba} = (32\pi 3\nu^3 / 3c^3\hbar) D_{ab} \qquad (2.24)$$

D_{ab} ≡ Dipole strength

In absence of radiation or any other perturbations or interaction, the rate of excitation of species initially in state S_b will be

$$dn_b / dt = -A_{ba} n_b \qquad (2.25)$$

$$\int dn_b / n_b = \int - A_{ba} dt \qquad (2.26)$$

$$n_{ba}(t) = n_b(0)\exp(-A_{ba}t) \qquad (2.27)$$

Where n_b (0) is concentration of excited states at zero time. The radioactive lifetime of state S_b is:

$$\tau_R = 1/A_{ba} \qquad (2.28)$$

The intrinsic fluorescence rate constant (K_F) can be discussion as

$$K_F = A_{ba} = 1/\tau_R \qquad (2.29)$$

The fluorescence quantum yield (ϕ_F) is equal to the ratio of photons emitted to photons absorbed by the system. The fluorescence quantum yield (ϕ_F) can be evaluated from the preceding discussion as .

$$\phi_F = K_F / \lfloor K_F + K_{ic} + K_{is} + K_q(Q) \rfloor \qquad (2.30)$$

Where K_{ic} ≡ induced internal conversion

K_{is} ≡inter system crossing

$K_q(Q)$ ≡ quenching of various types

The kinetic equation describing the decay of the concentration of exited singlet, ($S_b(t)$) is constructed by adding all parallel de-excitation path way [12].

$$-dS_b/dt = \lfloor K_f + K_{ic}K_{is} + K_q(Q) \rfloor S_b \qquad (2.31)$$

This equation has the solution

$$S_b(t) = (S_b(0))e^{-t/\tau_f} \qquad (2.32)$$

Where ($S_b(0)$) is the concentration at zero and τ_F is the fluorescence decay time.

$$\tau_F = [K_F + K_{ic} + K_{is} + K_q(Q)]^{-1} \qquad (2.33)$$

Combining the definitions of τ_F, τ_R and ϕ_F we obtain [12].

$$\phi_F = \tau_F / \tau_R \qquad (2.34)$$

2.4.3 Characteristics of fluorescence:

Fluorescence occurs at lower energies. General property of fluorescence is that the same fluorescence emission spectrum is generally observed irrespective of the excitation wavelength. In fluorescence but many exceptions occur.

These exceptions are due to different reasons like:

1.Geometric arrangement of nuclei in the excited stated as compared to the ground state .

2.Excited state reactions.

2.5 Laser induced fluorescence:

Laser Induced Fluorescence (LIF) is a sensitive measurement technique. It has a large range of applications in spectroscopy.

LIF serves as monitor for the absorption of laser photon in fluorescence excitation spectroscopy. Also it can gain information on molecular states. Another aspect of LIF is the spectroscopic study of collision processes, and it is application to the determination of the internal state distribution in molecular reaction. It has many advantages; some of them are required small sample volumes, increased signal to noise ratio, and at low light levels the intensity of fluorescence is directly proportional to emitter concentration .

2.5.1 Theory of laser induced fluorescence:

For homogeneous broad line in which the spectral line width exceeds the laser line width and if one assumes the line width is Lorentzian, then the line shape function is: .

$$f\left(\nu_0 - \nu\right) = \frac{A_{ij} - Q}{4\pi^2\left(\nu_0 - \nu\right)^2 + \left[\left(A_{ij} + Q\right)/2\right]^2} \qquad (2.35)$$

Where: ν_0 is the absorber frequency?

ν is the laser frequency

A_{ij} is the radiation decay rate between level i and j

ϕ is the total coalitional relaxation rate

In equilibrium, excitation and deactivation balance for each level. If N is the population of the level then,

$$\frac{N_2}{N} = \frac{D_\rho B_{12}}{\left(1+L\right)4\pi^2\left(\nu_0 - \nu\right)\left[\left(A_{21} + Q\right)/2\right]^2 + D_\rho\{B_{21} + \left[\left(1+K\right)/\left(1+L\right)\right]B_{12}\}} \qquad (2.\ 36)$$

Where N is the total number of fluorescent species; $K = \frac{A_{23} + Q_{23}}{Q_{31} + A_{31}}$, $L = \frac{Q_{13}}{Q_{31} + +A_{31}}$

and $D = \frac{A_{21} + Q}{A_{21} + Q_{21} + A_{32} + Q_{23}}$

Where Q_{ij} represent the collisional relaxation rate between levels *i* and *j*. If the transition from level 1 to 2 has width $\Delta_{\gamma d}$

$$N_2 / N = \left(F, a\right)\frac{R}{\left(1+R\right)\left(1+L\right)}\frac{1}{\left[g_1 / g_2 + \left(1+K\right)/\left(1+L\right)\right]} \qquad (2.37)$$

Where

$$F\left(a, R\right) = \pi^{1/2} a\left(1+R\right)^{1/2} \exp\left[a^2\left(1+R\right)\right] erfc\left[a\left(1+R\right)^{1/2}\right] \qquad (2.38)$$

In which a can be found using: $a = 0.13\frac{A_{21}+Q}{\Delta_{\upsilon d}}$

$$R = \frac{4D_{\rho}\left[B_{21}+\{(1+K)/(1+L)\}B_{12}\right]}{(A_{21}+Q)^2} \tag{2.39}$$

And erfc is the error function,

g$_1$: and g$_2$ are the degeneracies of level 1 and level 2 respectively,

a: measures the ratio of the homogeneous line width to Doppler width

R: is the saturation of the 1→2 transition

When a or R become large, the parameter F (a, R) approaches unity, in the case the fractional population in level 2 becomes

$$\lim_{R\to\infty} F(a,R) \to 1; \left(\frac{N_2}{N}\right) = \frac{1}{(1+L)\left[g_1/g_2+(1+K)/(1+L)\right]} \tag{2.40}$$

Fluorescent photons are emitted from the excited volume, at the rate

$$\eta = A_{21}VN_2 \tag{2. 41}$$

The energy of each photon is hυ where υ is the laser frequency and h is Planck's constant. Then the fluorescent power collected by an optical system, that accepts light over solid angle Ω is ,

$$P_F = h\nu\frac{\Omega}{4\pi}N_2A_{21}V \tag{2.42}$$

For a two -level systems the population in level 2 is

$$\frac{N_2}{N} = \frac{1}{(g_1/g_2+1)}\frac{R}{(1+R)}F(a,R) \tag{2.43}$$

With

$$R = \frac{4D_{\rho}(B_{21}+B_{12})}{(A_{21}+Q)^2} \tag{2.44}$$

When saturation occurs

$$N_2/N = \left[(g_1/g_2)+1\right]^{-1} \tag{2.45}$$

If P is measured under controlled conditions

$$N = \frac{4\pi P}{h\nu\Omega A_{21}V}\left(\frac{g_1}{g_2}+1\right) \qquad (2.46)$$

And N The emitter density can be obtained directly.

The laser intensity required to achieve saturation in a two level system, can be estimated from

$$B_{12}\rho(\nu_0)\rangle\rangle Q + A_{21} \qquad (2.47)$$

Where

$\rho(\nu_\circ)$ is the laser energy density when tuned to the center of the absorption line,

$$B = \frac{A\lambda^3}{8\pi h} \qquad (2.48)$$

Where:

λ is wavelength.

Laser intensity can be written as:

$$I_{\upsilon 0} = \rho(\nu_0)c\Delta_\upsilon \qquad (2.49)$$

According to the analysis of Daily and Baronavsski, the laser line width exceeds the spectral line width

$$\frac{N_2}{N} = \frac{1}{[(1+g_1)/g_2]}\frac{r}{(r+1)} \qquad (2.50)$$

Where

$$r = \frac{(B_{12}+B_{21})\rho}{Q+A_{21}}$$

If the laser pulse is tailored to saturate the 1→2 transition and then switch off abruptly, the time dependence of P_F

$$P_F(t) = P_F(0)\exp[-(Q+A_{21})t] \qquad (2.51)$$

The above relation can give a value for Q, if A_{21} is known, then:

$$N_2(total)=\frac{4\pi}{\Omega h\nu A_{21}V}\int_0^\infty P_F(t)dt \qquad (2.52)$$

A measurement of the decay carve for fluorescence then provides direct information on N$_2$ (total) and hence N$_1$ (total).this method can be applied only when quenching is not large. In the region near saturation

$$\frac{N_2}{N_1}=\frac{g_2}{(g_1+g_2)}\frac{(B_{12}+B_{21})\rho}{[A_{12}+Q+\rho(B_{12}+B_{21})]} \qquad (2.53)$$

Then

$$p_F=h\nu\frac{\Omega}{4\pi}A_{21}V\frac{g_2}{(g_1+g_2)}\frac{(B_{12}+B_{21})\rho N}{[A_{21}+Q+\rho(B_{12}+B_{21})]} \qquad (2.54)$$

With g$_1$=g$_2$ and B$_{12}$=B$_{21}$ this becomes

$$P_F=h\nu\frac{\Omega}{4\pi}A_{21}V\frac{\rho NB}{A_{21}+Q+2\rho B} \qquad (2.55)$$

$$P_F\cong h\nu\frac{\Omega}{8\pi}A_{21}VN\left(1-\frac{Q+A_{21}}{2\rho B}\right) \qquad (2.56)$$

From this equation a plot of P$_F$ vs. ρ^{-1} yield a straight line with slope is from which Q can be obtained. The intercept of this $[(Q+A_{21})/2B]$ plot gives N directly .

2.6 Literature review:

In (2013) **Joakim Rosell** et al. presented an investigation of three excitation/detection schemes for two-photon excitation laser-induced fluorescence on carbon monoxide. The schemes are evaluated for pressure and quenching partner dependencies and C_2 interference.

W.G. Bessler et al. in 2003 investigated the Laser-induced fluorescence (LIF) of carbon dioxide with excitation between 215 and 255 nm with spectrally resolved detection in 5–40 bar premixed CH4/O_2/Ar and CH_4/air flat-flames at fuel/air ratios between 0.8 and 1.9. They found that the LIF signal consists of a broad (200–450 nm) continuum with a faint superimposed structure, and this signal is absent in similar H_2/O_2/Ar flames. They concluded that there is strong evidence this signal arises from CO_2, as the signal variations with excitation wavelength, equivalence ratio and flame temperature all correlate with CO_2 absorption cross-sections, and they showed that signal is linear with pressure and laser fluence within the investigated ranges.

Wolfram Kirchweger et al. in 2007, was successfully applied the method of laser-induced fluorescence for the measurement requirements in the field of development of innovative combustion processes for hydrogen engines. One main goal achieved was the development of methods suitable for quantified measurements of the fuel distribution during mixture formation. The excellent linearity of the LIF-signal when using Triethylamine as a tracer lead to the development of a very time efficient calibration routine, which can be used as a standard procedure for measurements of mixture formation. The resulting images provide an essential data basis for the development of new mixture formation concepts and the verification of 3D-CFD models. The other goal was to apply a method for the measurement of the flame propagation, whereas OH-LIPF has provided the best accuracy, whereas the method of measuring the flame front with a tracer has shown great potential for giving a quick overview of the effects of variations in combustion parameters. Used in combination these, techniques provide the tools for further improvements in the field of hydrogen engines and therefore can help to advance clean mobility for the future.

T. K. Subramaniam in 2015 Analyzed of Internal Combustion Engines Using Laser Induced Fluorescence Spectroscopy. The aim and scope of his paper was to bring out the methods to improve combustion efficiency and to reduce thereby the emission of pollutants by the automobiles into the atmosphere, causing vehicular pollution. Internal Combustion (IC) engines have to be modified with the inclusion of glass windows that will permit laser photons to enter and exit the combustion chamber. Three factors, have been identified and studied here, namely, a)Fuel and Air mixing ratio, b)Spatial and temporal variations of temperature inside the combustion chamber and c)Chemical intermediates concentration that rapidly change with time. Excimer lasers (or) excited exciplex lasers are used, notably, ArF (193 nm) (or) XeCl (308 nm) are generally used to study fuel-air mixing process. Studying nitrous oxide (NO) distribution inside the combustion chamber leads to a better understanding of the NOx formation mechanisms and this will ultimately help to improve the efficiency of the IC engine and reduction of the NOx emissions. Imaging of hydroxide (OH) radicals inside the combustion chamber will give details of the structure of the flame , which in turn, provides qualitative measure of the combustion efficiency .Only liquid fuel oils are discussed in this paper.

A .M. Emam in 2010 was the utilization of laser induced breakdown spectroscopy (LIBS) technique in combustion characterization to measure local species concentration of carbon, hydrogen, nitrogen and oxygen in the flame of special designed burner beside the measurement of fuel /air ratio, and equivalence ratio. Qswitched-Nd: YAG Laser oriented in orthogonal configuration was used with another Q-swtiched-Nd: YAG laser, with similar parameters, to enhance LIBS signals detected from the flame. Each of these lasers has 10 ns pulse duration, 10 Hz repetition rate and 100 mJ energy per pulse. These lasers were applied to form plasma inside the flame. Spectrograph of ECHELLE type, combined with intensified charge coupled devices camera, was used to record the emission spectra of the combustion plasma. The stability curve for liquid petroleum gas (LPG) fuel in special design burner was deduced to characterize the combustion process for LPG in this burner, then mass fraction of the elements in the flame were calculated in additional to air/fuel ratio at different heights in the flame and through the width of the flame. The mass fraction profiles of the elements was deduced and from the mass fraction profiles of carbon and hydrogen the reaction zone of the flame was obtained to be range from ratio 2.5 to ratio 3 for the height 0.5 cm above the burner and from the ratio 2 to ratio 3 for the other heights. Nitrogen and oxygen masses fractions were calculated at the boundary of the flame (outside the flame), which were found to be 0.77 and 0.23, respectively. The mass fractions were found in good agreement with literatures. For the special design burner used in this work, the results showed that the combustion is rich because ϕ in the ring from 1to 2.5, for the ratio less than three.

Chapter three

Modeling of Laser Induced Fluorescence

3.1 Introduction:

This chapter describes the important equations used in building the simulation software of the laser induced fluorescence (LIF) of the gasoline combustion products of the gasoline engine, and give a description of the software and at its end the method to use the software in the analysis and measurements of the gasoline combustion products.

3.2 PYTHON

PYTHON is a higher-level programming language. It is considered to be a higher-level language than C, C++, Java, and C#. Guido van Rossum created the **PYTHON** programming language in the late 1980s. In contrast to other popular languages such as C, C++, Java, and C#, **PYTHON** strives to provide a simple but powerful syntax. **PYTHON** is used for software development at companies and organizations such as Google, Yahoo, CERN, Industrial Light and Magic, and NASA. Experienced programmers can accomplish great things with Python, but Python's beauty is that it is accessible to beginning programmers and allows them to tackle interesting problems more quickly than many other, more complex languages that have a steeper learning curve (Richard L. Halterman, 2011). More information about **PYTHON**, including links to download the latest version for Microsoft Windows, Mac OS X, and Linux, can be found at https://www. **PYTHON**.org.

Writing a **PYTHON** Program Python programs must be written with a particular structure. The syntax must be correct, or the interpreter will generate error messages and not execute the program. This section introduces **PYTHON** by providing a simple example program

3.3 Modeling of the LIF using PYTHON:

The governing process equations to be modeled are that describes the laser induced fluorescence of the gasoline combustion products, according to the problem definitions starting by understanding the fluorescence.

When a fluorophoreres absorbs a photon of light, an energetically excited state is formed. The fate of this species is varied, depending upon the exact nature of the fluorophoreres and its surroundings, but the end result is deactivation (loss of energy) and return to the ground state. The main deactivation processes which occur are fluorescence (loss of energy by emission of a photon), internal conversion and vibrational relaxation (non-radiative loss of energy as heat to the surroundings), and intersystem crossing to the triplet manifold and subsequent non-radiative deactivation.

The fluorescence quantum yield (Φ) is the ratio of photons absorbed to photons emitted through fluorescence. In other words the quantum yield gives the probability of the excited state being deactivated by fluorescence rather than by another, non-radiative mechanism. As a result the first equation that must be taken into account in modeling is the quantum efficiency as:

The fluorescence quantum efficiency or sometime known as fluorescence quantum yield is the ratio of the number of photons emitted by the fluorophoreres to the number absorbed. The possible values are between 0 and 1. For photochemical reaction which is expressed by: $E = h\nu$; and using the relation between frequency and wavelength:

$$v = \frac{c}{\lambda} \qquad \mathbf{(3\,.1)}$$

where C is the speed of light in vacuum, using a laser 266 nm in this study as it's a widely used wavelength in experimental works of LIF [7]. Then the frequency can be calculated.

Second: the initial intensity of the emission which is given by:

$$\mathrm{I}_0 = \frac{2\nu^2\ \mathrm{K_B}\ \mathrm{T}}{\mathrm{c}^2} \qquad \mathbf{(3\,.2)}$$

And from the fact that the

$$number\ mole = \frac{whieght}{molar\ mass} \qquad \mathbf{(3\,.3)}$$

so the number of decomposed molecules can be expressed as:

$$number\ of\ molecules\ decomposed = number\ m\ ole \times N_A \quad (3\,,4)$$

Therefore the number of photons absorbed will be:

$$number\ of\ photons\ absorbed\ = I_0\ \frac{\lambda}{\mathrm{hc}}\ t \qquad \mathbf{(3\,.5)}$$

There for the quantum efficiency will be

$$\emptyset\ = \frac{\boldsymbol{Number\ of\ molcules\ decomposed\ or\ formed}}{\boldsymbol{Number\ of\ photons\ of\ radiation\ enery\ absorbed}} \qquad \mathbf{(3\,.6)}$$

The absorption equation is:

$$\boldsymbol{A = \varepsilon \times l \times c} \qquad (\mathbf{3\,.7})$$

Where ε_m is the molar absorbtivity, taking the fact that the molecular molar absorbtivity's are in the range

$$3\ \times\ 10^5\ to\ 3\ \times\ 10^8\ \ M^{-1}\ cm^{-1}$$

and the concentration can be calculated from:

$$concentration = \frac{wieght}{molar\ mass} \times \frac{22.414}{1.6} \qquad \mathbf{(3\ .8)}$$

Finally the equation that describes the fluorescence is given by :

$$F = I_0 \times \emptyset \times A \qquad (3\ .9)$$

The black body radiation law can be used to model and calculate the P$_{\text{emission}}$, first we must calculate the emissivity from:

$$\mathrm{E}(\lambda, \mathrm{T}) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{\exp\left(\frac{hf}{KT}\right) - 1} \qquad (3\ .10)$$

$$\epsilon = \frac{E\ (\lambda)}{\mathrm{E}(\lambda, T)} \qquad (3\ .11\ )$$

Where $E(\lambda)$ represent the real body energy at the same wavelength, and $E(\lambda,T)$ is the black body energy at the same wavelength and temperature. Bodies radiate thermal energy according to their temperature. The emissions are electromagnetic radiation, and therefore have the properties of energy and momentum. The energy leaving a body tends to reduce its temperature. The momentum of the radiation causes a reactive force, expressed as a pressure across the radiating surface.

The Stefan–Boltzmann law describes the power radiated from a black body. The law states that the total energy radiated per unit surface area of a black body across all wavelengths per unit time (also known as the black-body radiant exitance or emissive power) is directly proportional to the fourth power of the body's absolute temperature. The emissions from 'gray' bodies can be approximated by this law.

The emissions by other bodies are treated in an empirical manner, relying on in particular the coefficient of emission (emissivity), which is determined by measurements.

A body that does not absorb all incident radiation (sometimes known as a gray body) emits less total energy than a black body and is characterized by an emissivity, so the emitted energy flux (intensity) is: ($J{\cdot}s^{-1}{\cdot}m^{-2}$ or $W{\cdot}m^{-2}$) where is the Stefan–Boltzmann constant and is absolute temperature. The emissivity depends on the wavelength, The radiation pressure on an emitting surface by emitted radiation is then: ($N{\cdot}m^{-2}$ or Pa)

Then the equation of the $P_{emission}$ is:

$$p_{emission} = \frac{\epsilon\sigma}{c} T^4 \qquad (3.12)$$

As the gasoline combustion products are in gas phase the term viscosity is enter during the process, the last is the lifetime, and we need to calculate the viscosity of the gas to calculate it, and the equation of viscosity:

Viscosity of Gases: η

$$\eta = \eta_0 + \alpha t - \beta t^2 \qquad (3.13)$$

Where η is the viscosity of the gas at t ^{o}C in poise, η_0 is the viscosity of the gas at 0 ^{o}C in poise, and *t = temperature,* and α *and* β are constants; $\alpha = 0.56 \times 10^{-7}$ and $\beta = 0.1189 \times 10^{-9}$

Table (3.1) shows the physical properties of the products of the gasoline combustion which needed to be studied via LIF modeled using Python language.

Table (3.1): Physical properties of the gasoline combustion products:

Name	For mula	Molar mass	Viscosity of gas at t c poise	Radius (m)
Carbon dioxide	CO_2	44.0095	0.00013830126	165×10^{-12}
Carbon monoxide	CO	28.0101	0.00016483165	188×10^{-12}
Nitrogen	N_2	28.0134	0.0001781	132×10^{-12}
Nitric oxide	NO	30.0061	0.00011584498	158×10^{-12}
Nitrogen dioxide	NO_2	46.005	0.00010254	151×10^{-12}
water	H_2O	18.0153	0.00016531187	132.5×10^{-12}
Hydrocarbon	HC	13.0186	0.00012025	146×10^{-12}

Table (3.2) shows possible depopulation pathways of excited fluorescence

Process	possible photo bleaching pathway	symbol	*Time scale*
Internal conversion	$s_n \to s_1 , T_n \ \to \ T_1$	K_{ic}	$10^{10} - \ 10^{14}\ s^{-1}$
Internal conversion	$s_1 \to s_0$	K_{ic}	$10^{6} - \ 10^{7}\ s^{-1}$
Vibrational relaxation	$s_{1,v=n} \to s_{1,v=0}$	K_{vr}	$10^{10} - \ 10^{12}\ s^{-1}$
singlet–singlet absorption	$s_1 \to s_n$	K_{exc}	$10^{15}\ s^{-1}$
Fluorescence	$s_1 \to s_0$,	K_f	$10^{7} - \ 10^{9}\ s^{-1}$
Intersystem crossing	$s_1 \to T_1 , s_n \ \to \ T_n, T_n \ \to \ s_n$	K_{isc}	$10^{5} - \ 10^{8}\ s^{-1}$
Phosphorescence	$T_1 \to S_0$	K_p	$10^{-2} - \ 10^{3}\ s^{-1}$
Triplet–triplet absorption	$T_1 \ \to \ T_n$	K_{exc}	$10^{15}\ s^{-1}$

$$K_{diff} = \frac{8RT}{3\eta} \qquad (3\ .14)$$

$$K_{diff} \equiv diffusion\, rate$$

$$R\ \equiv universal\ gas\ constant = 8.31/mol\_K$$

$$K_q\ \approx\ K_{diff} \qquad (3\ .15)$$

$$K_q = \frac{8RT}{3\eta} \qquad (3\ .16)$$

$$\tau_F = \left[\, K_F + \ K_{ic} + \ K_{is} + K_q\ \right]^{-1} \qquad (\,3\,.17\,)$$

3.4 Description of LIF program:

The flow chart that describes the operation of the LIF program developed using Python language and is build according to the equations (from 3.1 to 3.15) in addition to the basics instructions of the python programs.

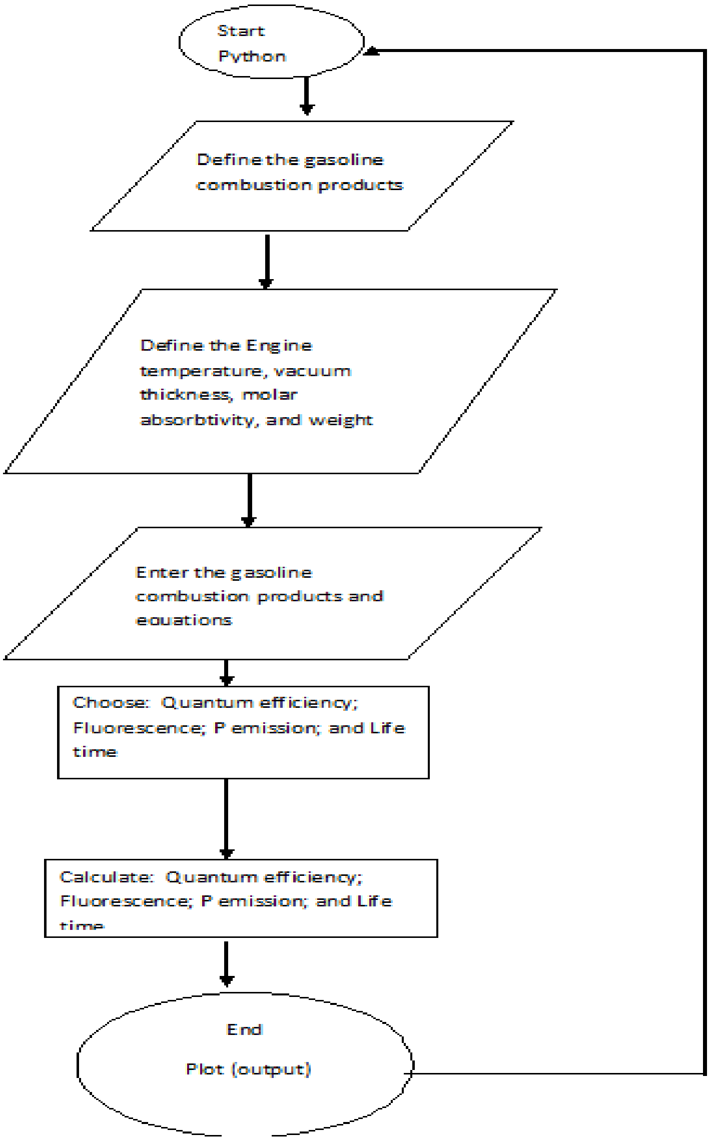

Start
Python
Define the gasoline combustion products
Define the Engine temperature, vacuum thickness, molar absorbtivity, and weight
Enter the gasoline combustion products and equations
Choose: Quantum efficiency; Fluorescence; P emission; and Life time
Calculate: Quantum efficiency; Fluorescence; P emission; and Life time
End
Plot (output)

Figure (3.1) shows the LIF software developed user interface.

Figure (3.2) shows the starting menu of the LIF software built using the Python language.

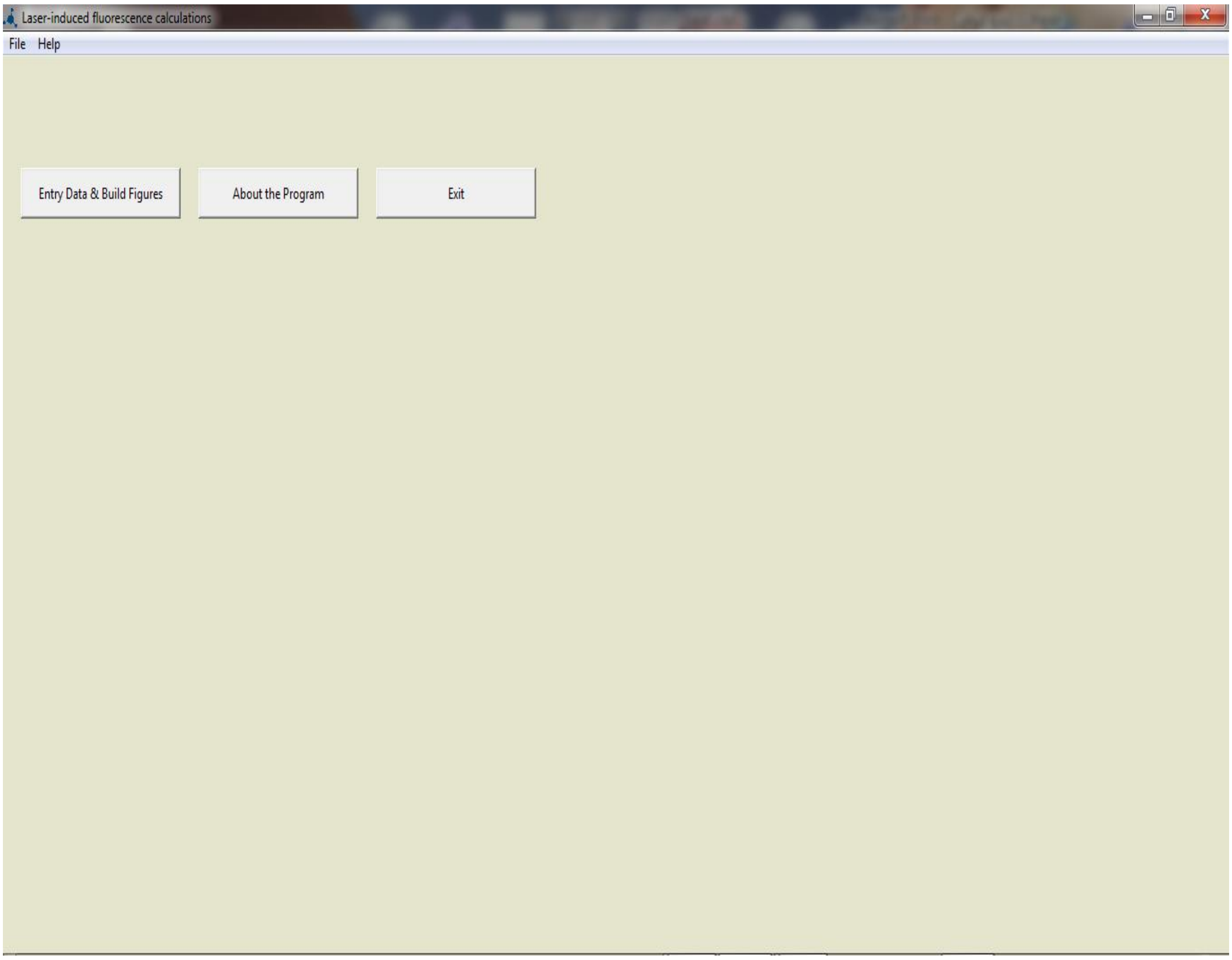
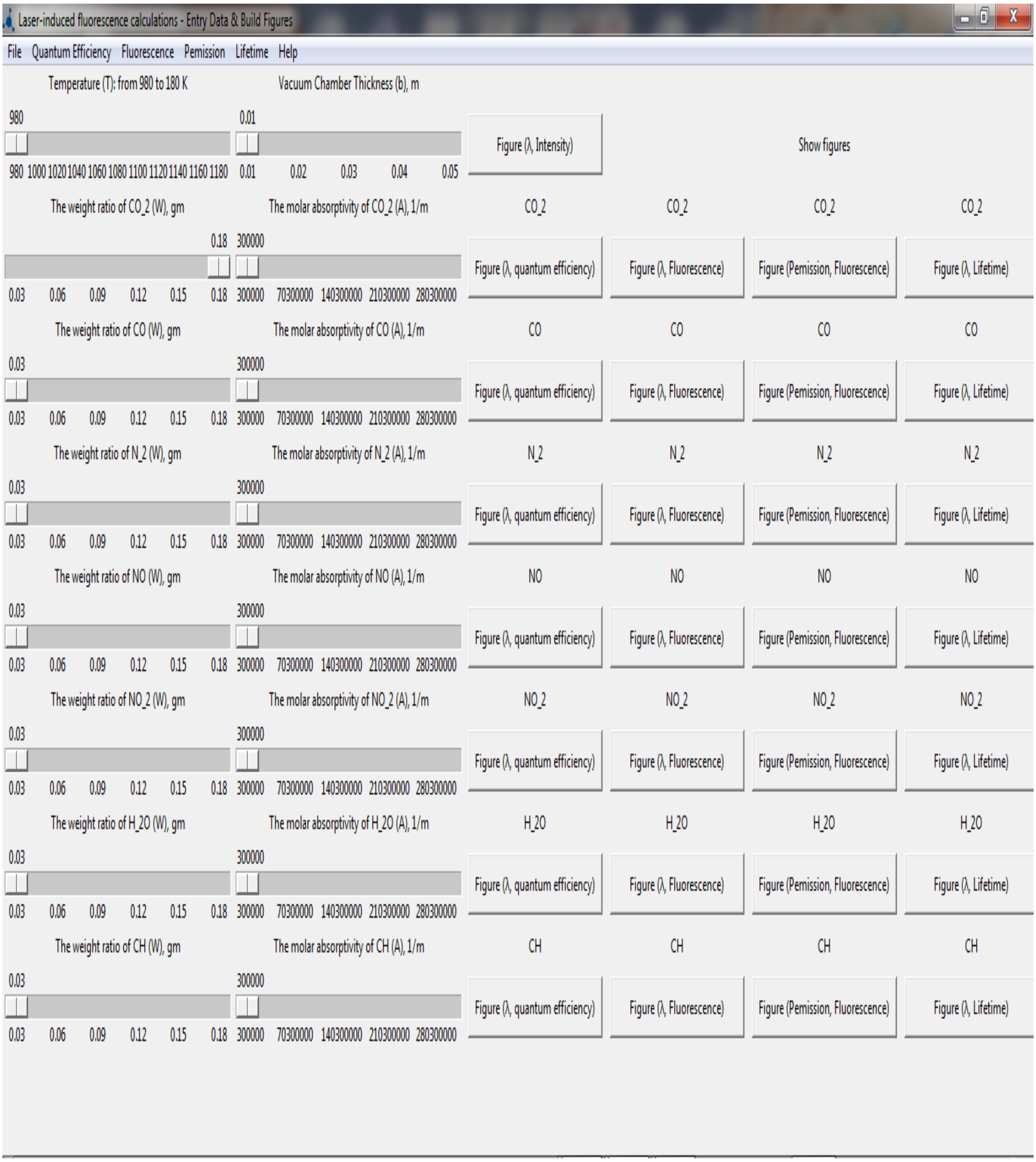

Laser-induced fluorescence calculations
File
Help
Entry Data & Build Figures
About the Program
Exit

Figure (3.2) shows the starting menu LIF software developed using PYTHON language.

Figure (3.3), illustrates the window of data entry and build the figures of the LIF software developed using the high level PYTHON language in which the combustion products of the gasoline combustion products can be studied and the description of using the LIF software to carry out the results about the gasoline combustion products (seven molecules : CO, CO_2, NO, NO, N_2, H_2O, and HC) can be studied in a number of physical parameters described shortly in method section 3.5 of this chapter.



Temperature (T): from 980 to 180 K: 980

Vacuum Chamber Thickness (b), m: 0.01

The weight ratio of CO_2 (W), gm: 0.18

The molar absorptivity of CO_2 (A), 1/m: 300000

The weight ratio of CO (W), gm: 0.03

The molar absorptivity of CO (A), 1/m: 300000

The weight ratio of N_2 (W), gm: 0.03

The molar absorptivity of N_2 (A), 1/m: 300000

The weight ratio of NO (W), gm: 0.03

The molar absorptivity of NO (A), 1/m: 300000

The weight ratio of NO_2 (W), gm: 0.03

The molar absorptivity of NO_2 (A), 1/m: 300000

The weight ratio of H_2O (W), gm: 0.03

The molar absorptivity of H_2O (A), 1/m: 300000

The weight ratio of CH (W), gm: 0.03

The molar absorptivity of CH (A), 1/m: 300000

Figure (3.3): the window of the data entry and build the figures of the LIF developed software

3.5 Method:

The methodology of recording the laser induced fluorescence of the gasoline combustion products are done as flows:

As shown in figure (3.2) first is to open the software through the Entry Data and Figures icon, second choose one of either quantum efficiency, fluorescence, Pemission or life time to used in the recording the figures(inside each there are the possible seven gasoline combustion molecules) we chose one them to be studied (for example: Use quantum efficiency then a sub window that contains the seven molecules will be opened then chose carbon dioxide after that you can vary all of the following four parameters, the four parameters are:

1. Temperature can be varied from 980 to 1180 Kelvin

2. Vacuum chamber can be varied from 1 cm to 5 cm.

3. The weight ratio which can be varied from 0.03 to 0.18 gm

4. The Molar absorptivity which according to the types of the gasoline combustion products as shown in table (3.1) can be varied from

3×10^5 to 3×10^8 $M^{-1}\, cm^{-1}$

This parameters can be adjusted for the seven molecules separately, that is to say Carbon dioxide can be used and the four above parameters are applicable, similarly one can use the above four varied parameters with the all other six molecules (gasoline combustion products).On the right hand side of the software window as shown in figure (3.3) you can use the figures icons for your entry molecule information then you can generate the plots (quantum efficiency, fluorescence, Pemission, lifetime) for the molecule under study

(as given in the example carbon dioxide).The software also provides figure of the fluorescence intensity for the seven molecules which can be studied as a result of changing the four parameters (1, 2, 3, and 4). Followed the above method each gasoline combustion product was investigated, then the results obtained were analyzed and discussed.

Chapter four

Results and Discussion

4.1 Introduction:

In this chapter we presents and discussed the results of gasoline exhaust combustion products (CO_2, CO, N_2, NO, NO_2, HC, and H_2O) using laser induced fluorescence technique modeled using Python language.

The study using many physical parameters of the gasoline products such as quantum efficiency, fluorescence life time, fluorescence intensity, and P emission, in addition to the effect of the temperature.

4.2 Results of carbon dioxide (CO_2):

Carbon dioxide is the most commonly encountered gasoline combustion products, figure (4.1) shows the results of quantum efficiency against the laser wavelengths as a function of the temperature (the vacuum chamber thickness and the molecular weight were kept fixed to 1 cm, and 0.03 gm respectively).

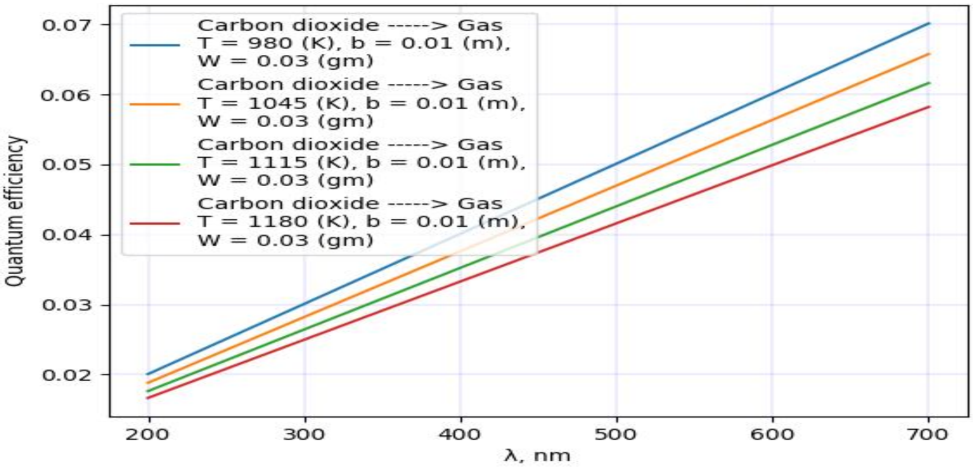


Figure (4.1): Quantum efficiency of CO_2 at different temperatures with laser wavelength

Figure (4.1) showed that the relationship between the quantum efficiency and the laser wavelengths of carbon dioxide (CO_2) as a function of temperature. When the laser wavelength increased from 200 nm to 700 nm the quantum efficiency also increases as shown by the colour lines in the figure. The blue line represent the quantum efficiency with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.020 and it became 0.07 at 700 nm . While the Orange line represent the quantum efficiency with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the quantum efficiency was 0.018 and it become 0.066 at 700 nm. The quantum efficiency at 200 nm was 0.017 and 0.061 at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the quantum efficiency of the carbon dioxide was 0.016 at 200 nm wavelength and it increased to 0.058 at 700 nm wavelength of the laser source. Figure (4.2) shows effect of temperature on lifetime of CO_2 at different temperatures.

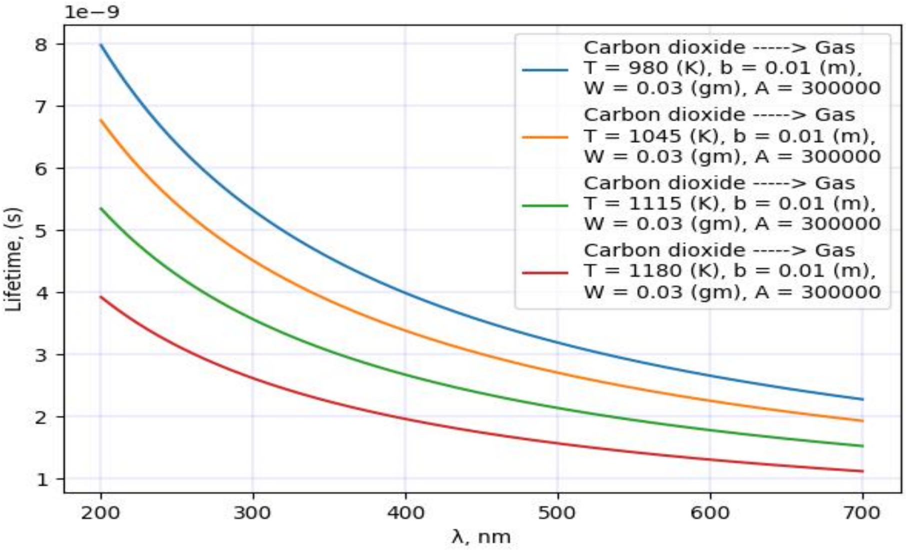


Figure (4.2): The effect of temperature on lifetime of CO_2

Figure (4.2) showed that the relationship between the lifetime and the laser wavelengths of carbon dioxide (CO_2) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the lifetime decreases as shown by the colour lines in the figure. The blue line represent the lifetime with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the lifetime was 8×10^{-9} s and it became 2.31×10^{-9} s at 700 nm . While the Orange line represent the lifetime with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the lifetime was 6.8×10^{-9}s and it become 1.9×10^{-9} s at 700 nm. The lifetime at 200 nm was 5.3×10^{-9} s and 1.5×10^{-9} s at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the lifetime of the carbon dioxide was 3.9×10^{-9} s at 200 nm wavelength and it increased to 1.1×10^{-9} s at 700 nm wavelength. Figure (4.3) shows the effect of weight on fluorescence of carbon dioxide at different weight.

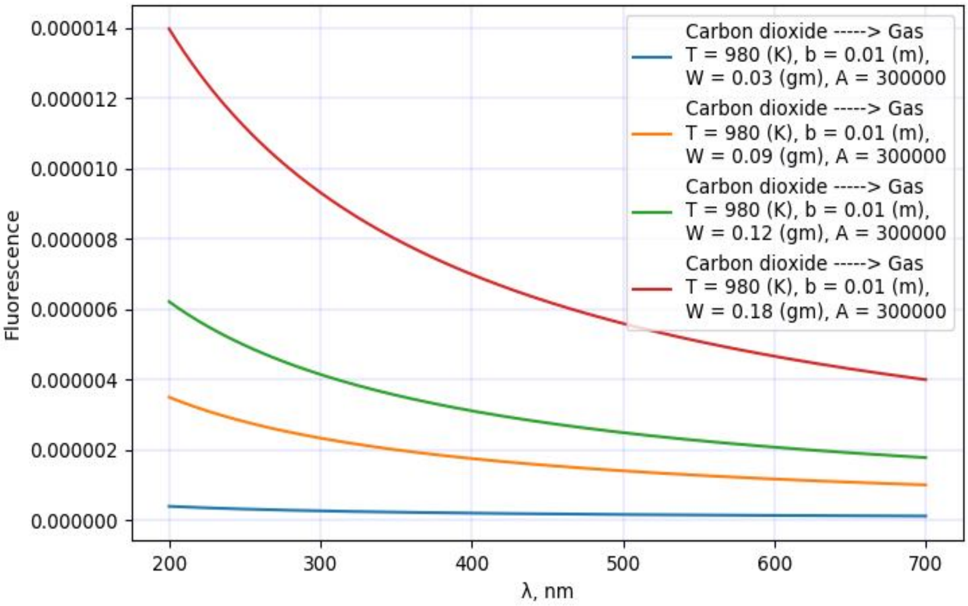


Figure (4.3): The effect of weight on fluorescence of carbon dioxide

Figure (4.3) showed that the relationship between the fluorescence and the laser wavelengths of carbon dioxide (CO_2) as a function of weight. When the laser wavelengths increased from 200 nm to 700 nm the fluorescence decreases as shown by the colour lines in the figure. The blue line represents the fluorescence with the laser wavelengths when the weight was 0.03 gm at 200 nm wavelength the fluorescence was 3.75×10^{-7} and it became 1.27×10^{-7} at 700 nm. While the Orange line represents the fluorescence with the laser wavelengths when the weight was 0.09 gm at 200 nm wavelength the fluorescence was 3.5×10^{-6} and it become 1.03×10^{-6} at 700 nm. The fluorescence at 200 nm was 6.3×10^{-6} and 1.82×10^{-6} at 700 nm when the temperature was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the carbon dioxide was 1.4×10^{-5} at 200 nm wavelength and it decreased to 4×10^{-6} at 700 nm wavelength. As shown in figure (4.3) it's clearly that the fluorescence increased as the weight increased.

The effect of weight on Pemission of carbon dioxide at weights of 0.03 gm to 0.18 gm is shown in figure (4.4).

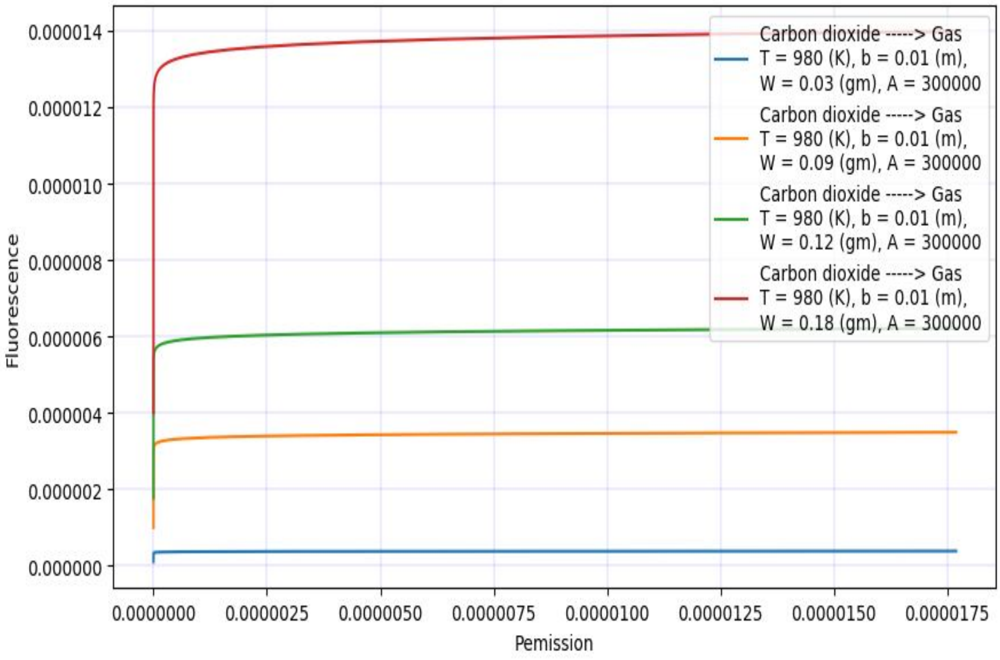


Figure (4.4): The effect of weight on Pemission of carbon dioxide at weights of 0.03 gm to 0.18 gm

Figure (4.4) showed that the relationship between the fluorescence and the laser Pemission of carbon dioxide (CO_2) as a function of weight. When the weight increased from 0.03 gm to 0.18 gm the fluorescence increases and also pemission increases as shown by the colour lines in the figure. The blue line represents the fluorescence with pemission when the weight was 0.03 gm the fluorescence was 8.6×10^{-8} pemission was 2.14×10^{-8} . While the Orange line represent when the weight was 0.09 gm the fluorescence was 9.9×10^{-7} Pemission was 2.14×10^{-8}. The fluorescence was 1.73×10^{-6} and Pemission was 2.14×10^{-8} when the weight was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the carbon dioxide was 4×10^{-6} and Pemission was 2.14×10^{-8}. According to figure (4.4) the fluorescence increased as the weight increased.

4.3: Results of carbon monoxide (CO):

Investigations of gasoline combustion emissions (pollutants) from vehicles confirmed CO_2, Hydrocarbon (HC), Nitrogen oxides (NOx) and CO as major pollutants, with the CO higher contribution.

Figure (4.5) shows the results of quantum efficiency against the laser wavelengths as' a function of the temperature (the vacuum chamber thickness and the molecular weight were kept fixed to 1 cm and 0.03 gm respectively).

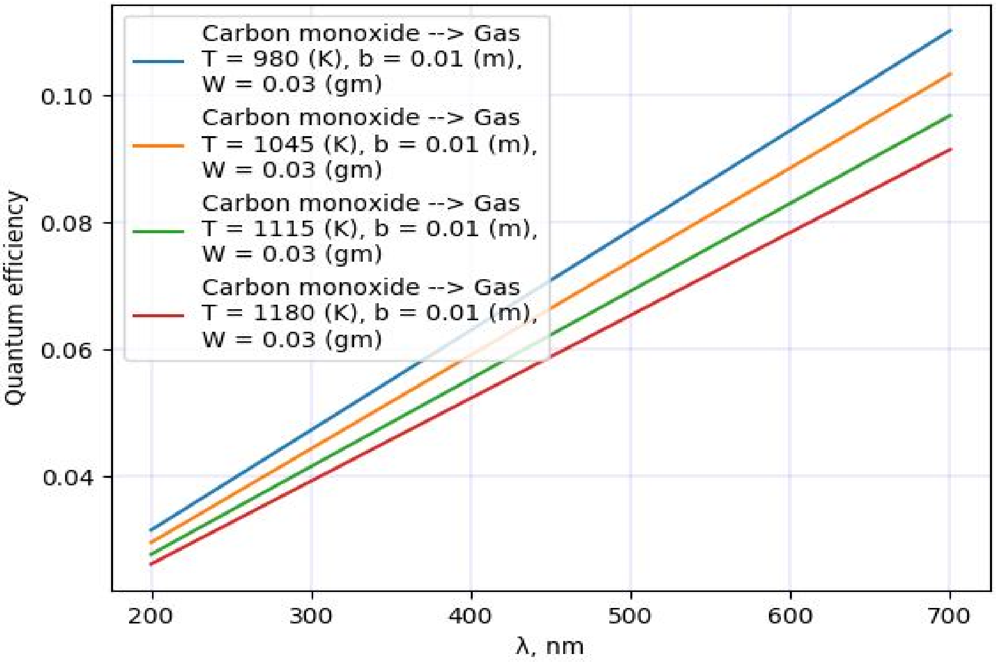


Figure (4.5) effect of temperature on the Quantum efficiency of carbon monoxide

Figure (4.5) showed that the relationship between the quantum efficiency and the laser wavelengths of carbon monoxide (CO) as a function of temperature. As shown that when the laser wavelengths increased from 200 nm to 700 nm the quantum efficiency also increases as shown by the colour lines in the figure. The blue line represent the quantum efficiency with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.031 and it became 0.11 at 700 nm . While the Orange line represent the quantum efficiency with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the quantum efficiency was 0.029 and it become 0.1 at 700 nm. The quantum efficiency at 200 nm was 0.027 and 0.097 at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the quantum efficiency of the carbon monoxide was 0.025 at 200 nm wavelength and it increased to 0.091 at 700 nm wavelength.

Figure (4.6) depicts the effect of temperature on the Lifetime of carbon monoxide at a temperature of 980 to 1180 Kelvin

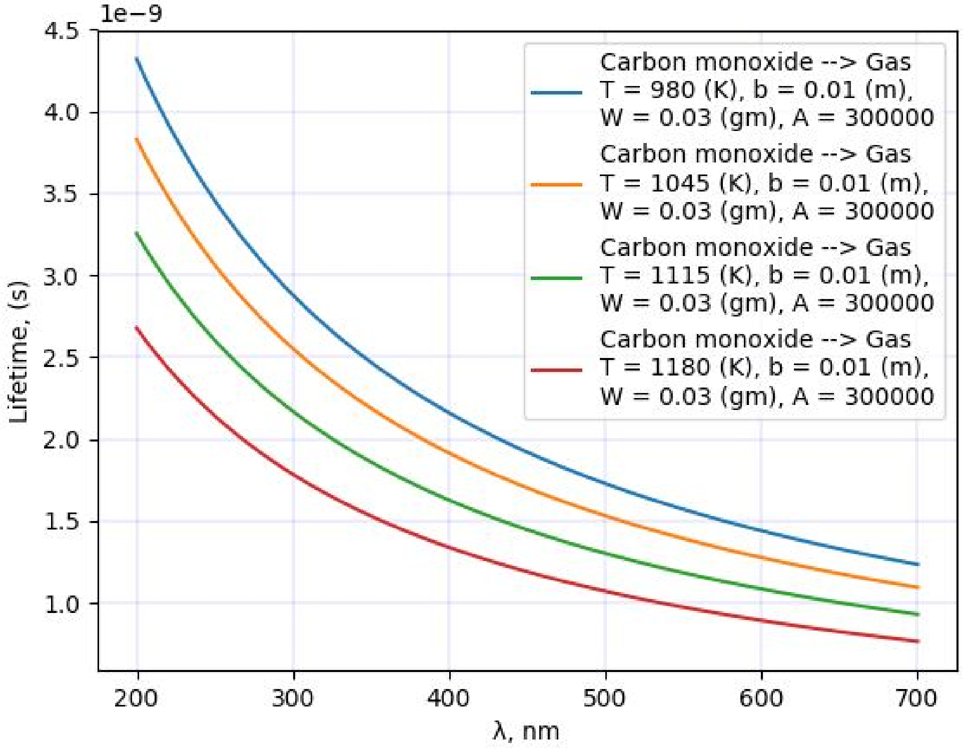


Figure (4.6) the effect of temperature on the Lifetime of carbon monoxide

Figure (4.6) showed that the relationship between the lifetime and the laser wavelengths of carbon monoxide (CO) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the lifetime decreases as shown by the colour lines in the figure. The blue line represent the lifetime with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the lifetime was 4.33×10^{-9} and it became 1.26×10^{-9} at 700 nm. While the Orange line represent the lifetime with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the lifetime was $3.8 \times 10^{-9}$and it became 1.1×10^{-9} at 700 nm. The lifetime at 200 nm was 3.3×10^{-9} and 9.5×10^{-10} at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the lifetime of the carbon monoxide was 2.7×10^{-10} at 200 nm wavelength and it increased to 8×10^{-10} at 700 nm wavelength.

The effect of the weight on the fluorescence of the carbon monoxide is shown in figure (4.7)

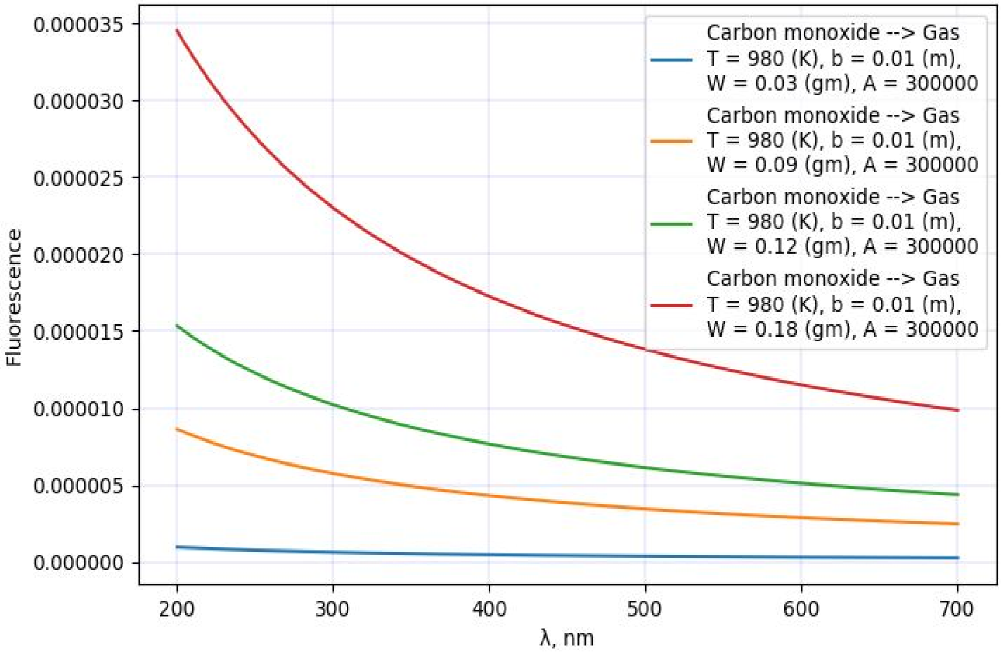


Figure (4.7) the effect of weight on fluorescence of carbon monoxide

Figure (4.7) showed that the relationship between the fluorescence and the laser wavelengths of carbon monoxide (CO) as a function of weight. When the laser wavelengths increased from 200 nm to 700 nm the fluorescence decreases as shown by the colour lines in the figure. The blue line represents the fluorescence with the laser wavelengths when the weight was 0.03 gm at 200 nm wavelength the fluorescence was 1.04×10^{-6} and it became 3.9×10^{-7} at 700 nm. While the Orange line represents the fluorescence with the laser wavelengths when the weight was 0.09 gm at 200 nm wavelength the fluorescence was 8.7×10^{-6} and it become 2.5×10^{-6} at 700 nm. The fluorescence at 200 nm was 1.5×10^{-5} and 4.5×10^{-6} at 700 nm when the temperature was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the carbon monoxide was 3.5×10^{-5} at 200 nm wavelength and it decreased to 9.9×10^{-6} at 700 nm wavelength of the laser used. Figure (4.8) shows the effect of weight on Pemission of carbon monoxide at different weights.

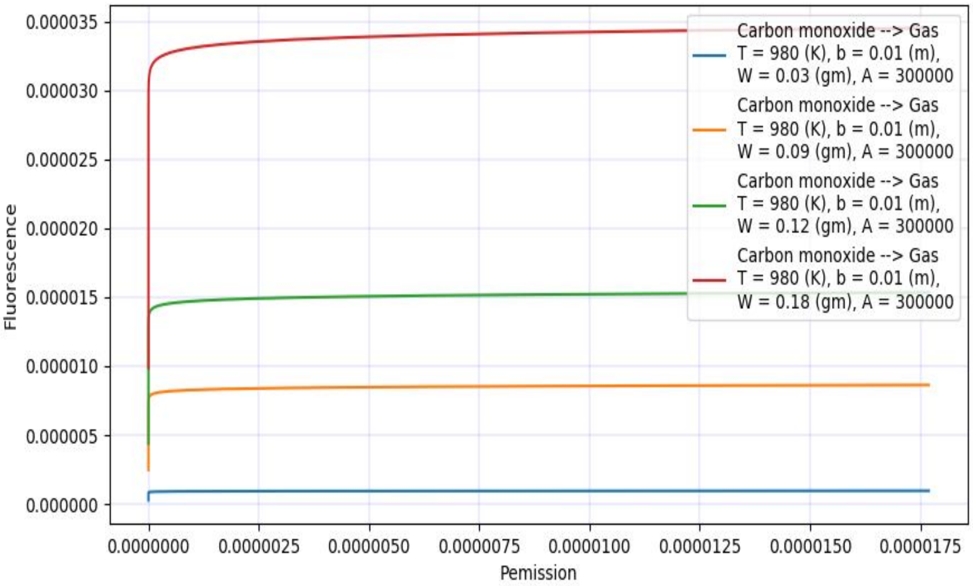


Figure (4.8) the effect of weight on Pemission of carbon monoxide

Figure (4.8) showed that the relationship between the fluorescence and the laser Pemission of carbon monoxide (CO) as a function of weight and thus the fluorescence increased as the weight increased. When the weight increased from 0.03 gm to 0.18 gm the fluorescence increases and also pemission increases as shown by the colour lines in the figure. The blue line represents the fluorescence with pemission when the weight was 0.03 gm the fluorescence was 3.1×10^{-7} pemission was 1.6×10^{-8} . While the Orange line represent when the weight was 0.09 gm the fluorescence was 2.4×10^{-6} Pemission was 1.6×10^{-8}. The fluorescence was 4.4×10^{-6} and pemission was 1.6×10^{-8} when the weight was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the carbon monoxide was 9.9×10^{-8} and pemission was

1.6×10^{-8}. As predicted from the results shown in figure (4.8) the fluorescence increased as the weight increased.

4.4 Results of Nitrogen (N_2):

Figure (4.9) shows the results of quantum efficiency against the laser wavelengths as a function of the temperature of Nitrogen combustion product (the vacuum chamber thickness and the molecular weight were kept fixed to 1 cm and 0.03 gm respectively).

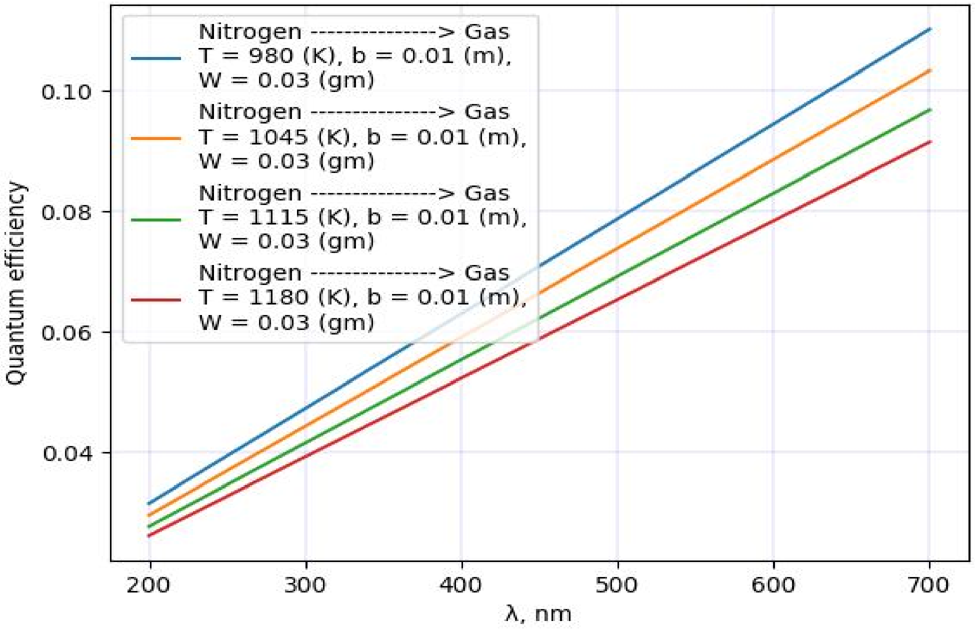


Figure (4.9) the effect of temperature on the Quantum efficiency of Nitrogen (N_2)

Figure (4.9) showed that the relationship between the quantum efficiency and the laser wavelengths of Nitrogen (N_2) as a function of temperature. As laser wavelength increased from 200 nm to 700 nm the quantum efficiency also increases as shown by the colour lines in the figure. The blue line represent the quantum efficiency with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.031 and it became 0.11

at 700 nm . While the Orange line represent the quantum efficiency with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the quantum efficiency was 0.029 and it become 0.1 at 700 nm. The quantum efficiency at 200 nm was 0.027 and 0.097 at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the quantum efficiency of the Nitrogen was 0.025 at 200 nm wavelength and it increased to 0.091 at wavelength of 700 nm.

Figure (4.10) shows the effect of temperature on the Lifetime of Nitrogen.

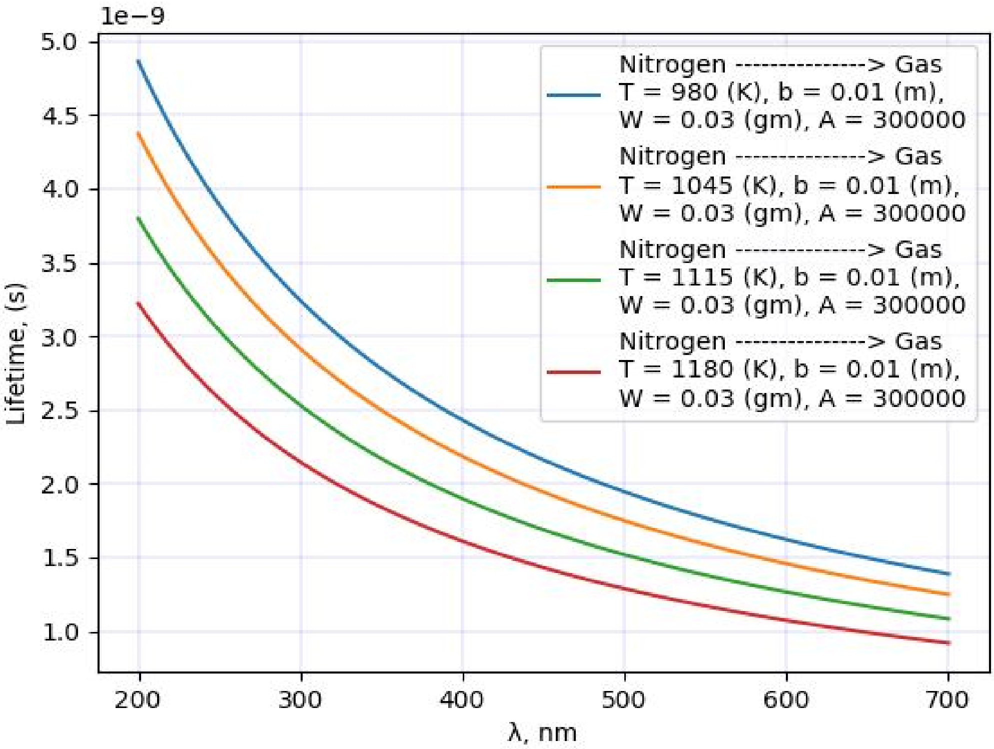


Figure (4.10) the effect of temperature on the Lifetime of Nitrogen

Figure (4 .10) showed that the relationship between the lifetime and the laser wavelengths of Nitrogen (N_2) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the lifetime decreases as shown by the colour lines in the figure. The blue line represent the lifetime with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the lifetime was 4.8×10^{-9} and it became 1.4×10^{-9} at 700 nm. While the Orange line represent the lifetime with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the lifetime was $4.4 \times 10^{-9}$and it became 1.3×10^{-9} at 700 nm. The lifetime at 200 nm was 3.8×10^{-9} and 1.1×10^{-9} at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the lifetime of the Nitrogen was 3.2×10^{-10} at 200 nm wavelength and it increased to 9.3×10^{-10} at 700 nm wavelength

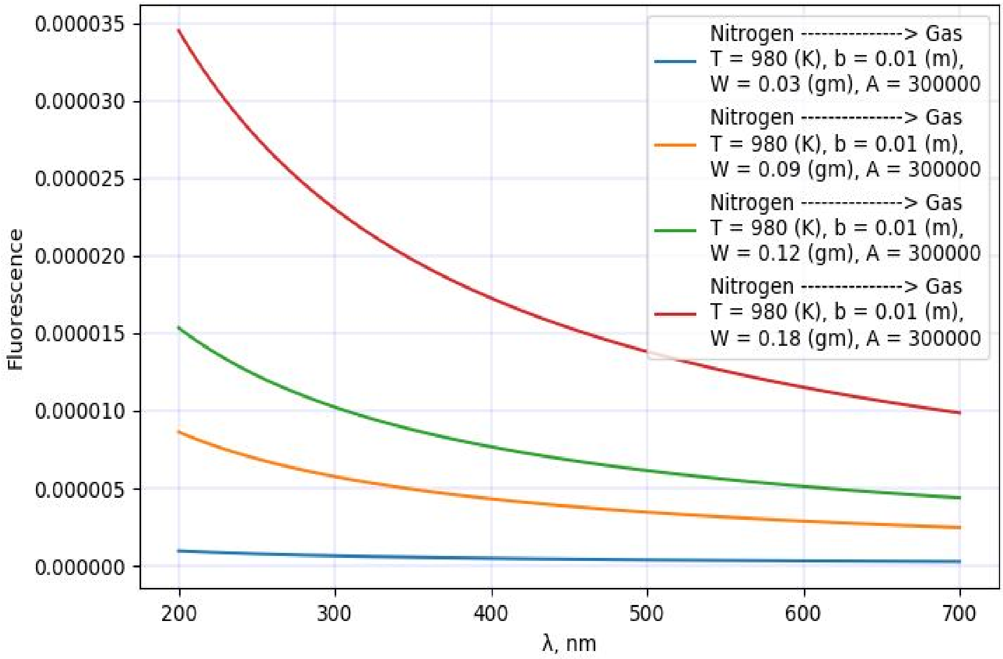


Figure (4.11) the effect of weight on fluorescence of Nitrogen

Figure (4.11) showed that the relationship between the fluorescence and the laser wavelengths of Nitrogen (N_2) as a function of weight. The fluorescence decreases when the wavelength increased from 200 nm to 700 nm as shown by the colour lines in the figure. The blue line represents the fluorescence with the laser wavelengths when the weight was 0.03 gm at 200 nm wavelength the fluorescence was 1.1×10^{-6} and it became 4.2×10^{-7} at 700 nm. While the Orange line represents the fluorescence with the laser wavelengths when the weight was 0.09 gm at 200 nm wavelength the fluorescence was 8.7×10^{-6} and it become 2.6×10^{-6} at 700 nm. The fluorescence at 200 nm was 1.5×10^{-5} and 4.4×10^{-6} at 700 nm when the temperature was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Nitrogen was 3.4×10^{-5} at 200 nm wavelength and it decreased to 1×10^{-5} at 700 nm wavelength. As shown in figure (4.11) it's easy understood that the fluorescence increased as the weight increased.

Figure (4.12) shows effect of weight on Pemission of Nitrogen.

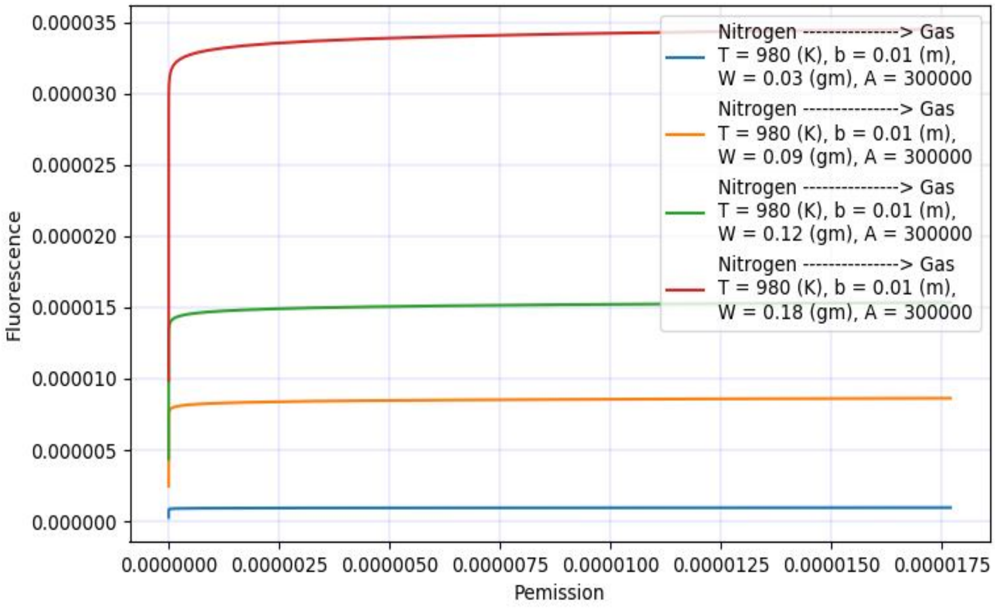


Figure (4.12) the effect of weight on Pemission of Nitrogen

Figure (4.12) showed that the relationship between the fluorescence and the laser Pemission of Nitrogen (N_2) as a function of weight. The fluorescence increases when the weight increases from 0.03 gm to 0.18 gm and also Pemission increases as shown by the colour lines in the figure. The blue line represents the fluorescence with pemission when the weight was 0.03 gm the fluorescence was

4.18×10^{-7} pemission was 2.14×10^{-8}. While the Orange line represent when the weight was 0.09 gm the fluorescence was 2.45×10^{-6} pemission was 2.14×10^{-8}. The fluorescence was 4.4×10^{-6} and pemission was 4.49×10^{-6} when the weight was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Nitrogen was 9.9×10^{-6} and pemission was 2.14×10^{-8}, it can be said that the fluorescence increased as the weight increased.

4.5 Results of Nitric oxide (NO):

Figure (4.13) shows the results of quantum efficiency against the laser wavelengths as a function of the temperature of the Nitric oxide gasoline combustion product (the vacuum chamber thickness and the molecular weight were kept fixed to 1 cm and 0.03 gm respectively).

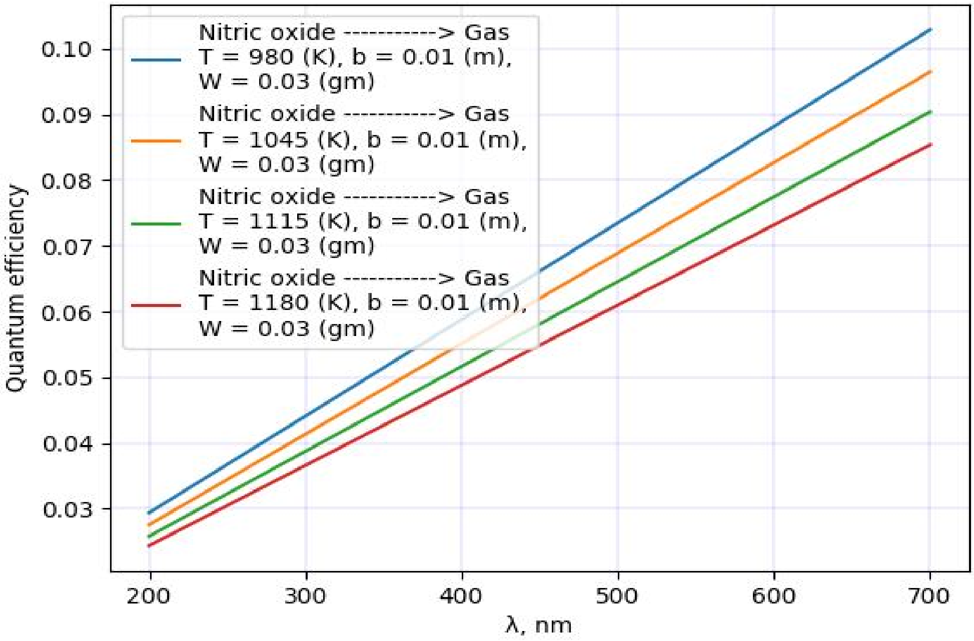


Figure (4.13): Quantum efficiency of CO at different temperatures with laser wavelength

The relationship between the quantum efficiency and the laser wavelengths of carbon monoxide (CO) as a function of temperature as shown in figure (4 .13), when the laser wavelengths increased from 200 nm to 700 nm the quantum efficiency also increases as shown by the colour lines in the figure. The blue line represent the quantum efficiency with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.029 and it became 0.1 at 700 nm . While the Orange line represent the quantum efficiency with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the quantum efficiency was 0.027 and it become 0.096 at 700 nm. The quantum efficiency at 200 nm was 0.025 and 0.09 at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the quantum efficiency of the carbon monoxide was 0.02 at a wavelength of 200 nm wavelength and it increased to 0.085 at a wavelength of 700 nm.

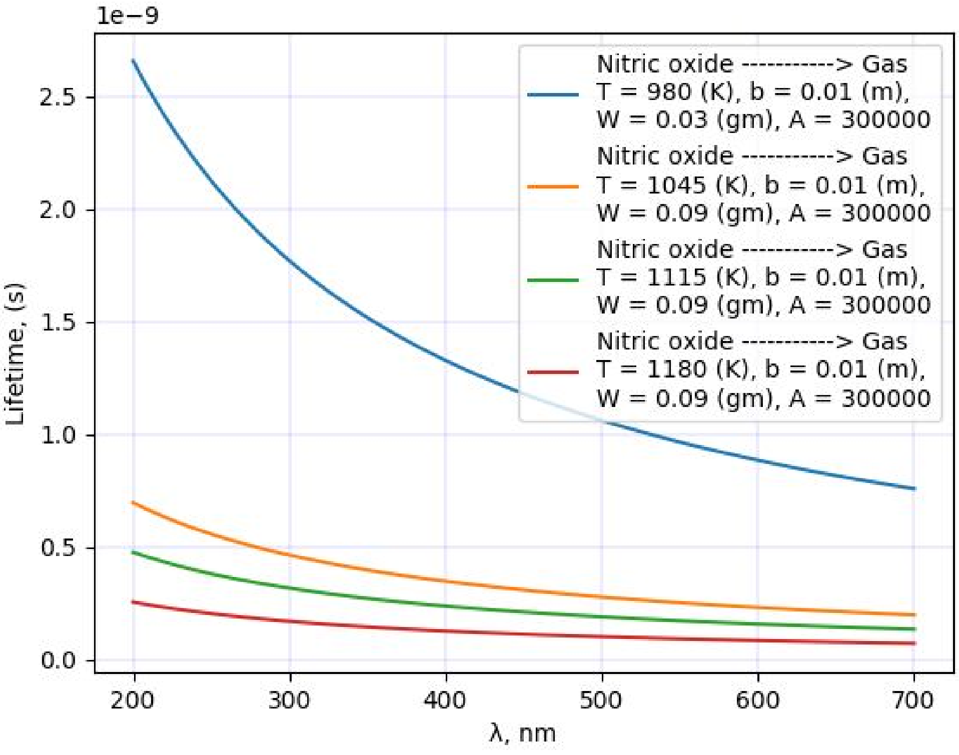


Figure (4.14): The effect of temperature on lifetime of Nitric oxide (NO)

Figure (4.14) showed that the relationship between the lifetime and the laser wavelengths of Nitric oxide (NO) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the lifetime decreases as shown by the colour lines in the figure. The blue line represent the lifetime with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the lifetime was 7.7×10^{-10} s and it became 2.7×10^{-9} s at 700 nm . While the Orange line represent the lifetime with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the lifetime was 7.06×10^{-10} s and it became 2.1×10^{-10} s at 700 nm. The lifetime at 200 nm was 4.9×10^{-10} s and 1.5×10^{-10} s at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the lifetime of the Nitric oxide was 2.7×10^{-10} s at 200 nm wavelength and it decreased to 8.4×10^{-11} s at 700 nm wavelength

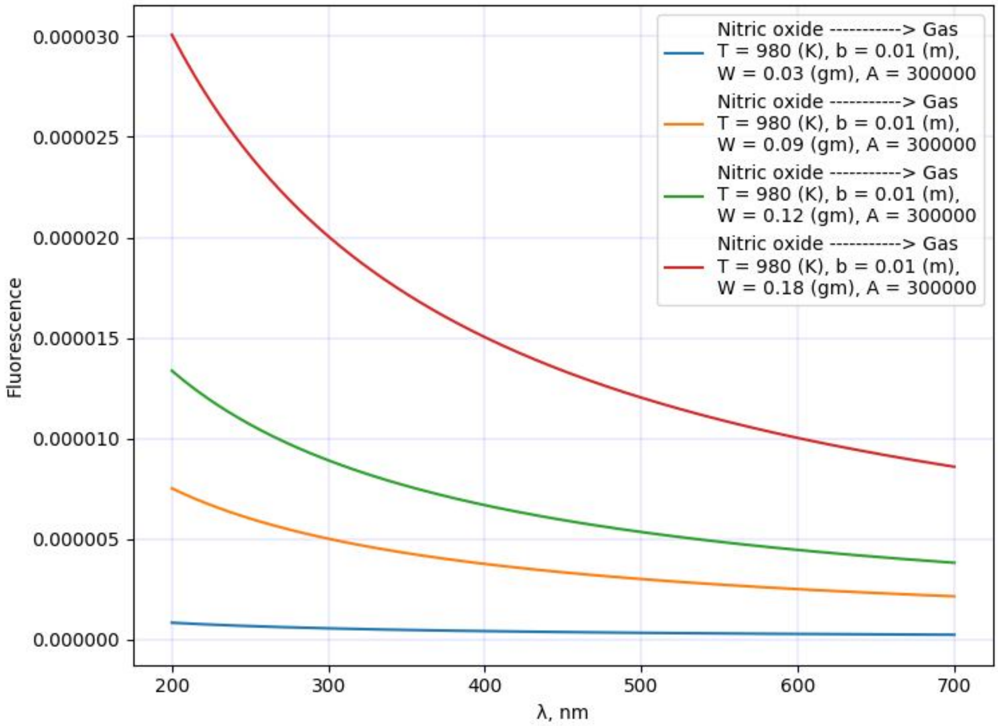


Figure (4.15) the effect of weight on fluorescence of Nitric oxide

Figure (4.15) showed that the relationship between the fluorescence and the laser wavelengths of Nitric oxide (NO) as a function of weight. When the laser wavelengths increased from 200 nm to 700 nm the fluorescence decreases as shown by the colour lines in the figure. The line represents the fluorescence with the laser wavelengths when the weight was 0.03 gm at 200 nm wavelength the fluorescence was 9.5×10^{-6} and it became 3.8×10^{-7} at 700 nm. While the Orange line represents the fluorescence with the laser wavelengths when the weight was 0.09 gm at 200 nm wavelength the fluorescence was 7.5×10^{-6} and it become 2.2×10^{-6} at 700 nm. The fluorescence at 200 nm was 1.3×10^{-5} and 3.9×10^{-6} at 700 nm when the temperature was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Nitric oxide was 3×10^{-5} at 200 nm wavelength and it decreased to 8.7×10^{-6} at 700 nm wavelength. Again the fluorescence increased as the weight increased.

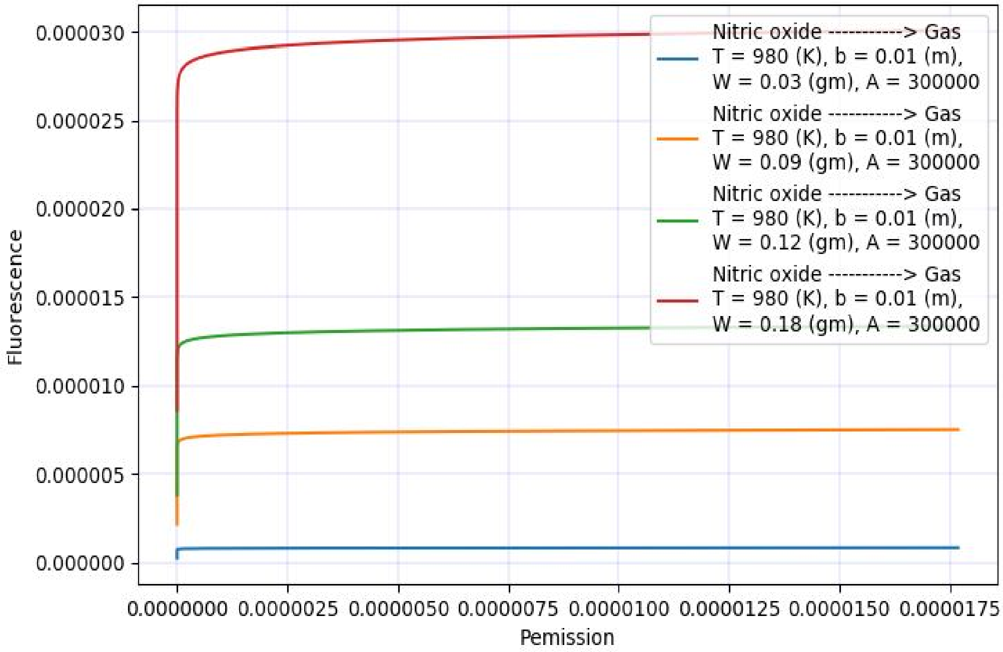


Figure (4.16) the effect of weight on Pemission of Nitric oxide

Figure (4.16) showed that the relationship between the fluorescence and the laser Pemission of Nitric oxide (NO) as a function of weight. The figure showed that as the weight increases from 0.03 gm to 0.18 gm the fluorescence increases and also pemission increases as shown by the colour lines in the figure. The blue line represents the fluorescence with pemission when the weight was 0.03 gm the fluorescence was 1.8×10^{-7} pemission was 4.5×10^{-8} . While the Orange line represent when the weight was 0.09 gm the fluorescence was 2.1×10^{-6} Pemission was 4.5×10^{-8}. The fluorescence was 3.8×10^{-6} and Pemission was 4.5×10^{-8} when the weight was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Nitric oxide was 8.3×10^{-6} and Pemission was 4.5×10^{-8}, the same can be said about the relation of weight and Pemission of Nitric oxide that is increasing the weight results in an increase of the Pemission.

4.6 Results of Nitrogen dioxide (NO_2):

Figure (4.17) shows the results of quantum efficiency against the laser wavelengths as a function of the temperature for the Nitrogen dioxide (the vacuum chamber thickness and the molecular weight were kept fixed to 1 cm and 0.03 gm respectively).

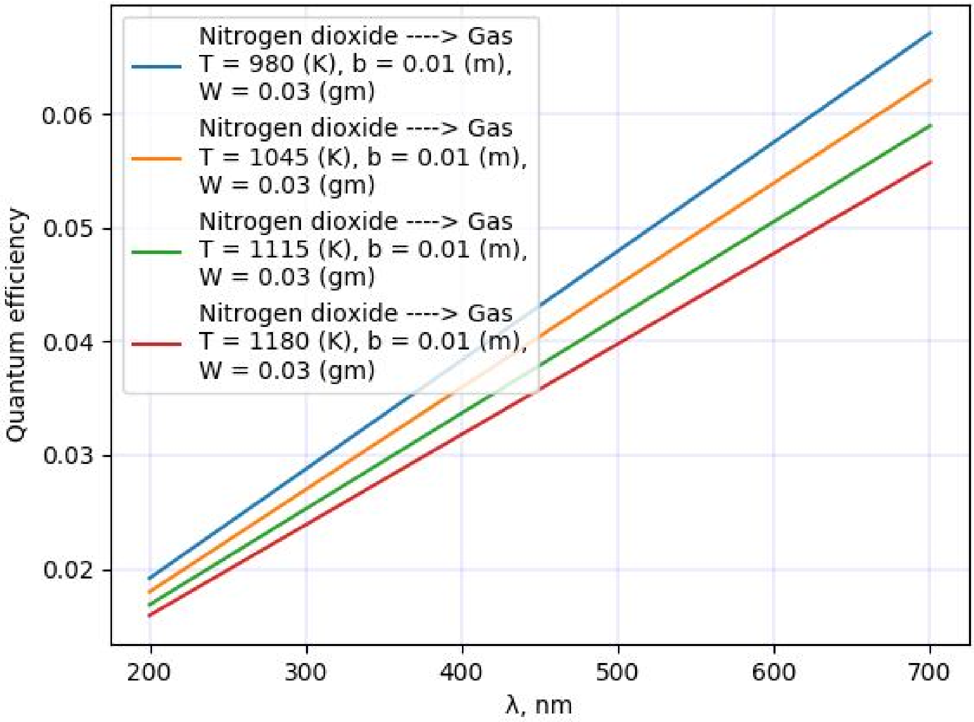


Figure (4 .17) the effect of temperature on the Quantum efficiency of Nitrogen dioxide

Figure (4.17) showed that the relationship between the quantum efficiency and the laser wavelengths of Nitrogen dioxide (NO_2) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the quantum efficiency also increases as shown by the colour lines in the figure. The blue line represent the quantum efficiency with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.019 and it became 0.067 at 700 nm . While the Orange line represent the quantum efficiency with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the quantum efficiency was 0.018 and it become 0.063 at 700 nm. The quantum efficiency at 200 nm was 0.017 and 0.059 at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the quantum efficiency of the Nitrogen dioxide was 0.016 at 200 nm wavelength and it increased to 0.056 at 700 nm wavelength.

Figure (4.18) shows the lifetime of the Nitrogen dioxide as a function of temperature.

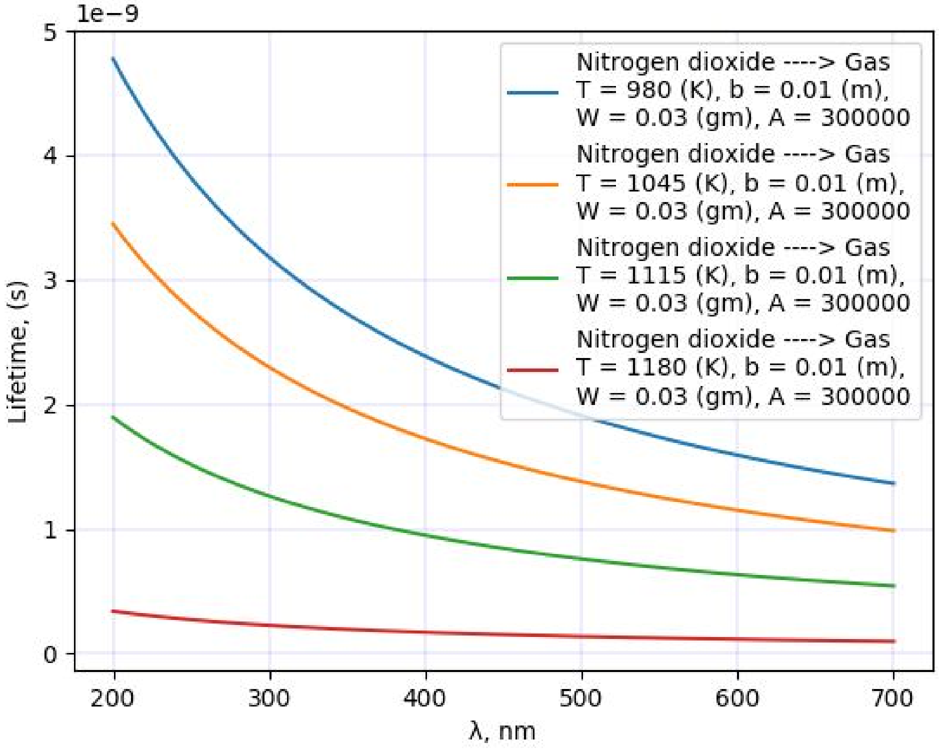


Figure (4 .18) the effect of temperature on lifetime of Nitrogen dioxide

Figure (4 .18) showed that the relationship between the lifetime and the laser wavelengths of Nitrogen dioxide (NO_2) as a function of temperature, as the laser wavelength increased from 200 nm to 700 nm the lifetime decreases as shown by the colour lines in the figure. The blue line represent the lifetime with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the lifetime was 4.8×10^{-9} and it became 1.4×10^{-9} at 700 nm . While the Orange line represent the lifetime with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the lifetime was $3.5 \times 10^{-10}$and it became 9.9×10^{-10} at 700 nm. The lifetime at 200 nm was 1.9×10^{-10} and 5.6×10^{-10} at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the lifetime of the nitrogen dioxide was 3.5×10^{-10} at 200 nm wavelength and it increased to 1.1×10^{-10} at 700 nm wavelength.

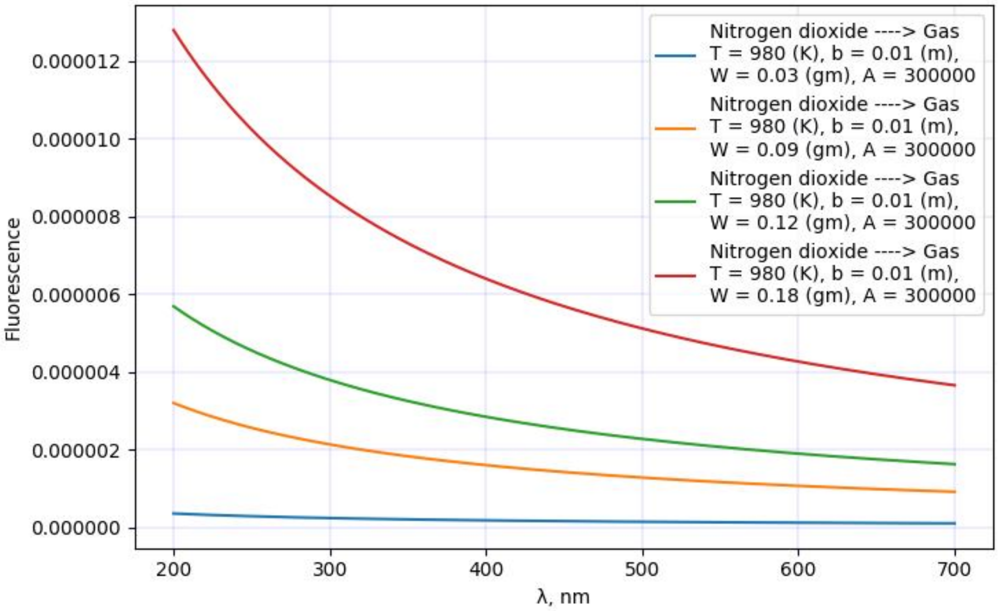


Figure (4.19) the effect of weight on fluorescence of Nitrogen dioxide (NO_2)

The relation of the Nitrogen dioxide (NO_2) as a function of weight shown in figure (4.19), as the laser wavelength increased from 200 nm to 700 nm the fluorescence decreases as shown by the colour lines in the figure. The blue line represents the fluorescence with the laser wavelengths when the weight was 0.03 gm at 200 nm wavelength the fluorescence was 3.8×10^{-7} and it became 1.5×10^{-7} at 700 nm. While the Orange line represents the fluorescence with the laser wavelengths when the weight was 0.09 gm at 200 nm wavelength the fluorescence was 3.2×10^{-6} and it become 9.4×10^{-7} at 700 nm. The fluorescence at 200 nm was 5.7×10^{-5} and 1.6×10^{-6} at 700 nm when the temperature was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Nitrogen dioxide was 1.3×10^{-5} at 200 nm wavelength and it decreased to 3.7×10^{-6} at 700 nm wavelength, also it was shown that the fluorescence with the increase in weight.

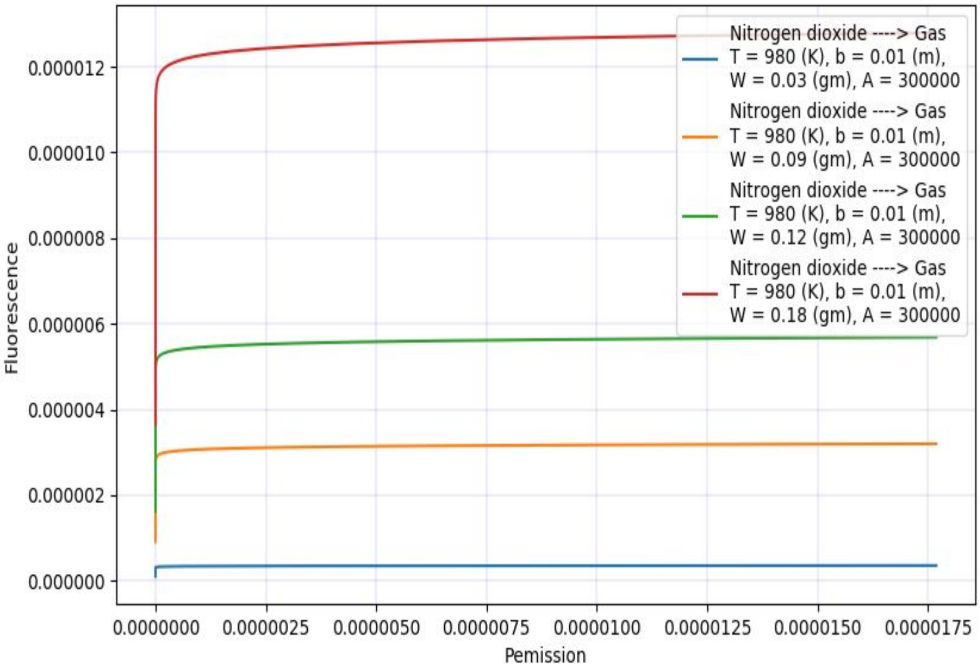


Figure (4.20) the effect of weight on Pemission of Nitrogen dioxide (NO_2)

Figure (4.20) showed that the relationship between the fluorescence and the laser Pemission of Nitrogen dioxide (NO_2) as a function of weight. When the weight increased from 0.03 gm to 0.18 gm the fluorescence increases and also pemission increases as shown by the colour lines in the figure. The blue line represents the fluorescence with pemission when the weight was 0.03 gm the fluorescence was 7.7×10^{-8} pemission was 2.7×10^{-8} . While the Orange line represent when the weight was 0.09 gm the fluorescence was 9.3×10^{-7} pemission was 2.7×10^{-8}. The fluorescence was 1.6×10^{-6} and pemission was 2.7×10^{-8} when the weight was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Nitrogen dioxide was 3.7×10^{-6} and pemission was 2.7×10^{-8}

4.7: Results of water (H_2O):

Water is another type of combustion products, figure (4.21) shows the results of quantum efficiency against the laser wavelengths as a function of the temperature for H_2O combustion product (the vacuum chamber thickness and the molecular weight were kept fixed to 1 cm, and 0.03 gm respectively).

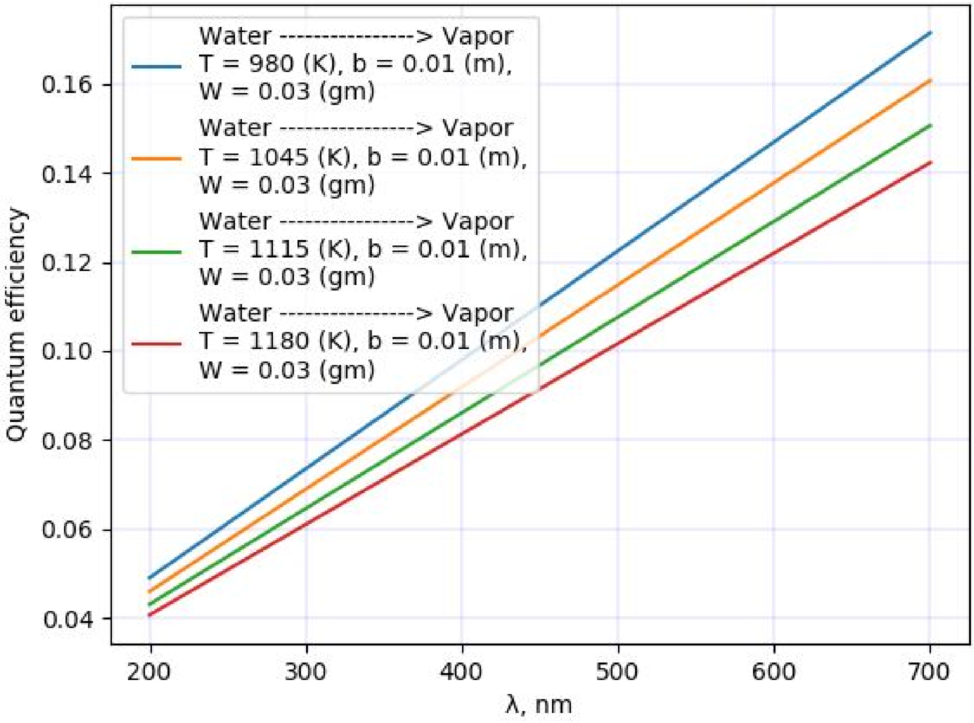


Figure (4.21) effect of temperature on the Quantum efficiency of water

Figure (4 .21) showed that the relationship between the quantum efficiency and the laser wavelengths of water (H_2O) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the quantum efficiency also increases as shown by the colour lines in the figure. The blue line represent the quantum efficiency with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.049 and it became

0.017 at 700 nm . While the Orange line represent the quantum efficiency with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the quantum efficiency was 0.045 and it become 0.016 at 700 nm. The quantum efficiency at 200 nm was 0.043 and 0.015 at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the quantum efficiency of the water was 0.041at 200 nm wavelength and it increased to 0.14at 700 nm wavelength. the lifetime of the water combustion product as a function of temperature is shown in figure (4.22)

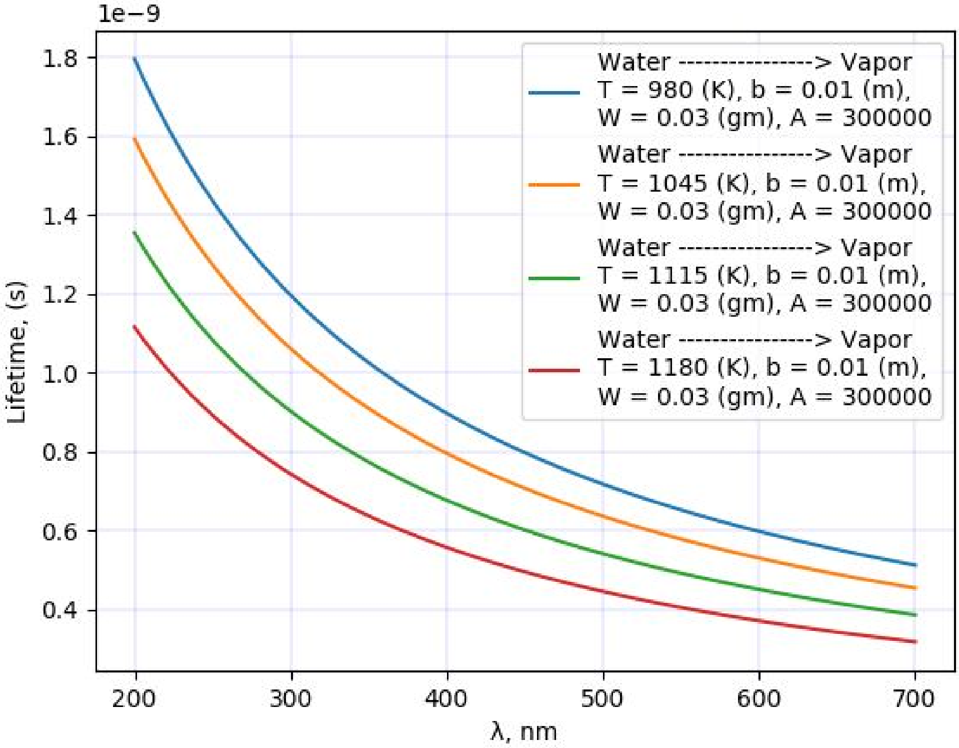


Figure (4. 22) the effect of temperature on lifetime of water

Figure (4. 22) showed that the relationship between the lifetime and the laser wavelengths of water (H_2O) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the lifetime decreases as shown by the colour lines in the figure. The blue line represent the lifetime with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the lifetime was 1.8×10^{-9} and it became 5.1×10^{-10} at 700 nm . While the Orange line represent the lifetime with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the lifetime was $1.6 \times 10^{-9}$and it became 4.6×10^{-10} at 700 nm. The lifetime at 200 nm was 1.3×10^{-9} and 3.9×10^{-10} at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the lifetime of the water was 1.1×10^{-9} at 200 nm wavelength and it increased to 3.2×10^{-10} at 700 nm wavelength, the fluorescence as a function of weight is shown in figure (4.23).

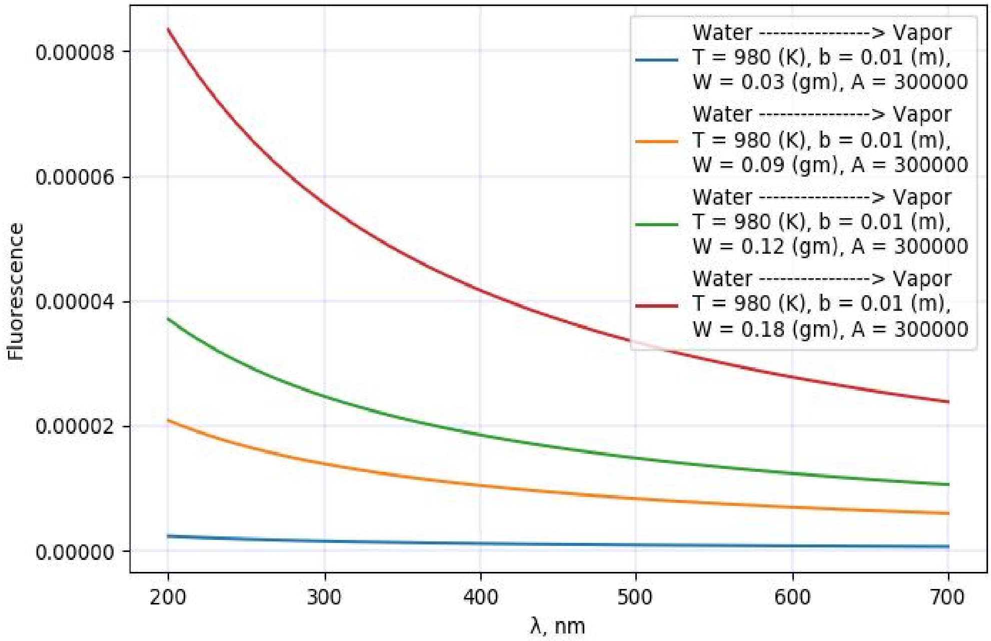


Figure (4.23) the effect of weight on fluorescence of water

Figure (4.23) showed that the relationship between the fluorescence and the laser wavelengths of water (H_2O) as a function of weight and its evident that the fluorescence increased as the weight increased. When the laser wavelengths increased from 200 nm to 700 nm the fluorescence decreases as shown by the colour lines in the figure. The blue line represents the fluorescence with the laser wavelengths when the weight was 0.03 gm at 200 nm wavelength the fluorescence was 2.4×10^{-6} and it became 1×10^{-6} at 700 nm. While the Orange line represents the fluorescence with the laser wavelengths when the weight was 0.09 gm at 200 nm wavelength the fluorescence was 2.1×10^{-5} and it become 6.1×10^{-6} at 700 nm. The fluorescence at 200 nm was 3.7×10^{-5} and 1×10^{-5} at 700 nm when the temperature was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the water was 8.3×10^{-5} at 200 nm wavelength and it decreased to 4.2×10^{-5} at 700 nm wavelength, figure (4.24) shows the effect of weight on the Pemission of water combustion product.

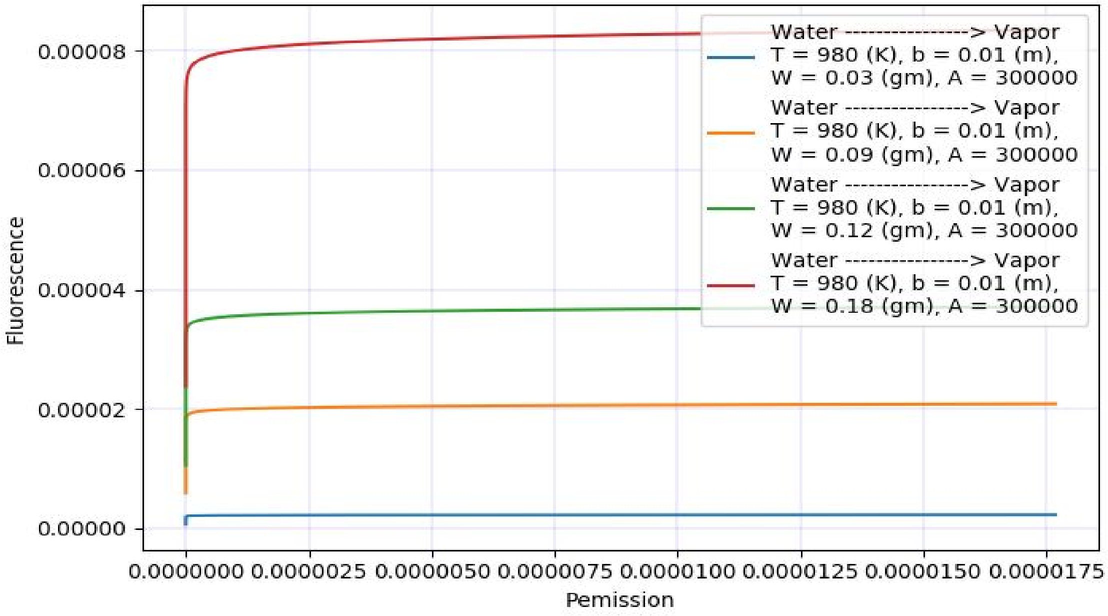


Figure (4.24) the effect of weight on Pemission of water

Figure (4.24) showed that the relationship between the fluorescence and the laser Pemission of water (H_2O) as a function of weight. When the weight increased from 0.03 gm to 0.18 gm the fluorescence increases and also pemission increases as shown by the colour lines in the figure. The blue line represents the fluorescence with pemission when the weight was 0.03 gm the fluorescence was 4.7×10^{-7} pemission was 4.3×10^{-8} . While the Orange line represent when the weight was 0.09 gm the fluorescence was 5.9×10^{-6} pemission was4.3×10^{-8}. The fluorescence was 1.1×10^{-5} and pemission was 4.3×10^{-8} when the weight was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the water was 2.4×10^{-5} and pemission was 4.3×10^{-8}.

4.8 Results of Hydrocarbon (HC)

Researches shows that exhaust emissions from gasoline fuelled motorcycles are of higher concentrations in total hydrocarbon (**HC**), i.e., the Hydrocarbon (**HC**) has high concentration as a combustion products, figure (4.25) shows the results of quantum efficiency against the laser wavelengths as a function of the temperature of the Hydrocarbon gasoline combustion product (the vacuum chamber thickness and the molecular weight were kept fixed to 1 cm, and 0.03 gm respectively).

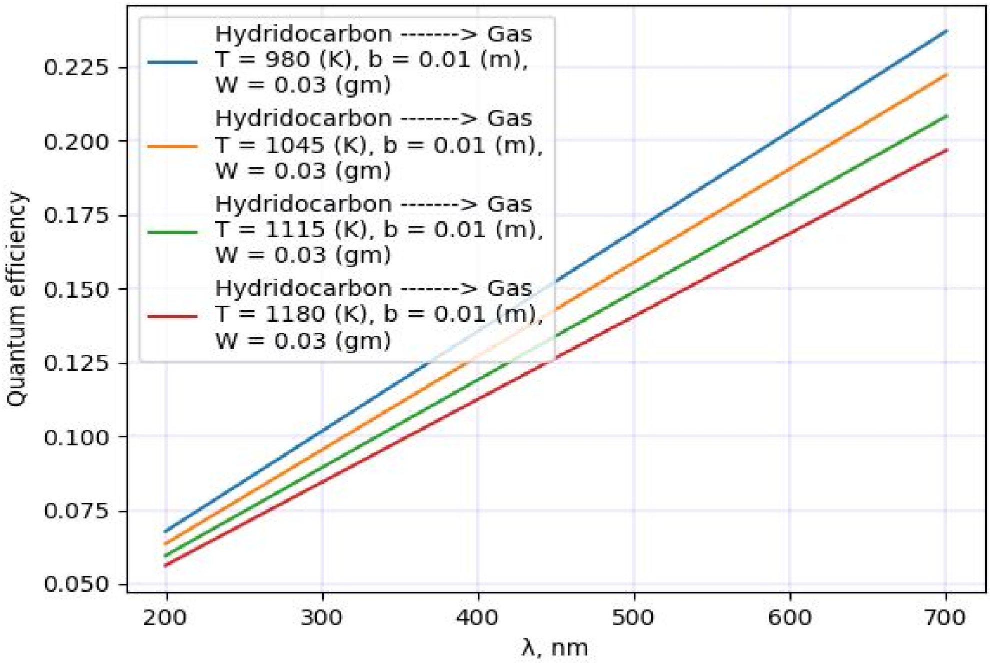


Figure (4.25) the effect of temperature on the Quantum efficiency of Hydrocarbon

Figure (4.25) showed that the relationship between the quantum efficiency and the laser wavelengths of Hydridocarbon (**HC**) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the quantum efficiency also increases as shown by the colour lines in the figure. The blue line represents the quantum efficiency with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.068and it became 0.24 at 700 nm. While the Orange line represents the quantum efficiency with the laser wavelengths when the temperature was 1045 K and at (200 nm) wavelength the quantum efficiency was 0.063and it become 0.22at 700 nm. The quantum efficiency at 200 nm was 0.060 and 0.2 at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the quantum efficiency of the Hydridocarbon was 0.056at 200 nm wavelength and it increased to 0.19 at (700 nm) wavelength.

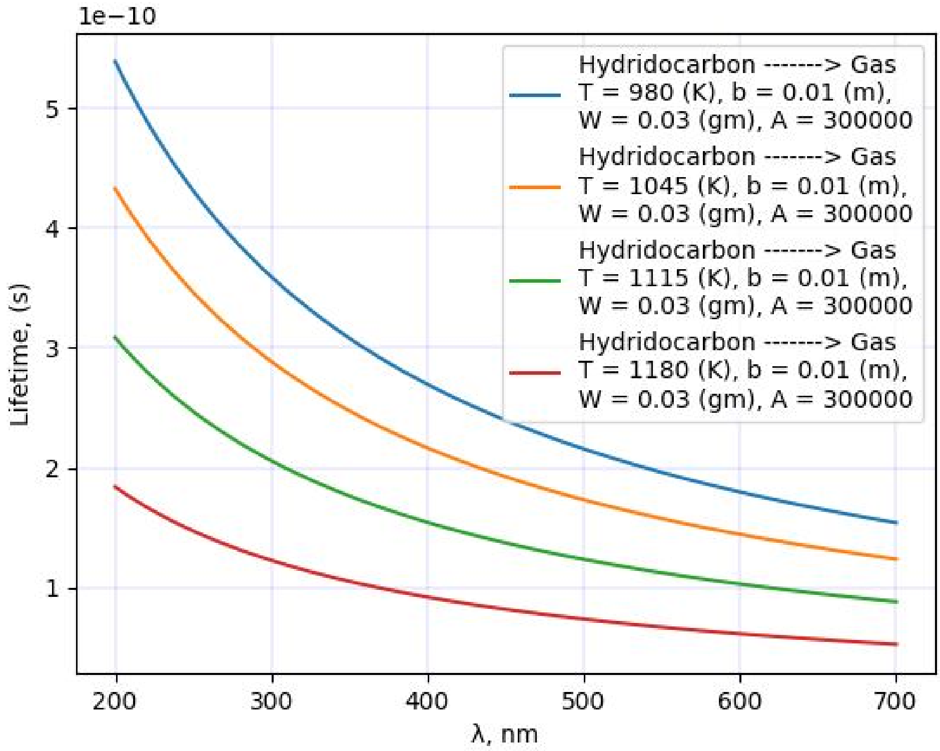


Figure (4 .26) the effect of temperature on lifetime of Hydrocarbon (HC)

Figure (4 .26) showed that the relationship between the lifetime and the laser wavelengths of Hydridocarbon (**HC**) as a function of temperature. When the laser wavelengths increased from 200 nm to 700 nm the lifetime decreases as shown by the colour lines in the figure. The blue line represent the lifetime with the laser wavelengths when the temperature was 980 K and at 200 nm wavelength the lifetime was 5.4×10^{-10} and it became 1.5×10^{-10} at 700 nm. While the Orange line represent the lifetime with the laser wavelengths when the temperature was 1045 K and at 200 nm wavelength the lifetime was $4.3 \times 10^{-10}$and it became 1.2×10^{-10} at 700 nm. The lifetime at 200 nm was 3.1×10^{-10} and 9×10^{-11} at 700 nm when the temperature was 1115 K as shown by the green line. At 1180 K represented by the red line the lifetime of the Hydridocarbon was 1.8×10^{-10} at 200 nm wavelength and it increased to 5.4×10^{-11} at 700 nm wavelength. Figure (2.27) shows the effect of weight on fluorescence of Hydrocarbon (**HC**)

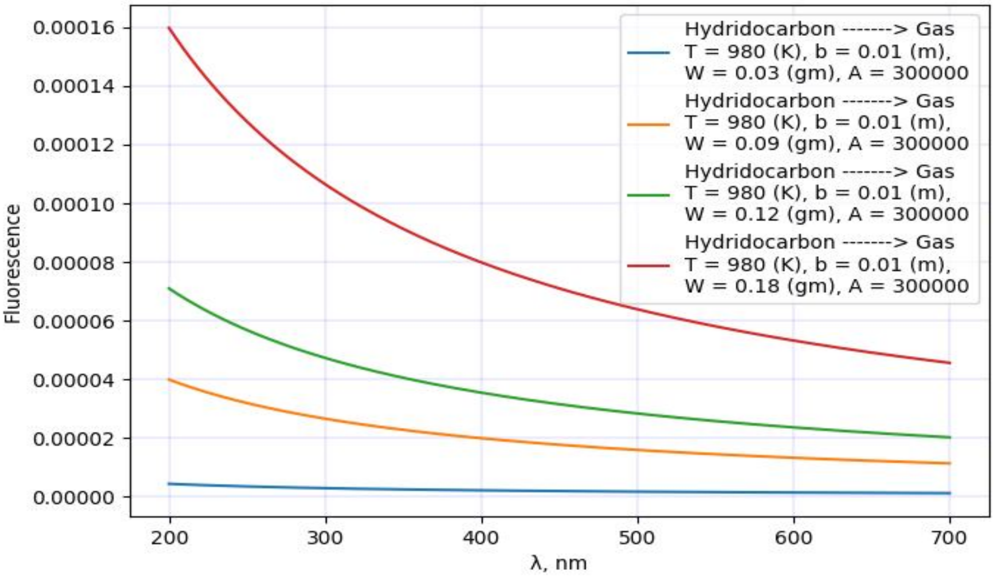


Figure (4.27) the effect of weight on fluorescence of Hydridocarbon (HC)

As shown in figure (4.27) it's clearly that the fluorescence increased as the weight increased. Figure (4.27) showed that the relationship between the fluorescence and the laser wavelengths of Hydridocarbon (**HC**) as a function of weight, when the laser wavelengths increased from 200 nm to 700 nm the fluorescence decreases as shown by the colour lines in the figure. The blue line represents the fluorescence with the laser wavelengths when the weight was 0.03 gm at 200 nm wavelength the fluorescence was 4.7×10^{-6} and it became 2×10^{-6} at 700 nm. While the Orange line represents the fluorescence with the laser wavelengths when the weight was 0.09 gm at 200 nm wavelength the fluorescence was 2.1×10^{-5} and it become 6.1×10^{-6} at 700 nm. The fluorescence at 200 nm was 7.1×10^{-5} and 2.1×10^{-5} at 700 nm when the temperature was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Hydridocarbon was 1.6×10^{-4} at 200 nm wavelength and it decreased to 4.6×10^{-5} at 700 nm wavelength, figure (4.28): The effect of weight on P emission of Hydrocarbon (**HC**) at different temperatures.

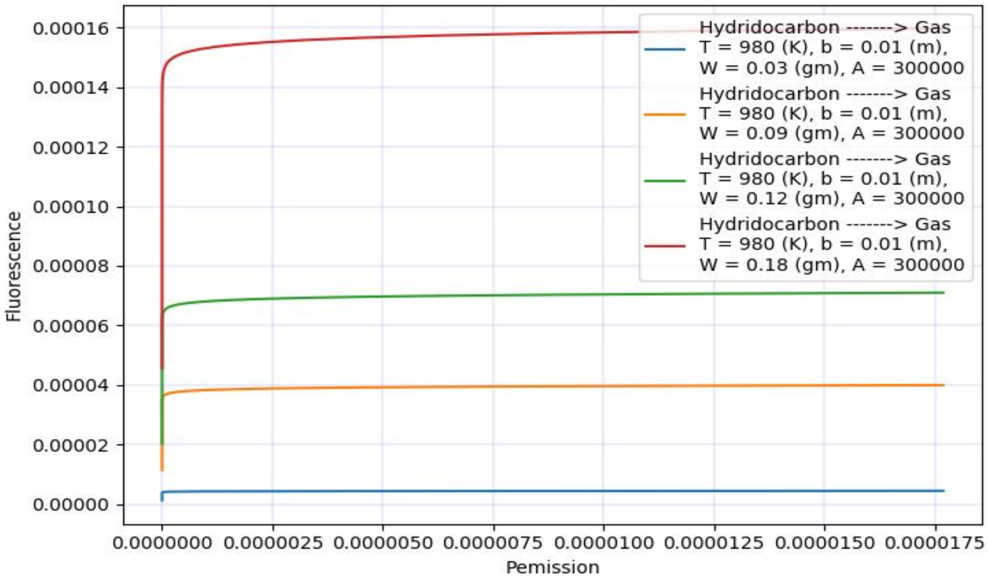


Figure (4.28): The effect of weight on Pemission of Hydridocarbon (HC)

Figure (4.28) showed that the relationship between the fluorescence and the laser $P_{emission}$ of Hydridocarbon (HC) as a function of weight. As shown in figure (4.28) it's clearly that the fluorescence increased as the weight increased. When the weight increased from 0.03 gm to 0.18 gm the fluorescence increases and also pemission increases as shown by the colour lines in the figure. The blue line represents the fluorescence with pemission when the weight was 0.03 gm the fluorescence was 1.4×10^{-6} pemission was 2.7×10^{-8} . While the Orange line represent when the weight was 0.09 gm the fluorescence was 1.1×10^{-5} Pemission was 2.7×10^{-8}. The fluorescence was 2×10^{-5} and Pemission was 2.7×10^{-8} when the weight was 0.12 gm as shown by the green line. At 0.18 gm represented by the red line the fluorescence of the Hydridocarbon was 4.5×10^{-5} and pemission was 2.7×10^{-8}, an important point is that there is a requirement for more investigations about the specific excitation wavelength, and this is done for the case of carbon dioxide gasoline which is the most types of gasoline combustion product. Figure (4.29) shows the fluorescence intensity as a function of the temperature.

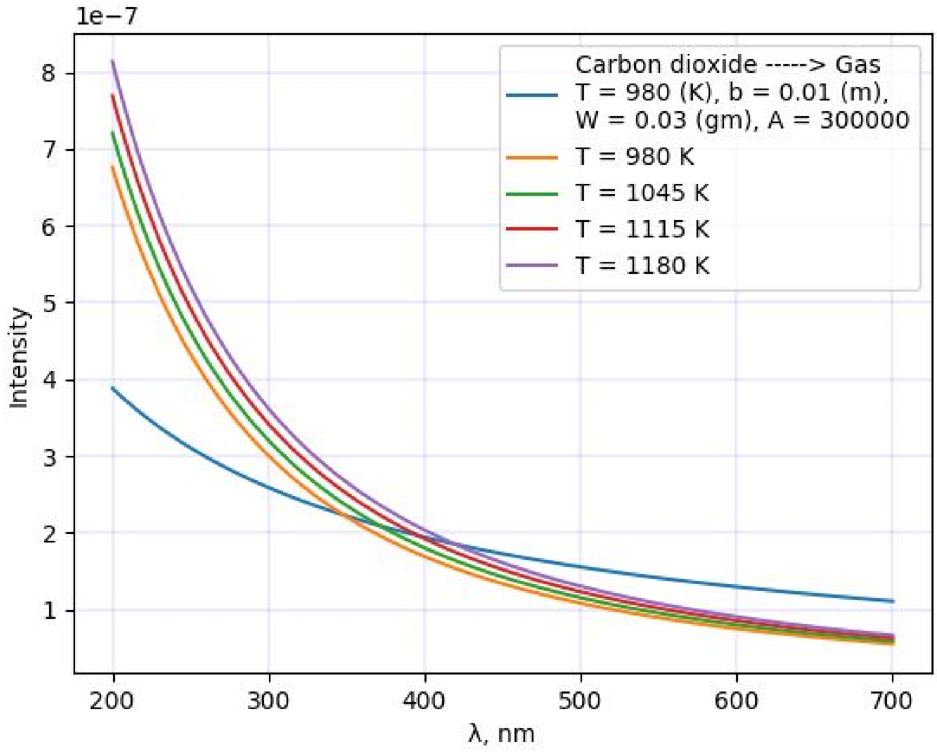


Figure (4.29): fluorescence intensity as a function of temperature

The figure (4.29) shows another contribution of this work which showed that the fluorescence intensity is affected by the temperature. The blue line shows the fluorescence which intercept the intensity wavelength relation as a function of the temperature and the point of intersection give the exact fluorescence wavelength at the specific temperature.

4.9 Discussion:

Figures (4.1, through 4.4) showed the quantum efficiency at different temperatures with laser wavelength, the effect of temperature on lifetime, the effect of weight on fluorescence, and effect of weight on $P_{emission}$ of carbon dioxide gasoline combustion product, respectively. As shown in figure (4.1) the relation of the quantum efficiency with wavelength as a function of temperature is linear one, that is to say increase in the temperature results in an increase in the quantum efficiency of the CO_2, for example when the temperature was 980 K and at 200 nm wavelength the quantum efficiency was 0.020 and it became 0.07 at 700 nm . As shown in figure (4.2) the lifetime of the CO_2 is exponentially decay with the wavelengths, and also the figure showed that the lifetime decreased as the temperature increase, this agree with the theoretical concepts of the excited energy levels in which the increase in energy (thermal temperature) result in the excited species be deactivated or relaxed. Figure (4.3) showed that the fluorescence of carbon dioxide (CO_2) decreased with the increase in the laser wavelengths, also the figure showed that the fluorescence increases with the increase in weight and this is clearly agree with the concepts of population (when the weight of CO_2 is increased that means the population is increase and the probability of the CO_2 for the excitation and hence to perform the fluorescence is increased).Figure (4.4) showed that the CO_2 fluorescence and the $P_{emission}$ both increases with the increase in weight.

For the other gasoline combustion products (CO, N2, NO, NO_2, H_2O and HC), the same results was obtained thus a similar discussion can be used.

Figure (4.29) showed that the fluorescence decreased as the results of increase in temperature but the wavelength is increased.

4.10 Conclusion:

The PYTHON language was used to model the LIF of the gasoline combustion products using the governing equations successfully, the article presented the results of the Carbon dioxide (CO_2) gasoline combustion only, but the software built to investigate the rest of the gasoline combustion products such (CO, NO, NO_2, N_2, H_2O, and HC), the built LIF simulation software was in good agreement with the experimental data of LIF for gasoline combustion products. In conclusion, the fluorescence emission lifetime against wavelength as a function in temperature, quantum efficiency versus laser wavelength as a function of temperature, fluorescence versus wavelength as a function of the weight ratio, and the fluorescence versus $P_{emission}$ as a function of weight ratio was studied using the build LIF simulation for the CO_2 gasoline combustion products. And it was found that the fluorescence intensity decrease while the fluorescence wavelength increase when the temperature is increased for the case of carbon dioxide gasoline combustion product. The same were done for all other six gasoline combustion products. As a conclusion the program was tested and it gives good result that agreed with experimental work of LIF.

4.11 Recommendation:

The following can be regarded as recommendations for future work :

- Generalize the developed LIF program to be used in for other laser spectroscopic technique.

- The program contains seven molecules, so it is recommended to add all other types of molecules.

- Connect the program with other software's such as chemical reaction equation which calculate and Office to generate Excel sheets and tables.

- Synchronize the program with internet and email.
- Mainten of combuston systems in gasoline machines.
- Addition of Exhaust gas Recirculation system and catalytic converter for gasoline machines.

References:

1 - Abbas A-Ali Draea, (2018) Introduction of Molecular Spectroscopy, University of Babylon, College of Science, lectures, downloaded from: http://www.uobabylon.edu.iq/eprints/publication_2_17947_331.pdf

2 - A. Rashid A. Aziz, Firmansyah and Raja Sha (2010), COMBUSTION ANALYSIS OF A CNG DIRECT INJECTION SPARK IGNITION ENGINE, *International Journal of Automotive and Mechanical Engineering* (IJAME), Volume 2, pp. 157-170, July-December 2010

3 - Christian Brackmann et al. *(2004), Laser-induced fluorescence of formaldehyde in combustion using third harmonic Nd: YAG laser excitation, Spectrochimica Acta Part A Molecular and Biomolecular Spectroscopy, Volume 59, issue 14, pp. 3347-56*

4 - Chang Liu & Lijun Xu (2018): Laser absorption spectroscopy for combustion diagnosis in reactive flows: A review, Applied Spectroscopy Reviews, DOI:10.1080/05704928.2018.1448854

5 - Christopher J. Ellison and John M. Torkelson *(2002), sensing the glass transition in thin and ultrathin polymer films via fluorescence probes and labels, Journal of Polymer Sciences: Part B: Polymer Physic, volume 40, pp.2745-2758*

6 - De Sercey, *Guillaume, (2004), Laser induced fluorescence for the measurement of air-to-fuel ratios in gasoline direct injection engines, (PhD thesis), University of Brighton, ISNI: 0000 0001 3420 6275*

7 - Erhard Rothe and Peter Andresen, (1997), Application of Tunable Excimer Lasers to Combustion Diagnostics: A Review, Applied Optics 36(18):3971-4033, DOI: 10.1364/AO.36.003971 74

8 - Gao Star et al. (2015*), Laser Induced fluorescence of fused silica irradiated by KrF laser, The Proceedings of SPIE - The International Society for Optical Engineering 9532, DOI: 10.1117/12.2186063*

9 - John Oliver Stoner et al. (2018), Spectroscopy science, [online] available at: https://www.britannica.com/science/spectroscopy, visited 15/10/2018

10 - Luke Thompson *(2015), Planar Laser Induced Fluorescence Experiments and Modeling Study of Jets in Cross flow at Various Injection Angles, (Master Thesis University of Central Florida), Electronic Theses and Dissertations, 1434, http://stars.library.ucf.edu/etd/1434*

Jochen Scholz *et al. (2006), Verification and Application of Fuel- Air-Ratio-LIF, 13th Int Symp on Applications of Laser Techniques to Fluid Mechanics Lisbon, Portugal, 26-29 June, 2006 #1256*

11 - RAFEEF ABU-GHARBIEH *(2001), Laser Sheet Imaging and Image Analysis for Combustion Research, (PhD thesis), School of Electrical and Computer Engineering Chalmers University of Technology, downloaded from: https://pdfs.semanticscholar.org/1bf3/33b423c846e25f7df53c8c93512c39ce904a.pdf*

12 - Ridchard L. Halterman, (2011*), Learning to Program with Python - Computer Science*

13 - Thalangunam Krishnaswamy Subramaniam (2015), Analysis of Internal Combustion Engines Using Laser Induced Fluorescence Spectroscopy, AASCIT Journal of Physics, 2015; 1(4): 288-291, published online July 10, 2015 (http://www.aascit.org/journal/physics)

14 - T. K. Subramaniam (2015), *Analysis of Internal Combustion Engines Using Laser Induced Fluorescence Spectroscopy, AASCIT Journal of Physics, Volume 1, issue 4, pp. 288-291* 75

15 - Wolfram Kirchweger et al., (2007), Applications of the LIF-method for the diagnostics of the combustion process of gas-IC-engines, 13th Int. Symp. on Applications of Laser Techniques to Fluid Mechanics Lisbon, Portugal, 26-29 June, 2006 Paper #1046-1-

Spectroscopy,(2018) encyclopedia available from: https://en.wikipedia.org/wiki/Spectroscopy, [online], visited 15/8/2018

16 - W.G. Bessler et al., (2003), Carbon dioxide UV laser induced fluorescence in high-pressure flames, Elsevier, Chemical Physics Letters 375,pp.344-349, doi:10.1016/S009-2614(03)00858-3

17 - Zarrin Es'haghi, 2011, Photodiode Array Detection in Clinical Applications; Quantitative Analyte Assay Advantages, Limitations and Disadvantages, Photodiodes - *Communications, Bio-Sensings, Measurements and High-Energy Physics, Associate Professor Jin-Wei Shi (Ed.), ISBN: 978-953-307-277-7,InTech,Availablefrom: http://www.intechopen.com/books/photodiodes-communications-bio-sensings-measurements-and-high-energy-physics/photodiode-array-detection-in-clinical-applications-quantitative-analyte-assay-advantages-limit*

Appendix A :

International Journal of Innovative Studies in Sciences and Engineering Technology (IJISSET)

ISSN 2455-4863 (Online) www.ijisset.org *Volume: 5 Issue: 2 | 2019*

Diagnosis of Carbon dioxide as Combustion Products of Gasoline Engine Using Laser Induce Fluorescence (LIF) Technique Modeled by Python program

Younis Ahmed Abu Aasha[1], Kh. M. Haroun[2], Mustafa Mohammed Badr Eldin[3], Yousif H. Alsheikh[4], Montasir Mohamed Badr Eldien[5]

[1]College of Education, West Kurdufan University, West Kurdufan, Sudan

[2]Faculty of radiobiological and imaging sciences, Alzaeim Al-Azhari University, Omdurman, Sudan

[3] Institute of Laser, Sudan University of Science & Technology, Khartoum, Sudan

[4]Department of Applied Physics & Mathematics, Omdurman Ahlia University, Omdurman, Sudan

[5]College of Education, West Kurdufan University, West Kurdufan, Sudan

Abstract:- This work is aimed to diagnosis the gasoline engine products using laser induce fluorescence technique modelled by python language. The software developed allows us to diagnosis the gasoline combustion products in terms of fluorescence, quantum efficiency, pressure of the emission, and lifetime. In the simulation model the temperature, the vacuum chamber thickness, the weight ratio in gram units of the combustion gas, and the molar absorbtivity were allowed to be varied for the all of gasoline combustion products (**CO_x, NO_x, H_2O, Ni** and **HC**). From the simulation samples of the results of the laser induce fluorescence from Nd: YAG (266 nm) laser source were presented and discussed. The samples of the carried results showed that the Python software developed to study the emission of the combustion products such as CO_2 sample from the gasoline engine is in a good agreement with the literature work

Keywords: Gasoline engine: LIF (laser induces fluorescence): combustion products: Python language

Appendix B :

```
/!#usr/bin/env python
 -*- #coding: utf-8-*-

#import sys
#import os
from tkinter import*
from math import*
from icon_app import file_png

import numpy as np
np.seterr(divide='ignore', invalid='ignore')  # RuntimeWarning: invalid value encountered in
true_divide: https://stackoverflow.com/questions/14861891/runtimewarning-invalid-value-
encountered-in-divide
import matplotlib.pyplot as plt
import webbrowser

window = Tk()

window.wm_title("Laser-induced fluorescence calculations("
window.geometry('615x250') # Size 615, 250
window.configure(background="#E5E5CC("

icon = PhotoImage(file=r'icon.png('
window.tk.call('wm', 'iconphoto', window._w, icon(

======================================================================
Start Equations
======================================================================
var_temperature = DoubleVar() # for scale scale_1, b1
T = var_temperature.get()
 #Lambda fromm 200 toخ »max nm
#var_lambda_max = DoubleVar()  # for خ»max  (scale_16)
#lambda_max = var_lambda_max.get()
lambda_var = (np.arange(200, 701, 1)) * (10 ** -9) # fromm 200 to 400 nm
# light speed (C) = 299 792 458 (m/s)
C = 299792458  # C = 2.99792458 * (10 ** 8)
# Planck's constant (h) = 6.62607004 34^10 —أ (m^2 kg / s)
h = 6.62607004 * (10 ** -34)
# Energy
E = (h * C) / lambda_var
v = C / lambda_var
# I_0 = (2v^2 * K_B * T) / c^2
I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)

# The molar mass of CO_2, CO, N_2, NO, NO_2, H_2O, CH
======================================================================
CO_2,    CO,      N_2,      NO,      NO_2,    H_2O,    CH      = \
44.0095, 28.0101, 28.0134, 30.0061, 46.0055, 18.0153, 13.0186

#print (CO_2)


# The weight of  CO_2, CO, N_2, NO, NO_2, H_2O, CH
======================================================================
var_wCO_2 = DoubleVar()
var_wCO = DoubleVar()
var_wN_2 = DoubleVar()
```

```
var_wNO = DoubleVar()
var_wNO_2 = DoubleVar()
var_wH_2O = DoubleVar()
var_wCH = DoubleVar()

wCO_2 = var_wCO_2.get()
wCO = var_wCO.get()
wN_2 = var_wN_2.get()
wNO = var_wNO.get()
wNO_2 = var_wNO_2.get()
wH_2O = var_wH_2O.get()
wCH = var_wCH.get()

# The concentration of  CO_2, CO, N_2, NO, NO_2, H_2O, CH
============================================================================
concenCO_2 = (wCO_2 / CO_2) * (22.414 / 1.6)
concenCO = (wCO / CO) * (22.414 / 1.6)
concenN_2 = (wN_2 / N_2) * (22.414 / 1.6)
concenNO = (wNO / NO) * (22.414 / 1.6)
concenNO_2 = (wNO_2 / NO_2) * (22.414 / 1.6)
concenH_2O = (wH_2O / H_2O) * (22.414 / 1.6)
concenCH = (wCH / CH) * (22.414 / 1.6)

# Vacuum Chamber Thickness, m
============================================================================
var_b = DoubleVar()
b = var_b.get()

# The molar absorptivity of  CO_2, CO, N_2, NO, NO_2, H_2O, CH
============================================================================
var_mabs_CO_2 = DoubleVar()
var_mabs_CO = DoubleVar()
var_mabs_N_2 = DoubleVar()
var_mabs_NO = DoubleVar()
var_mabs_NO_2 = DoubleVar()
var_mabs_H_2O = DoubleVar()
var_mabs_CH = DoubleVar()

mabs_CO_2 = var_mabs_CO_2.get()
mabs_CO = var_mabs_CO.get()
mabs_N_2 = var_mabs_N_2.get()
mabs_NO = var_mabs_NO.get()
mabs_NO_2 = var_mabs_NO_2.get()
mabs_H_2O = var_mabs_H_2O.get()
mabs_CH = var_mabs_CH.get()

# The absorption of  CO_2, CO, N_2, NO, NO_2, H_2O, CH
============================================================================
aCO_2 = mabs_CO_2 * b * concenCO_2
aCO = mabs_CO * b * concenCO
aN_2 = mabs_N_2 * b * concenN_2
aNO = mabs_NO * b * concenNO
aNO_2 = mabs_NO_2 * b * concenNO_2
aH_2O = mabs_H_2O * b * concenH_2O
aCH = mabs_CH * b * concenCH

# The number mole of  CO_2, CO, N_2, NO, NO_2, H_2O, CH
============================================================================
nmCO_2 = (wCO_2 / CO_2)
nmCO = (wCO / CO)
```

```
nmN_2 = (wN_2 / N_2)
nmNO = (wNO / NO)
nmNO_2 = (wNO_2 / NO_2)
nmH_2O = (wH_2O / H_2O)
nmCH = (wCH / CH)

# The number of molecules reacted of CO_2, CO, N_2, NO, NO_2, H_2O, CH
=============================================================================
nmrCO_2 = nmCO_2 * N_A
nmrCO = nmCO * N_A
nmrN_2 = nmN_2 * N_A
nmrNO = nmNO * N_A
nmrNO_2 = nmNO_2 * N_A
nmrH_2O = nmH_2O * N_A
nmrCH = nmCH * N_A

# Refractive index (n) of (Benzene - gas state) CO_2, CO, N_2, NO, NO_2, H_2O, CH
=============================================================================
# delta_M = n - n_293
n_293_CO_2 = 1.000449 # Refractive index of CO_2 (Carbon Dioxide - gas) when T = 20 C or
293 K
n_293_CO = 1.0003364  # Refractive index of CO (Carbon monoxide) when T = 20 C or 293 K
n_293_N_2 = 1.000297  # Refractive index of N_2 (Nitrogen) when T = 20 C or 293 K
n_293_NO = 1.0002697  # Refractive index of NO (Nitric oxide) when T = 20 C or 293 K
n_293_NO_2 = 1.000516 # Refractive index of NO_2 (Nitrous oxide or Nitrogen dioxide)
when T = 20 C or 293 K
n_293_H_2O = 1.000261 # Refractive index of H_2O (Water vapor) when T = 20 C or 293 K
n_293_CH = 1.000586   # Refractive index of (Methylidyne radical or hydridocarbon(â€¢) CH)
when T = 20 C or 293 K

delta_T = T - 293

n_CO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_CO_2
n_CO = ((3.7 * 10 ** -6) * delta_T) + n_293_CO
n_N_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_N_2
n_NO = ((3.7 * 10 ** -6) * delta_T) + n_293_NO
n_NO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_NO_2
n_H_2O = ((3.7 * 10 ** -6) * delta_T) + n_293_H_2O
n_CH = ((3.7 * 10 ** -6) * delta_T) + n_293_CH

# The time of (CO_2, CO, N_2, NO, NO_2, H_2O, CH): t = b / (C * n)
=============================================================================
t_CO_2 = b / (C * n_CO_2)
t_CO = b / (C * n_CO)
t_N_2 = b / (C * n_N_2)
t_NO = b / (C * n_NO)
t_NO_2 = b / (C * n_NO_2)
t_H_2O = b / (C * n_H_2O)
t_CH = b / (C * n_CH)

# The number of photons absorbed of CO_2, CO, N_2, NO, NO_2, H_2O, CH
=============================================================================
numPhAb_CO_2 = I_0 * (lambda_var / (h * C)) / t_CO_2
numPhAb_CO = I_0 * (lambda_var / (h * C)) / t_CO
numPhAb_N_2 = I_0 * (lambda_var / (h * C)) / t_N_2
numPhAb_NO = I_0 * (lambda_var / (h * C)) / t_NO
numPhAb_NO_2 = I_0 * (lambda_var / (h * C)) / t_NO_2
numPhAb_H_2O = I_0 * (lambda_var / (h * C)) / t_H_2O
numPhAb_CH = I_0 * (lambda_var / (h * C)) / t_CH
```

```
# The Quantum efficiency of CO_2, CO, N_2, NO, NO_2, H_2O, CH
==============================================================================
qeffCO_2 = (nmrCO_2 / numPhAb_CO_2)
qeffCO = (nmrCO / numPhAb_CO)
qeffN_2 = (nmrN_2 / numPhAb_N_2)
qeffNO = (nmrNO / numPhAb_NO)
qeffNO_2 = (nmrNO_2 / numPhAb_NO_2)
qeffH_2O = (nmrH_2O / numPhAb_H_2O)
qeffCH = (nmrCH / numPhAb_CH)

# The fluorescence of  CO_2, CO, N_2, NO, NO_2, H_2O, CH
==============================================================================
fluorCO_2 = I_0 * qeffCO_2 * aCO_2
fluorCO   = I_0 * qeffCO * aCO
fluorN_2  = I_0 * qeffN_2 * aN_2
fluorNO   = I_0 * qeffNO * aNO
fluorNO_2 = I_0 * qeffNO_2 * aNO_2
fluorH_2O = I_0 * qeffH_2O * aH_2O
fluorCH = I_0 * qeffCH * aCH

# Black-body radiation
==============================================================================
BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
    (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))

# Emissivity
==============================================================================
Emissivity = E / BbR

# Pemission
==============================================================================
Pemission = (Emissivity * sigma / C) * (T ** 4)

. R_specific_CO_2 = (8.31432 * 10 ** 3) / CO_2
R_specific_CO = (8.31432 * 10 ** 3) / CO
R_specific_N_2 = (8.31432 * 10 ** 3) / N_2
R_specific_NO = (8.31432 * 10 ** 3) / NO
R_specific_NO_2 = (8.31432 * 10 ** 3) / NO_2
R_specific_H_2O = (8.31432 * 10 ** 3) / H_2O
R_specific_CH = (8.31432 * 10 ** 3) / CH

# Viscosity of Gases of  CO_2, CO, N_2, NO, NO_2, H_2O, CH
==============================================================================
vis_CO = vis_o_CO + (alpha * T) - (beta * (T ** 2))
vis_N_2 = vis_o_N_2 + (alpha * T) - (beta * (T ** 2))
vis_NO = vis_o_NO + (alpha * T) - (beta * (T ** 2))
vis_NO_2 = vis_o_NO_2 + (alpha * T) - (beta * (T ** 2))
vis_H_2O = vis_o_H_2O + (alpha * T) - (beta * (T ** 2))
vis_CH = vis_o_CH + (alpha * T) - (beta * (T ** 2))

# Kq of CO_2, CO, N_2, NO, NO_2, H_2O, CH
==============================================================================
r_CO_2 = 165 * (10 ** -12)
r_CO = 188 * (10 ** -12)
r_N_2 = 132 * (10 ** -12)
r_NO = 158.5 * (10 ** -12)
r_NO_2 = 151 * (10 ** -12)
r_H_2O = 132.5 * (10 ** -12)
r_CH = 146 * (10 ** -12)
```

```
# Lifetime of CO_2, CO, N_2, NO, NO_2, H_2O, CH
==========================================================================
lifetime_CO_2 = 1 / (Kf_CO_2 + ((K_B * T) / 4 * (r_CO_2 ** 3) * vis_CO_2))
lifetime_CO = 1 / (Kf_CO + ((K_B * T) / 4 * (r_CO ** 3) * vis_CO))
lifetime_N_2 = 1 / (Kf_N_2 + ((K_B * T) / 4 * (r_N_2 ** 3) * vis_N_2))
lifetime_NO = 1 / (Kf_NO + ((K_B * T) / 4 * (r_NO ** 3) * vis_NO))
lifetime_NO_2 = 1 / (Kf_NO_2 + ((K_B * T) / 4 * (r_NO_2 ** 3) * vis_NO_2))
lifetime_H_2O = 1 / (Kf_H_2O + ((K_B * T) / 4 * (r_H_2O ** 3) * vis_H_2O))
lifetime_CH = 1 / (Kf_CH + ((K_B * T) / 4 * (r_CH ** 3) * vis_CH))

Start plots
==========================================================================
def plot_lambda_Io():
   T = var_temperature.get()
   #print(T)

   # I_0 = (2v^2 * K_B * T) / c^2
   I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
   #print(I_0)

   #plt.plot(lambda_var, I_0)
   #plt.close()
   plt.plot((lambda_var * (10 ** 9)), I_0, '-', markeredgewidth = '0.5', label= 'T = ' + str(int(T)) + '
K')
   plt.xlabel(' nm')
   plt.ylabel('Intensity')
   plt.legend()
   plt.grid(color='b', linestyle='-', linewidth=0.1)
   plt.show()


def plot_lambda_qeffCO_2():
   T = var_temperature.get()
   I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
   #print(I_0)
   wCO_2 = var_wCO_2.get()
   #print(wCO_2)
   b = var_b.get()
   #print(b)
   concenCO_2 = (wCO_2 / CO_2) * (22.414 / 1.6)
   #print(concenCO_2)
   mabs_CO_2 = var_mabs_CO_2.get()
   aCO_2 = mabs_CO_2 * b * concenCO_2
   #print(aCO_2)
   nmCO_2 = (wCO_2 / CO_2)
   #print(nmCO_2)
   nmrCO_2 = nmCO_2 * N_A
   #print(nmrCO_2)
   n_293_CO_2 = 1.000449 # Refractive index of CO_2 (Carbon Dioxide - gas) when T = 20 C
or 293 K
   delta_T = T - 293
   n_CO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_CO_2
   #print(n_CO_2)
   t_CO_2 = b / (C * n_CO_2)
   #print(t_CO_2)
   numPhAb_CO_2 = I_0 * (lambda_var / (h * C)) / t_CO_2
   #print(numPhAb_CO_2)
   qeffCO_2 = (nmrCO_2 / numPhAb_CO_2)
   #print(qeffCO_2)
```

```
    plt.plot((lambda_var * (10 ** 9)), qeffCO_2, '-', markeredgewidth = '0.5', label= 'Carbon
dioxide -----> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO_2) +
' (gm)')
    plt.xlabel('خnm')
    plt.ylabel('Quantum efficiency')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lambda_qeffCO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCO = var_wCO.get()
    #print(wCO)
    b = var_b.get()
    #print(b)
    concenCO = (wCO / CO) * (22.414 / 1.6)
    #print(concenCO_2)
    mabs_CO = var_mabs_CO.get()
    aCO = mabs_CO * b * concenCO
    #print(aCO)
    nmCO = (wCO / CO)
    #print(nmCO)
    nmrCO = nmCO * N_A
    #print(nmrCO)
    n_293_CO = 1.0003364  # Refractive index of CO (Carbon monoxide) when T = 20 C or 293
    delta_T = T - 293
    n_CO = ((3.7 * 10 ** -6) * delta_T) + n_293_CO
    #print(n_CO)
    t_CO = b / (C * n_CO)
    #print(t_CO)
    numPhAb_CO = I_0 * (lambda_var / (h * C)) / t_CO
    #print(numPhAb_CO)
    qeffCO = (nmrCO / numPhAb_CO)
    #print(qeffCO)

    plt.plot((lambda_var * (10 ** 9)), qeffCO, '-', markeredgewidth = '0.5', label= 'Carbon
monoxide --> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO) + '
(gm)')
    plt.xlabel(' nm')
    plt.ylabel('Quantum efficiency')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lambda_qeffN_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wN_2 = var_wN_2.get()
    #print(wN_2)
    b = var_b.get()
    #print(b)
    concenN_2 = (wN_2 / N_2) * (22.414 / 1.6)
    #print(concenN_2)
    mabs_N_2 = var_mabs_N_2.get()
    aN_2 = mabs_N_2 * b * concenN_2
    #print(aN_2)
    nmN_2 = (wN_2 / N_2)
    #print(nmN_2)
    nmrN_2 = nmN_2 * N_A
    #print(nmrN_2)
    n_293_N_2 = 1.000297  # Refractive index of N_2 (Nitrogen) when T = 20 C or 293 K
    delta_T = T - 293
    n_N_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_N_2
    #print(n_N_2)
    t_N_2 = b / (C * n_N_2)
    #print(t_N_2)
    numPhAb_N_2 = I_0 * (lambda_var / (h * C)) / t_N_2
    #print(numPhAb_N_2)
    qeffN_2 = (nmrN_2 / numPhAb_N_2)
    #print(qeffN_2)

    plt.plot((lambda_var * (10 ** 9)), qeffN_2, '-', markeredgewidth = '0.5', label= 'Nitrogen ------
---------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wN_2) + ' (gm)')
    plt.xlabel('خ», nm')
    plt.ylabel('Quantum efficiency')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lambda_qeffNO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO = var_wNO.get()
    #print(wNO)
    b = var_b.get()
    #print(b)
    concenNO = (wNO / NO) * (22.414 / 1.6)
    #print(concenNO)
    mabs_NO = var_mabs_NO.get()
    aNO = mabs_NO * b * concenNO
    #print(aNO)
    nmNO = (wNO / NO)
    #print(nmNO)
    nmrNO = nmNO * N_A
    #print(nmrNO)
    n_293_NO = 1.0002697  # Refractive index of NO (Nitric oxide) when T = 20 C or 293 K
    delta_T = T - 293
    n_NO = ((3.7 * 10 ** -6) * delta_T) + n_293_NO
    #print(n_NO)
    t_NO = b / (C * n_NO)
    #print(t_NO)
    numPhAb_NO = I_0 * (lambda_var / (h * C)) / t_NO
    #print(numPhAb_NO)
    qeffNO = (nmrNO / numPhAb_NO)
    #print(qeffNO)

    plt.plot((lambda_var * (10 ** 9)), qeffNO, '-', markeredgewidth = '0.5', label= 'Nitric oxide ---
--------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wNO) + ' (gm)')
    plt.xlabel('خ», nm')
    plt.ylabel('Quantum efficiency')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()
```

```
def plot_lambda_qeffNO_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO_2 = var_wNO_2.get()
    #print(wNO_2)
    b = var_b.get()
    #print(b)
    concenNO_2 = (wNO_2 / NO_2) * (22.414 / 1.6)
    #print(concenNO_2)
    mabs_NO_2 = var_mabs_NO_2.get()
    aNO_2 = mabs_NO_2 * b * concenNO_2
    #print(aNO_2)
    nmNO_2 = (wNO_2 / NO_2)
    #print(nmNO_2)
    nmrNO_2 = nmNO_2 * N_A
    #print(nmrNO_2)
    n_293_NO_2 = 1.000516 # Refractive index of NO_2 (Nitrous oxide or Nitrogen dioxide)
when T = 20 C or 293 K
    delta_T = T - 293
    n_NO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_NO_2
    #print(n_NO_2)
    t_NO_2 = b / (C * n_NO_2)
    #print(t_NO_2)
    numPhAb_NO_2 = I_0 * (lambda_var / (h * C)) / t_NO_2
    #print(numPhAb_NO_2)
    qeffNO_2 = (nmrNO_2 / numPhAb_NO_2)
    #print(qeffNO_2)

    plt.plot((lambda_var * (10 ** 9)), qeffNO_2, '-', markeredgewidth = '0.5', label= 'Nitrogen
dioxide ----> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wNO_2) + '
(gm)')
    plt.xlabel('ج», nm')
    plt.ylabel('Quantum efficiency')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lambda_qeffH_2O():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wH_2O = var_wH_2O.get()
    #print(wH_2O)
    b = var_b.get()
    #print(b)
    concenH_2O = (wH_2O / H_2O) * (22.414 / 1.6)
    #print(concenH_2O)
    mabs_H_2O = var_mabs_H_2O.get()
    aH_2O = mabs_H_2O * b * concenH_2O
    #print(aH_2O)
    nmH_2O = (wH_2O / H_2O)
    #print(nmH_2O)
    nmrH_2O = nmH_2O * N_A
    #print(nmrH_2O)
    n_293_H_2O = 1.000261 # Refractive index of H_2O (Water vapor) when T = 20 C or 293 K
    delta_T = T - 293
    n_H_2O = ((3.7 * 10 ** -6) * delta_T) + n_293_H_2O
    #print(n_H_2O)
    t_H_2O = b / (C * n_H_2O)
    #print(t_H_2O)
    numPhAb_H_2O = I_0 * (lambda_var / (h * C)) / t_H_2O
    #print(numPhAb_H_2O)
    qeffH_2O = (nmrH_2O / numPhAb_H_2O)
    #print(qeffH_2O)

    plt.plot((lambda_var * (10 ** 9)), qeffH_2O, '-', markeredgewidth = '0.5', label= 'Water --------
--------> Vapor\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wH_2O) + '
(gm)')
    plt.xlabel('خ», nm')
    plt.ylabel('Quantum efficiency')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lambda_qeffCH():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCH = var_wCH.get()
    #print(wCH)
    b = var_b.get()
    #print(b)
    concenCH = (wCH / CH) * (22.414 / 1.6)
    #print(concenCH)
    mabs_CH = var_mabs_CH.get()
    aCH = mabs_CH * b * concenCH
    #print(aCH)
    nmCH = (wCH / CH)
    #print(nmCH)
    nmrCH = nmCH * N_A
    #print(nmrCH)
    n_293_CH = 1.000586   # Refractive index of (Methylidyne radical or hydridocarbon(â€¢)
CH) when T = 20 C or 293 K
    delta_T = T - 293
    n_CH = ((3.7 * 10 ** -6) * delta_T) + n_293_CH
    #print(n_CH)
    t_CH = b / (C * n_CH)
    #print(t_CH)
    numPhAb_CH = I_0 * (lambda_var / (h * C)) / t_CH
    #print(numPhAb_CH)
    qeffCH = (nmrCH / numPhAb_CH)
    #print(qeffCH)

    plt.plot((lambda_var * (10 ** 9)), qeffCH, '-', markeredgewidth = '0.5', label=
'Hydridocarbon -------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' +
str(wCH) + ' (gm)')
    plt.xlabel('خ», nm')
    plt.ylabel('Quantum efficiency')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()



def plot_fluorCO_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCO_2 = var_wCO_2.get()
    #print(wCO_2)
    b = var_b.get()
    #print(b)
    concenCO_2 = (wCO_2 / CO_2) * (22.414 / 1.6)
    #print(concenCO_2)
    mabs_CO_2 = var_mabs_CO_2.get()
    aCO_2 = mabs_CO_2 * b * concenCO_2
    #print(aCO_2)
    nmCO_2 = (wCO_2 / CO_2)
    #print(nmCO_2)
    nmrCO_2 = nmCO_2 * N_A
    #print(nmrCO_2)
    n_293_CO_2 = 1.000449 # Refractive index of CO_2 (Carbon Dioxide - gas) when T = 20 C
or 293 K
    delta_T = T - 293
    n_CO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_CO_2
    #print(n_CO_2)
    t_CO_2 = b / (C * n_CO_2)
    #print(t_CO_2)
    numPhAb_CO_2 = I_0 * (lambda_var / (h * C)) / t_CO_2
    #print(numPhAb_CO_2)
    qeffCO_2 = (nmrCO_2 / numPhAb_CO_2)
    #print(qeffCO_2)
    fluorCO_2 = I_0 * qeffCO_2 * aCO_2
    #print(fluorCO_2)

    plt.plot((lambda_var * (10 ** 9)), fluorCO_2, '-', markeredgewidth = '0.5', label= 'Carbon
dioxide -----> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO_2) +
' (gm), ' + 'A = ' + str(int(mabs_CO_2)))
    plt.xlabel(' nm')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_fluorCO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCO = var_wCO.get()
    #print(wCO)
    b = var_b.get()
    #print(b)
    concenCO = (wCO / CO) * (22.414 / 1.6)
    #print(concenCO)
    mabs_CO = var_mabs_CO.get()
    aCO = mabs_CO * b * concenCO
    #print(aCO)
    nmCO = (wCO / CO)
    #print(nmCO)
    nmrCO = nmCO * N_A
    #print(nmrCO)
    n_293_CO = 1.0003364 # Refractive index of CO (Carbon Dioxide - gas) when T = 20 C or
293 K
    delta_T = T - 293
    n_CO = ((3.7 * 10 ** -6) * delta_T) + n_293_CO
    #print(n_CO)
    t_CO = b / (C * n_CO)
    #print(t_CO)
    numPhAb_CO = I_0 * (lambda_var / (h * C)) / t_CO
    #print(numPhAb_CO)
    qeffCO = (nmrCO / numPhAb_CO)
    #print(qeffCO)
    fluorCO = I_0 * qeffCO * aCO
    #print(fluorCO)

    plt.plot((lambda_var * (10 ** 9)), fluorCO, '-', markeredgewidth = '0.5', label= 'Carbon
monoxide --> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO) + '
(gm), ' + 'A = ' + str(int(mabs_CO)))
    plt.xlabel(' nm')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_fluorN_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wN_2 = var_wN_2.get()
    #print(wN_2)
    b = var_b.get()
    #print(b)
    concenN_2 = (wN_2 / N_2) * (22.414 / 1.6)
    #print(concenN_2)
    mabs_N_2 = var_mabs_N_2.get()
    aN_2 = mabs_N_2 * b * concenN_2
    #print(aN_2)
    nmN_2 = (wN_2 / N_2)
    #print(nmN_2)
    nmrN_2 = nmN_2 * N_A
    #print(nmrN_2)
    n_293_N_2 = 1.000297  # Refractive index of N_2 (Nitrogen) when T = 20 C or 293 K
    delta_T = T - 293
    n_N_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_N_2
    #print(n_N_2)
    t_N_2 = b / (C * n_N_2)
    #print(t_N_2)
    numPhAb_N_2 = I_0 * (lambda_var / (h * C)) / t_N_2
    #print(numPhAb_N_2)
    qeffN_2 = (nmrN_2 / numPhAb_N_2)
    #print(qeffN_2)
    fluorN_2 = I_0 * qeffN_2 * aN_2
    #print(fluorN_2)

    plt.plot((lambda_var * (10 ** 9)), fluorN_2, '-', markeredgewidth = '0.5', label= 'Nitrogen -----
----------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wN_2) + ' (gm),
' + 'A = ' + str(int(mabs_N_2)))
    plt.xlabel(' nm')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()
```

```
def plot_fluorNO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO = var_wNO.get()
    #print(wNO)
    b = var_b.get()
    #print(b)
    concenNO = (wNO / NO) * (22.414 / 1.6)
    #print(concenNO)
    mabs_NO = var_mabs_NO.get()
    aNO = mabs_NO * b * concenNO
    #print(aNO)
    nmNO = (wNO / NO)
    #print(nmNO)
    nmrNO = nmNO * N_A
    #print(nmrNO)
    n_293_NO = 1.0002697  # Refractive index of NO (Nitric oxide) when T = 20 C or 293 K
    delta_T = T - 293
    n_NO = ((3.7 * 10 ** -6) * delta_T) + n_293_NO
    #print(n_NO)
    t_NO = b / (C * n_NO)
    #print(t_NO)
    numPhAb_NO = I_0 * (lambda_var / (h * C)) / t_NO
    #print(numPhAb_NO)
    qeffNO = (nmrNO / numPhAb_NO)
    #print(qeffNO)
    fluorNO = I_0 * qeffNO * aNO
    #print(fluorNO)

    plt.plot((lambda_var * (10 ** 9)), fluorNO, '-', markeredgewidth = '0.5', label= 'Nitric oxide --
---------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wNO) + ' (gm), '
+ 'A = ' + str(int(mabs_NO)))
    plt.xlabel('خ», nm')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_fluorNO_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO_2 = var_wNO_2.get()
    #print(wNO_2)
    b = var_b.get()
    #print(b)
    concenNO_2 = (wNO_2 / NO_2) * (22.414 / 1.6)
    #print(concenNO_2)
    mabs_NO_2 = var_mabs_NO_2.get()
    aNO_2 = mabs_NO_2 * b * concenNO_2
    #print(aNO_2)
    nmNO_2 = (wNO_2 / NO_2)
    #print(nmNO_2)
    nmrNO_2 = nmNO_2 * N_A
    #print(nmrNO_2)
    n_293_NO_2 = 1.000516 # Refractive index of NO_2 (Nitrous oxide or Nitrogen dioxide)
when T = 20 C or 293 K
```

```
    delta_T = T - 293
    n_NO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_NO_2
    #print(n_NO_2)
    t_NO_2 = b / (C * n_NO_2)
    #print(t_NO_2)
    numPhAb_NO_2 = I_0 * (lambda_var / (h * C)) / t_NO_2
    #print(numPhAb_NO_2)
    qeffNO_2 = (nmrNO_2 / numPhAb_NO_2)
    #print(qeffNO_2)
    fluorNO_2 = I_0 * qeffNO_2 * aNO_2
    #print(fluorNO_2)

    plt.plot((lambda_var * (10 ** 9)), fluorNO_2, '-', markeredgewidth = '0.5', label= 'Nitrogen
dioxide ----> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wNO_2) + '
(gm), ' + 'A = ' + str(int(mabs_NO_2)))
    plt.xlabel('ج», nm')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_fluorH_2O():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wH_2O = var_wH_2O.get()
    #print(wH_2O)
    b = var_b.get()
    #print(b)
    concenH_2O = (wH_2O / H_2O) * (22.414 / 1.6)
    #print(concenH_2O)
    mabs_H_2O = var_mabs_H_2O.get()
    aH_2O = mabs_H_2O * b * concenH_2O
    #print(aH_2O)
    nmH_2O = (wH_2O / H_2O)
    #print(nmH_2O)
    nmrH_2O = nmH_2O * N_A
    #print(nmrH_2O)
    n_293_H_2O = 1.000261 # Refractive index of H_2O (Water vapor) when T = 20 C or 293 K
    delta_T = T - 293
    n_H_2O = ((3.7 * 10 ** -6) * delta_T) + n_293_H_2O
    #print(n_H_2O)
    t_H_2O = b / (C * n_H_2O)
    #print(t_H_2O)
    numPhAb_H_2O = I_0 * (lambda_var / (h * C)) / t_H_2O
    #print(numPhAb_H_2O)
    qeffH_2O = (nmrH_2O / numPhAb_H_2O)
    #print(qeffH_2O)
    fluorH_2O = I_0 * qeffH_2O * aH_2O
    #print(fluorH_2O)

    plt.plot((lambda_var * (10 ** 9)), fluorH_2O, '-', markeredgewidth = '0.5', label= 'Water -------
---------> Vapor\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wH_2O) + '
(gm), ' + 'A = ' + str(int(mabs_H_2O)))
    plt.xlabel('ج», nm')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()
```

```
def plot_fluorCH():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCH = var_wCH.get()
    #print(wCH)
    b = var_b.get()
    #print(b)
    concenCH = (wCH / CH) * (22.414 / 1.6)
    #print(concenCH)
    mabs_CH = var_mabs_CH.get()
    aCH = mabs_CH * b * concenCH
    #print(aCH)
    nmCH = (wCH / CH)
    #print(nmCH)
    nmrCH = nmCH * N_A
    #print(nmrCH)
    n_293_CH = 1.000586   # Refractive index of (Methylidyne radical or hydridocarbon(â€¢)
CH) when T = 20 C or 293 K
    delta_T = T - 293
    n_CH = ((3.7 * 10 ** -6) * delta_T) + n_293_CH
    #print(n_CH)
    t_CH = b / (C * n_CH)
    #print(t_CH)
    numPhAb_CH = I_0 * (lambda_var / (h * C)) / t_CH
    #print(numPhAb_CH)
    qeffCH = (nmrCH / numPhAb_CH)
    #print(qeffCH)
    fluorCH = I_0 * qeffCH * aCH
    #print(fluorCH)

    plt.plot((lambda_var * (10 ** 9)), fluorCH, '-', markeredgewidth = '0.5', label=
'Hydridocarbon -------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' +
str(wCH) + ' (gm), ' + 'A = ' + str(int(mabs_CH)))
    plt.xlabel(' nm')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()



def plot_Pemission_CO_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCO_2 = var_wCO_2.get()
    #print(wCO_2)
    b = var_b.get()
    #print(b)
    concenCO_2 = (wCO_2 / CO_2) * (22.414 / 1.6)
    #print(concenCO_2)
    mabs_CO_2 = var_mabs_CO_2.get()
    aCO_2 = mabs_CO_2 * b * concenCO_2
    #print(aCO_2)
    nmCO_2 = (wCO_2 / CO_2)
    #print(nmCO_2)
    nmrCO_2 = nmCO_2 * N_A
    #print(nmrCO_2)
    n_293_CO_2 = 1.000449 # Refractive index of CO_2 (Carbon Dioxide - gas) when T = 20 C
or 293 K
    delta_T = T - 293
    n_CO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_CO_2
    #print(n_CO_2)
    t_CO_2 = b / (C * n_CO_2)
    #print(t_CO_2)
    numPhAb_CO_2 = I_0 * (lambda_var / (h * C)) / t_CO_2
    #print(numPhAb_CO_2)
    qeffCO_2 = (nmrCO_2 / numPhAb_CO_2)
    #print(qeffCO_2)
    fluorCO_2 = I_0 * qeffCO_2 * aCO_2
    #print(fluorCO_2)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
          (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    plt.plot(Pemission, fluorCO_2, '-', markeredgewidth = '0.5', label= 'Carbon dioxide ----->
Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO_2) + ' (gm), ' + 'A =
' + str(int(mabs_CO_2)))
    plt.xlabel('Pemission')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_Pemission_CO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCO = var_wCO.get()
    #print(wCO)
    b = var_b.get()
    #print(b)
    concenCO = (wCO / CO) * (22.414 / 1.6)
    #print(concenCO)
    mabs_CO = var_mabs_CO.get()
    aCO = mabs_CO * b * concenCO
    #print(aCO)
    nmCO = (wCO / CO)
    #print(nmCO)
    nmrCO = nmCO * N_A
    #print(nmrCO)
    n_293_CO = 1.0003364 # Refractive index of CO (Carbon Dioxide - gas) when T = 20 C or
293 K
    delta_T = T - 293
    n_CO = ((3.7 * 10 ** -6) * delta_T) + n_293_CO
    #print(n_CO)
    t_CO = b / (C * n_CO)
    #print(t_CO)
    numPhAb_CO = I_0 * (lambda_var / (h * C)) / t_CO
    #print(numPhAb_CO)
    qeffCO = (nmrCO / numPhAb_CO)
    #print(qeffCO)
    fluorCO = I_0 * qeffCO * aCO
    #print(fluorCO)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
        (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    plt.plot(Pemission, fluorCO, '-', markeredgewidth = '0.5', label= 'Carbon monoxide -->
Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO) + ' (gm), ' + 'A = '
+ str(int(mabs_CO)))
    plt.xlabel('Pemission')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_Pemission_N_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wN_2 = var_wN_2.get()
    #print(wN_2)
    b = var_b.get()
    #print(b)
    concenN_2 = (wN_2 / N_2) * (22.414 / 1.6)
    #print(concenN_2)
    mabs_N_2 = var_mabs_N_2.get()
    aN_2 = mabs_N_2 * b * concenN_2
    #print(aN_2)
    nmN_2 = (wN_2 / N_2)
    #print(nmN_2)
    nmrN_2 = nmN_2 * N_A
    #print(nmrN_2)
    n_293_N_2 = 1.000297  # Refractive index of N_2 (Nitrogen) when T = 20 C or 293 K
    delta_T = T - 293
    n_N_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_N_2
    #print(n_N_2)
    t_N_2 = b / (C * n_N_2)
    #print(t_N_2)
    numPhAb_N_2 = I_0 * (lambda_var / (h * C)) / t_N_2
    #print(numPhAb_N_2)
    qeffN_2 = (nmrN_2 / numPhAb_N_2)
    #print(qeffN_2)
    fluorN_2 = I_0 * qeffN_2 * aN_2
    #print(fluorN_2)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
        (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)
    plt.plot(Pemission, fluorN_2, '-', markeredgewidth = '0.5', label= 'Nitrogen --------------->
Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wN_2) + ' (gm), ' + 'A = '
+ str(int(mabs_N_2)))
    plt.xlabel('Pemission')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_Pemission_NO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO = var_wNO.get()
    #print(wNO)
    b = var_b.get()
    #print(b)
    concenNO = (wNO / NO) * (22.414 / 1.6)
    #print(concenNO)
    mabs_NO = var_mabs_NO.get()
    aNO = mabs_NO * b * concenNO
    #print(aNO)
    nmNO = (wNO / NO)
    #print(nmNO)
    nmrNO = nmNO * N_A
    #print(nmrNO)
    n_293_NO = 1.0002697  # Refractive index of NO (Nitric oxide) when T = 20 C or 293 K
    delta_T = T - 293
    n_NO = ((3.7 * 10 ** -6) * delta_T) + n_293_NO
    #print(n_NO)
    t_NO = b / (C * n_NO)
    #print(t_NO)
    numPhAb_NO = I_0 * (lambda_var / (h * C)) / t_NO
    #print(numPhAb_NO)
    qeffNO = (nmrNO / numPhAb_NO)
    #print(qeffNO)
    fluorNO = I_0 * qeffNO * aNO
    #print(fluorNO)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
          (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    plt.plot(Pemission, fluorNO, '-', markeredgewidth = '0.5', label= 'Nitric oxide ----------->
Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wNO) + ' (gm), ' + 'A = '
+ str(int(mabs_NO)))
    plt.xlabel('Pemission')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_Pemission_NO_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO_2 = var_wNO_2.get()
    #print(wNO_2)
    b = var_b.get()
    #print(b)
    concenNO_2 = (wNO_2 / NO_2) * (22.414 / 1.6)
    #print(concenNO_2)
    mabs_NO_2 = var_mabs_NO_2.get()
    aNO_2 = mabs_NO_2 * b * concenNO_2
    #print(aNO_2)
    nmNO_2 = (wNO_2 / NO_2)
    #print(nmNO_2)
    nmrNO_2 = nmNO_2 * N_A
    #print(nmrNO_2)
    n_293_NO_2 = 1.000516 # Refractive index of NO_2 (Nitrous oxide or Nitrogen dioxide)
when T = 20 C or 293 K
    delta_T = T - 293
    n_NO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_NO_2
    #print(n_NO_2)
    t_NO_2 = b / (C * n_NO_2)
    #print(t_NO_2)
    numPhAb_NO_2 = I_0 * (lambda_var / (h * C)) / t_NO_2
    #print(numPhAb_NO_2)
    qeffNO_2 = (nmrNO_2 / numPhAb_NO_2)
    #print(qeffNO_2)
    fluorNO_2 = I_0 * qeffNO_2 * aNO_2
    #print(fluorNO_2)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
          (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    plt.plot(Pemission, fluorNO_2, '-', markeredgewidth = '0.5', label= 'Nitrogen dioxide ---->
Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wNO_2) + ' (gm), ' + 'A =
' + str(int(mabs_NO_2)))
    plt.xlabel('Pemission')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_Pemission_H_2O():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wH_2O = var_wH_2O.get()
    #print(wH_2O)
    b = var_b.get()
    #print(b)
    concenH_2O = (wH_2O / H_2O) * (22.414 / 1.6)
    #print(concenH_2O)
    mabs_H_2O = var_mabs_H_2O.get()
    aH_2O = mabs_H_2O * b * concenH_2O
    #print(aH_2O)
    nmH_2O = (wH_2O / H_2O)
    #print(nmH_2O)
    nmrH_2O = nmH_2O * N_A
    #print(nmrH_2O)
    n_293_H_2O = 1.000261 # Refractive index of H_2O (Water vapor) when T = 20 C or 293 K
    delta_T = T - 293
    n_H_2O = ((3.7 * 10 ** -6) * delta_T) + n_293_H_2O
    #print(n_H_2O)
    t_H_2O = b / (C * n_H_2O)
    #print(t_H_2O)
    numPhAb_H_2O = I_0 * (lambda_var / (h * C)) / t_H_2O
    #print(numPhAb_H_2O)
    qeffH_2O = (nmrH_2O / numPhAb_H_2O)
    #print(qeffH_2O)
    fluorH_2O = I_0 * qeffH_2O * aH_2O
    #print(fluorH_2O)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
        (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    plt.plot(Pemission, fluorH_2O, '-', markeredgewidth = '0.5', label= 'Water ---------------->
Vapor\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wH_2O) + ' (gm), ' +
'A = ' + str(int(mabs_H_2O)))
    plt.xlabel('Pemission')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_Pemission_CH():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCH = var_wCH.get()
    #print(wCH)
    b = var_b.get()
    #print(b)
    concenCH = (wCH / CH) * (22.414 / 1.6)
    #print(concenCH)
    mabs_CH = var_mabs_CH.get()
    aCH = mabs_CH * b * concenCH
    #print(aCH)
    nmCH = (wCH / CH)
    #print(nmCH)
    nmrCH = nmCH * N_A
    #print(nmrCH)
    n_293_CH = 1.000586   # Refractive index of (Methylidyne radical or hydridocarbon(â€¢)
CH) when T = 20 C or 293 K
    delta_T = T - 293
    n_CH = ((3.7 * 10 ** -6) * delta_T) + n_293_CH
    #print(n_CH)
    t_CH = b / (C * n_CH)
    #print(t_CH)
    numPhAb_CH = I_0 * (lambda_var / (h * C)) / t_CH
    #print(numPhAb_CH)
    qeffCH = (nmrCH / numPhAb_CH)
    #print(qeffCH)
    fluorCH = I_0 * qeffCH * aCH
    #print(fluorCH)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
        (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    plt.plot(Pemission, fluorCH, '-', markeredgewidth = '0.5', label= 'Hydridocarbon ------->
Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCH) + ' (gm), ' + 'A = '
+ str(int(mabs_CH)))
    plt.xlabel('Pemission')
    plt.ylabel('Fluorescence')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()

def plot_lifetime_CO_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCO_2 = var_wCO_2.get()
    #print(wCO_2)
    b = var_b.get()
    #print(b)
    concenCO_2 = (wCO_2 / CO_2) * (22.414 / 1.6)
    #print(concenCO_2)
    mabs_CO_2 = var_mabs_CO_2.get()
    aCO_2 = mabs_CO_2 * b * concenCO_2
    #print(aCO_2)
    nmCO_2 = (wCO_2 / CO_2)
    #print(nmCO_2)
    nmrCO_2 = nmCO_2 * N_A
    #print(nmrCO_2)
    n_293_CO_2 = 1.000449 # Refractive index of CO_2 (Carbon Dioxide - gas) when T = 20 C
or 293 K
    delta_T = T - 293
    n_CO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_CO_2
    #print(n_CO_2)
    t_CO_2 = b / (C * n_CO_2)
    #print(t_CO_2)
    numPhAb_CO_2 = I_0 * (lambda_var / (h * C)) / t_CO_2
    #print(numPhAb_CO_2)
    qeffCO_2 = (nmrCO_2 / numPhAb_CO_2)
    #print(qeffCO_2)
    fluorCO_2 = I_0 * qeffCO_2 * aCO_2
    #print(fluorCO_2)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
        (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    Pcompress = (4 * sigma / 3 * C) * (T ** 4)
    #print(Pcompress)
    R_specific_CO_2 = (8.31432 * 10 ** 3) / CO_2
    #print(R_specific_CO_2)
    vis_CO_2 = vis_o_CO_2 + (alpha * T) - (beta * (T ** 2))
    #print(vis_CO_2)
    Kq_CO_2 = 8 * R_specific_CO_2 * T / (3 * vis_CO_2)
    #print(Kq_CO_2)
    Kf_CO_2 = qeffCO_2 * Kq_CO_2
    #print(Kf_CO_2)
    lifetime_CO_2 = 1 / (Kf_CO_2 + ((K_B * T) / 4 * (r_CO_2 ** 3) * vis_CO_2))
    #print(lifetime_CO_2)

    plt.plot((lambda_var * (10 ** 9)), lifetime_CO_2, '-', markeredgewidth = '0.5', label= 'Carbon
dioxide -----> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO_2) +
' (gm), ' + 'A = ' + str(int(mabs_CO_2)))
    plt.xlabel('خ», nm')
    plt.ylabel('Lifetime, (s)')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lifetime_CO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCO = var_wCO.get()
    #print(wCO)
    b = var_b.get()
    #print(b)
    concenCO = (wCO / CO) * (22.414 / 1.6)
    #print(concenCO)
    mabs_CO = var_mabs_CO.get()
    aCO = mabs_CO * b * concenCO
    #print(aCO)
    nmCO = (wCO / CO)
    #print(nmCO)
    nmrCO = nmCO * N_A
    #print(nmrCO)
    n_293_CO = 1.0003364 # Refractive index of CO (Carbon Dioxide - gas) when T = 20 C or
293 K
    delta_T = T - 293
    n_CO = ((3.7 * 10 ** -6) * delta_T) + n_293_CO
    #print(n_CO)
    t_CO = b / (C * n_CO)
    #print(t_CO)
    numPhAb_CO = I_0 * (lambda_var / (h * C)) / t_CO
    #print(numPhAb_CO)
    qeffCO = (nmrCO / numPhAb_CO)
    #print(qeffCO)
    fluorCO = I_0 * qeffCO * aCO
    #print(fluorCO)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
        (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    Pcompress = (4 * sigma / 3 * C) * (T ** 4)
    #print(Pcompress)
    R_specific_CO = (8.31432 * 10 ** 3) / CO
    #print(R_specific_CO)
    vis_CO = vis_o_CO + (alpha * T) - (beta * (T ** 2))
    #print(vis_CO)
    Kq_CO = 8 * R_specific_CO * T / (3 * vis_CO)
    #print(Kq_CO)
    Kf_CO = qeffCO * Kq_CO
    #print(Kf_CO)
    lifetime_CO = 1 / (Kf_CO + ((K_B * T) / 4 * (r_CO ** 3) * vis_CO))
    #print(lifetime_CO)

    plt.plot((lambda_var * (10 ** 9)), lifetime_CO, '-', markeredgewidth = '0.5', label= 'Carbon
monoxide --> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wCO) + '
(gm), ' + 'A = ' + str(int(mabs_CO)))
    plt.xlabel('خ», nm')
    plt.ylabel('Lifetime, (s)')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lifetime_N_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wN_2 = var_wN_2.get()
    #print(wN_2)
    b = var_b.get()
    #print(b)
    concenN_2 = (wN_2 / N_2) * (22.414 / 1.6)
    #print(concenN_2)
    mabs_N_2 = var_mabs_N_2.get()
    aN_2 = mabs_N_2 * b * concenN_2
    #print(aN_2)
    nmN_2 = (wN_2 / N_2)
    #print(nmN_2)
    nmrN_2 = nmN_2 * N_A
    #print(nmrN_2)
    n_293_N_2 = 1.000297  # Refractive index of N_2 (Nitrogen) when T = 20 C or 293 K
    delta_T = T - 293
    n_N_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_N_2
    #print(n_N_2)
    t_N_2 = b / (C * n_N_2)
    #print(t_N_2)
    numPhAb_N_2 = I_0 * (lambda_var / (h * C)) / t_N_2
    #print(numPhAb_N_2)
    qeffN_2 = (nmrN_2 / numPhAb_N_2)
    #print(qeffN_2)
    fluorN_2 = I_0 * qeffN_2 * aN_2
    #print(fluorN_2)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
          (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    Pcompress = (4 * sigma / 3 * C) * (T ** 4)
    #print(Pcompress)
    R_specific_N_2 = (8.31432 * 10 ** 3) / N_2
    #print(R_specific_N_2)
    vis_N_2 = vis_o_N_2 + (alpha * T) - (beta * (T ** 2))
    #print(vis_N_2)
    Kq_N_2 = 8 * R_specific_N_2 * T / (3 * vis_N_2)
    #print(Kq_N_2)
    Kf_N_2 = qeffN_2 * Kq_N_2
    #print(Kf_N_2)
    lifetime_N_2 = 1 / (Kf_N_2 + ((K_B * T) / 4 * (r_N_2 ** 3) * vis_N_2))
    #print(lifetime_N_2)

    plt.plot((lambda_var * (10 ** 9)), lifetime_N_2, '-', markeredgewidth = '0.5', label= 'Nitrogen
---------------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wN_2) + '
(gm), ' + 'A = ' + str(int(mabs_N_2)))
    plt.xlabel('خ», nm')
    plt.ylabel('Lifetime, (s)')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lifetime_NO():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO = var_wNO.get()
    #print(wNO)
    b = var_b.get()
    #print(b)
    concenNO = (wNO / NO) * (22.414 / 1.6)
    #print(concenNO)
    mabs_NO = var_mabs_NO.get()
    aNO = mabs_NO * b * concenNO
    #print(aNO)
    nmNO = (wNO / NO)
    #print(nmNO)
    nmrNO = nmNO * N_A
    #print(nmrNO)
    n_293_NO = 1.0002697  # Refractive index of NO (Nitric oxide) when T = 20 C or 293 K
    delta_T = T - 293
    n_NO = ((3.7 * 10 ** -6) * delta_T) + n_293_NO
    #print(n_NO)
    t_NO = b / (C * n_NO)
    #print(t_NO)
    numPhAb_NO = I_0 * (lambda_var / (h * C)) / t_NO
    #print(numPhAb_NO)
    qeffNO = (nmrNO / numPhAb_NO)
    #print(qeffNO)
    fluorNO = I_0 * qeffNO * aNO
    #print(fluorNO)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
          (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)
    R_specific_NO = (8.31432 * 10 ** 3) / NO
    #print(R_specific_NO)
    vis_NO = vis_o_NO + (alpha * T) - (beta * (T ** 2))
    #print(vis_NO)
    Kq_NO = 8 * R_specific_NO * T / (3 * vis_NO)
    #print(Kq_NO)
    Kf_NO = qeffNO * Kq_NO
    #print(Kf_NO)
    lifetime_NO = 1 / (Kf_NO + ((K_B * T) / 4 * (r_NO ** 3) * vis_NO))
    #print(lifetime_NO)

    plt.plot((lambda_var * (10 ** 9)), lifetime_NO, '-', markeredgewidth = '0.5', label= 'Nitric
oxide -----------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wNO) +
' (gm), ' + 'A = ' + str(int(mabs_NO)))
    plt.xlabel('خ», nm')
    plt.ylabel('Lifetime, (s)')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lifetime_NO_2():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wNO_2 = var_wNO_2.get()
    #print(wNO_2)
    b = var_b.get()
    #print(b)
    concenNO_2 = (wNO_2 / NO_2) * (22.414 / 1.6)
    #print(concenNO_2)
    mabs_NO_2 = var_mabs_NO_2.get()
    aNO_2 = mabs_NO_2 * b * concenNO_2
    #print(aNO_2)
    nmNO_2 = (wNO_2 / NO_2)
    #print(nmNO_2)
    nmrNO_2 = nmNO_2 * N_A
    #print(nmrNO_2)
    n_293_NO_2 = 1.000516 # Refractive index of NO_2 (Nitrous oxide or Nitrogen dioxide)
when T = 20 C or 293 K
    delta_T = T - 293
    n_NO_2 = ((3.7 * 10 ** -6) * delta_T) + n_293_NO_2
    #print(n_NO_2)
    t_NO_2 = b / (C * n_NO_2)
    #print(t_NO_2)
    numPhAb_NO_2 = I_0 * (lambda_var / (h * C)) / t_NO_2
    #print(numPhAb_NO_2)
    qeffNO_2 = (nmrNO_2 / numPhAb_NO_2)
    #print(qeffNO_2)
    fluorNO_2 = I_0 * qeffNO_2 * aNO_2
    #print(fluorNO_2)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
        (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    Pcompress = (4 * sigma / 3 * C) * (T ** 4)
    #print(Pcompress)
    R_specific_NO_2 = (8.31432 * 10 ** 3) / NO_2
    #print(R_specific_NO_2)
    vis_NO_2 = vis_o_NO_2 + (alpha * T) - (beta * (T ** 2))
    #print(vis_NO_2)
    Kq_NO_2 = 8 * R_specific_NO_2 * T / (3 * vis_NO_2)
    #print(Kq_NO_2)
    Kf_NO_2 = qeffNO_2 * Kq_NO_2
    #print(Kf_NO_2)
    lifetime_NO_2 = 1 / (Kf_NO_2 + ((K_B * T) / 4 * (r_NO_2 ** 3) * vis_NO_2))
    #print(lifetime_NO_2)

    plt.plot((lambda_var * (10 ** 9)), lifetime_NO_2, '-', markeredgewidth = '0.5', label=
'Nitrogen dioxide ----> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' +
str(wNO_2) + ' (gm), ' + 'A = ' + str(int(mabs_NO_2)))
    plt.xlabel('ځ», nm')
    plt.ylabel('Lifetime, (s)')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lifetime_H_2O():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wH_2O = var_wH_2O.get()
    #print(wH_2O)
    b = var_b.get()
    #print(b)
    concenH_2O = (wH_2O / H_2O) * (22.414 / 1.6)
    #print(concenH_2O)
    mabs_H_2O = var_mabs_H_2O.get()
    aH_2O = mabs_H_2O * b * concenH_2O
    #print(aH_2O)
    nmH_2O = (wH_2O / H_2O)
    #print(nmH_2O)
    nmrH_2O = nmH_2O * N_A
    #print(nmrH_2O)
    n_293_H_2O = 1.000261 # Refractive index of H_2O (Water vapor) when T = 20 C or 293 K
    delta_T = T - 293
    n_H_2O = ((3.7 * 10 ** -6) * delta_T) + n_293_H_2O
    #print(n_H_2O)
    t_H_2O = b / (C * n_H_2O)
    #print(t_H_2O)
    numPhAb_H_2O = I_0 * (lambda_var / (h * C)) / t_H_2O
    #print(numPhAb_H_2O)
    qeffH_2O = (nmrH_2O / numPhAb_H_2O)
    #print(qeffH_2O)
    fluorH_2O = I_0 * qeffH_2O * aH_2O
    #print(fluorH_2O)

    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
         (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    Pcompress = (4 * sigma / 3 * C) * (T ** 4)
    #print(Pcompress)
    R_specific_H_2O = (8.31432 * 10 ** 3) / H_2O
    #print(R_specific_H_2O)
    vis_H_2O = vis_o_H_2O + (alpha * T) - (beta * (T ** 2))
    #print(vis_H_2O)
    Kq_H_2O = 8 * R_specific_H_2O * T / (3 * vis_H_2O)
    #print(Kq_H_2O)
    Kf_H_2O = qeffH_2O * Kq_H_2O
    #print(Kf_H_2O)
    lifetime_H_2O = 1 / (Kf_H_2O + ((K_B * T) / 4 * (r_H_2O ** 3) * vis_H_2O))
    #print(lifetime_H_2O)

    plt.plot((lambda_var * (10 ** 9)), lifetime_H_2O, '-', markeredgewidth = '0.5', label= 'Water -
---------------> Vapor\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' + str(wH_2O)
+ ' (gm), ' + 'A = ' + str(int(mabs_H_2O)))
    plt.xlabel(' nm')
    plt.ylabel('Lifetime, (s)')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()


def plot_lifetime_CH():
    T = var_temperature.get()
    I_0 = ((2 * (v ** 2) * K_B * T)) / (C ** 2)
    #print(I_0)
    wCH = var_wCH.get()
    #print(wCH)
    b = var_b.get()
    #print(b)
    concenCH = (wCH / CH) * (22.414 / 1.6)
    #print(concenCH)
    mabs_CH = var_mabs_CH.get()
    aCH = mabs_CH * b * concenCH
    #print(aCH)
    nmCH = (wCH / CH)
    #print(nmCH)
    nmrCH = nmCH * N_A
    #print(nmrCH)
    n_293_CH = 1.000586   # Refractive index of (Methylidyne radical or hydridocarbon(â€¢)
CH) when T = 20 C or 293 K
    delta_T = T - 293
    n_CH = ((3.7 * 10 ** -6) * delta_T) + n_293_CH
    #print(n_CH)
    t_CH = b / (C * n_CH)
    #print(t_CH)
    numPhAb_CH = I_0 * (lambda_var / (h * C)) / t_CH
    #print(numPhAb_CH)
    qeffCH = (nmrCH / numPhAb_CH)
    #print(qeffCH)
    fluorCH = I_0 * qeffCH * aCH
    #print(fluorCH)
    BbR = ((2 * np.pi * h * (C ** 2)) / (lambda_var ** 5)) * \
          (1 / ((np.exp((h * C) / (lambda_var * K_B * T))) - 1))
    #print(BbR)
    Emissivity = E / BbR
    #print(Emissivity)
    Pemission = (Emissivity * sigma / C) * (T ** 4)
    #print(Pemission)

    Pcompress = (4 * sigma / 3 * C) * (T ** 4)
    #print(Pcompress)
    R_specific_CH = (8.31432 * 10 ** 3) / CH
    #print(R_specific_CH)
    vis_CH = vis_o_CH + (alpha * T) - (beta * (T ** 2))
    #print(vis_CH)
    Kq_CH = 8 * R_specific_CH * T / (3 * vis_CH)
    #print(Kq_CH)
    Kf_CH = qeffCH * Kq_CH
    #print(Kf_CH)
    lifetime_CH = 1 / (Kf_CH + ((K_B * T) / 4 * (r_CH ** 3) * vis_CH))
    #print(lifetime_CH)

    plt.plot((lambda_var * (10 ** 9)), lifetime_CH, '-', markeredgewidth = '0.5', label=
'Hydridocarbon -------> Gas\nT = ' + str(int(T)) + ' (K), ' + 'b = ' + str(b) + ' (m),' + '\nW = ' +
str(wCH) + ' (gm), ' + 'A = ' + str(int(mabs_CH)))
    plt.xlabel('خ», nm')
    plt.ylabel('Lifetime, (s)')
    plt.legend()
    plt.grid(color='b', linestyle='-', linewidth=0.1)
    plt.show()
=========================================================================
End plots
=========================================================================


def sub_window_1():
    sub_window_1 = Toplevel(window)
    sub_window_1.wm_title("Laser-induced fluorescence calculations - Entry Data & Build
Figures")

    sub_window_1.tk.call('wm', 'iconphoto', sub_window_1._w, icon)


    label_1 = Label(sub_window_1, text="Temperature (T): from 980 to 180 K")
    label_1.grid(row = 0, column = 0, padx = 0, pady = 0)

    scale_1 = Scale(sub_window_1, from_=980, to=1180,tickinterval=20,
orient=HORIZONTAL, length = 300, variable = var_temperature)
    scale_1.grid(row = 1, column = 0, padx = 0, pady = 0)

    label_2 = Label(sub_window_1, text="Vacuum Chamber Thickness (b), m")
    label_2.grid(row = 0, column = 1, padx = 0, pady = 0)

    scale_2 = Scale(sub_window_1, from_=0.01, to=0.05 ,resolution=0.01, tickinterval=0.01,
orient=HORIZONTAL, length = 300, variable = var_b)
    scale_2.grid(row = 1, column = 1, padx = 0, pady = 0)

    label_3 = Label(sub_window_1, text="Show figures")
    label_3.grid(row = 1, column = 4, padx = 0, pady = 0)
    b4 = Button(sub_window_1, text="Figure ( Intensity)", command=plot_lambda_Io, width =
24, height = 2)
    b4.grid(row = 1, column = 2, padx = 5, pady = 0)


    label_4 = Label(sub_window_1, text="The weight ratio of CO_2 (W), gm")
    label_4.grid(row = 2, column = 0, padx = 0, pady = 0)

    scale_3 = Scale(sub_window_1, from_=0.03, to=0.18 ,resolution=0.03, tickinterval=0.03,
orient=HORIZONTAL, length = 300, variable = var_wCO_2)
    scale_3.grid(row = 3, column = 0, padx = 0, pady = 0)

    label_5 = Label(sub_window_1, text="The molar absorptivity of CO_2 (A), 1/m")
    label_5.grid(row = 2, column = 1, padx = 0, pady = 0)

    scale_4 = Scale(sub_window_1, from_=3 * 10 ** 5, to=3 * 10 ** 8 ,resolution=1 * 10 ** 1,
tickinterval=0.70 * 10 ** 8, orient=HORIZONTAL, length = 300, variable = var_mabs_CO_2)
    scale_4.grid(row = 3, column = 1, padx = 0, pady = 0)

    label_6 = Label(sub_window_1, text="CO_2")
    label_6.grid(row = 2, column = 2, padx = 0, pady = 0)

    b5 = Button(sub_window_1, text="Figure (quantum efficiency)",
command=plot_lambda_qeffCO_2, width = 24, height = 2)
    b5.grid(row = 3, column = 2, padx = 5, pady = 0)


    label_7 = Label(sub_window_1, text="The weight ratio of CO (W), gm")
    label_7.grid(row = 4, column = 0, padx = 0, pady = 0)

    scale_5 = Scale(sub_window_1, from_=0.03, to=0.18 ,resolution=0.03, tickinterval=0.03,
orient=HORIZONTAL, length = 300, variable = var_wCO)
    scale_5.grid(row = 5, column = 0, padx = 0, pady = 0)

    label_8 = Label(sub_window_1, text="The molar absorptivity of CO (A), 1/m")
    label_8.grid(row = 4, column = 1, padx = 0, pady = 0)

    scale_6 = Scale(sub_window_1, from_=3 * 10 ** 5, to=3 * 10 ** 8 ,resolution=1 * 10 ** 1,
tickinterval=0.70 * 10 ** 8, orient=HORIZONTAL, length = 300, variable = var_mabs_CO)
    scale_6.grid(row = 5, column = 1, padx = 0, pady = 0)

    label_9 = Label(sub_window_1, text="CO")
    label_9.grid(row = 4, column = 2, padx = 0, pady = 0)

    b6 = Button(sub_window_1, text="Figure (quantum efficiency)",
command=plot_lambda_qeffCO, width = 24, height = 2)
    b6.grid(row = 5, column = 2, padx = 5, pady = 0)


    label_8 = Label(sub_window_1, text="The weight ratio of N_2 (W), gm")
    label_8.grid(row = 6, column = 0, padx = 0, pady = 0)

    scale_7 = Scale(sub_window_1, from_=0.03, to=0.18 ,resolution=0.03, tickinterval=0.03,
orient=HORIZONTAL, length = 300, variable = var_wN_2)
    scale_7.grid(row = 7, column = 0, padx = 0, pady = 0)

    label_9 = Label(sub_window_1, text="The molar absorptivity of N_2 (A), 1/m")
    label_9.grid(row = 6, column = 1, padx = 0, pady = 0)
    scale_8 = Scale(sub_window_1, from_=3 * 10 ** 5, to=3 * 10 ** 8 ,resolution=1 * 10 ** 1,
tickinterval=0.70 * 10 ** 8, orient=HORIZONTAL, length = 300, variable = var_mabs_N_2)
    scale_8.grid(row = 7, column = 1, padx = 0, pady = 0)

    label_10 = Label(sub_window_1, text="N_2")
    label_10.grid(row = 6, column = 2, padx = 0, pady = 0)

    b7 = Button(sub_window_1, text="Figure ( quantum efficiency)",
command=plot_lambda_qeffN_2, width = 24, height = 2)
    b7.grid(row = 7, column = 2, padx = 5, pady = 0)


    label_11 = Label(sub_window_1, text="The weight ratio of NO (W), gm")
    label_11.grid(row = 8, column = 0, padx = 0, pady = 0)

    scale_9 = Scale(sub_window_1, from_=0.03, to=0.18 ,resolution=0.03, tickinterval=0.03,
orient=HORIZONTAL, length = 300, variable = var_wNO)
    scale_9.grid(row = 9, column = 0, padx = 0, pady = 0)

    label_12 = Label(sub_window_1, text="The molar absorptivity of NO (A), 1/m")
    label_12.grid(row = 8, column = 1, padx = 0, pady = 0)

    scale_10 = Scale(sub_window_1, from_=3 * 10 ** 5, to=3 * 10 ** 8 ,resolution=1 * 10 ** 1,
tickinterval=0.70 * 10 ** 8, orient=HORIZONTAL, length = 300, variable = var_mabs_NO)
    scale_10.grid(row = 9, column = 1, padx = 0, pady = 0)

    label_13 = Label(sub_window_1, text="NO")
    label_13.grid(row = 8, column = 2, padx = 0, pady = 0)

    b8 = Button(sub_window_1, text="Figure ( quantum efficiency)",
command=plot_lambda_qeffNO, width = 24, height = 2)
    b8.grid(row = 9, column = 2, padx = 5, pady = 0)


    label_14 = Label(sub_window_1, text="The weight ratio of NO_2 (W), gm")
    label_14.grid(row = 10, column = 0, padx = 0, pady = 0)

    scale_11 = Scale(sub_window_1, from_=0.03, to=0.18 ,resolution=0.03, tickinterval=0.03,
orient=HORIZONTAL, length = 300, variable = var_wNO_2)
    scale_11.grid(row = 11, column = 0, padx = 0, pady = 0)

    label_15 = Label(sub_window_1, text="The molar absorptivity of NO_2 (A), 1/m")
    label_15.grid(row = 10, column = 1, padx = 0, pady = 0)

    scale_12 = Scale(sub_window_1, from_=3 * 10 ** 5, to=3 * 10 ** 8 ,resolution=1 * 10 ** 1,
tickinterval=0.70 * 10 ** 8, orient=HORIZONTAL, length = 300, variable = var_mabs_NO_2)
    scale_12.grid(row = 11, column = 1, padx = 0, pady = 0)

    label_16 = Label(sub_window_1, text="NO_2")
    label_16.grid(row = 10, column = 2, padx = 0, pady = 0)

    b9 = Button(sub_window_1, text="Figure ( quantum efficiency)",
command=plot_lambda_qeffNO_2, width = 24, height = 2)
    b9.grid(row = 11, column = 2, padx = 5, pady = 0)


    label_17 = Label(sub_window_1, text="The weight ratio of H_2O (W), gm")
    label_17.grid(row = 12, column = 0, padx = 0, pady = 0)
```

```
   scale_13 = Scale(sub_window_1, from_=0.03, to=0.18 ,resolution=0.03, tickinterval=0.03,
orient=HORIZONTAL, length = 300, variable = var_wH_2O)
   scale_13.grid(row = 13, column = 0, padx = 0, pady = 0)

   label_18 = Label(sub_window_1, text="The molar absorptivity of H_2O (A), 1/m")
   label_18.grid(row = 12, column = 1, padx = 0, pady = 0)

   scale_14 = Scale(sub_window_1, from_=3 * 10 ** 5, to=3 * 10 ** 8 ,resolution=1 * 10 ** 1,
tickinterval=0.70 * 10 ** 8, orient=HORIZONTAL, length = 300, variable = var_mabs_H_2O)
   scale_14.grid(row = 13, column = 1, padx = 0, pady = 0)

   label_19 = Label(sub_window_1, text="H_2O")
   label_19.grid(row = 12, column = 2, padx = 0, pady = 0)

   b10 = Button(sub_window_1, text="Figure ( quantum efficiency)",
command=plot_lambda_qeffH_2O, width = 24, height = 2)
   b10.grid(row = 13, column = 2, padx = 5, pady = 0)


   label_20 = Label(sub_window_1, text="The weight ratio of CH (W), gm")
   label_20.grid(row = 14, column = 0, padx = 0, pady = 0)

   scale_15 = Scale(sub_window_1, from_=0.03, to=0.18 ,resolution=0.03, tickinterval=0.03,
orient=HORIZONTAL, length = 300, variable = var_wCH)
   scale_15.grid(row = 15, column = 0, padx = 0, pady = 0)

   label_21 = Label(sub_window_1, text="The molar absorptivity of CH (A), 1/m")
   label_21.grid(row = 14, column = 1, padx = 0, pady = 0)

   scale_16 = Scale(sub_window_1, from_=3 * 10 ** 5, to=3 * 10 ** 8 ,resolution=1 * 10 ** 1,
tickinterval=0.70 * 10 ** 8, orient=HORIZONTAL, length = 300, variable = var_mabs_CH)
   scale_16.grid(row = 15, column = 1, padx = 0, pady = 0)

   label_22 = Label(sub_window_1, text="CH")
   label_22.grid(row = 14, column = 2, padx = 0, pady = 0)

   b11 = Button(sub_window_1, text="Figure ( quantum efficiency)",
command=plot_lambda_qeffCH, width = 24, height = 2)
   b11.grid(row = 15, column = 2, padx = 5, pady = 0)


   #-----------------------------------------------------------------------------------------------------#
   label_22 = Label(sub_window_1, text="خ»max")
   label_22.grid(row = 1, column = 3, padx = 0, pady = 0)

   scale_16 = Scale(sub_window_1, from_=400, to=700, resolution=50, tickinterval=100,
orient=HORIZONTAL, length = 175, variable = var_lambda_max)
   scale_16.set(500)
   scale_16.grid(row = 1, column = 4, padx = 0, pady = 0)
   label_23 = Label(sub_window_1, text="CO_2")
   label_23.grid(row = 2, column = 3, padx = 0, pady = 0)

  command=plot_fluorCO_2, width = 24, height = 2)
   b12.grid(row = 3, column = 3, padx = 5, pady = 0)

   label_24 = Label(sub_window_1, text="CO_2")
   label_24.grid(row = 2, column = 4, padx = 0, pady = 0)

   b13 = Button(sub_window_1, text="Figure (Pemission, Fluorescence)",
command=plot_Pemission_CO_2, width = 26, height = 2)
```

```
   b13.grid(row = 3, column = 4, padx = 5, pady = 0)

   label_25 = Label(sub_window_1, text="CO_2")
   label_25.grid(row = 2, column = 5, padx = 0, pady = 0)

   b14 = Button(sub_window_1, text="Figure (خ», Lifetime)", command=plot_lifetime_CO_2,
width = 24, height = 2)
   b14.grid(row = 3, column = 5, padx = 5, pady = 0)

   label_26 = Label(sub_window_1, text="CO")
   label_26.grid(row = 4, column = 3, padx = 0, pady = 0)

   b15 = Button(sub_window_1, text="Figure (خ», Fluorescence)", command=plot_fluorCO,
width = 24, height = 2)
   b15.grid(row = 5, column = 3, padx = 5, pady = 0)

   label_27 = Label(sub_window_1, text="CO")
   label_27.grid(row = 4, column = 4, padx = 0, pady = 0)

   b16 = Button(sub_window_1, text="Figure (Pemission, Fluorescence)",
command=plot_Pemission_CO, width = 26, height = 2)
   b16.grid(row = 5, column = 4, padx = 5, pady = 0)

   label_28 = Label(sub_window_1, text="CO")
   label_28.grid(row = 4, column = 5, padx = 0, pady = 0)

   b17 = Button(sub_window_1, text="Figure ( Lifetime)", command=plot_lifetime_CO, width
= 24, height = 2)
   b17.grid(row = 5, column = 5, padx = 5, pady = 0)


   label_29 = Label(sub_window_1, text="N_2")
   label_29.grid(row = 6, column = 3, padx = 0, pady = 0)

   b18 = Button(sub_window_1, text="Figure (خ», Fluorescence)", command=plot_fluorN_2,
width = 24, height = 2)
   b18.grid(row = 7, column = 3, padx = 5, pady = 0)

   label_30 = Label(sub_window_1, text="N_2")
   label_30.grid(row = 6, column = 4, padx = 0, pady = 0)

   b19 = Button(sub_window_1, text="Figure (Pemission, Fluorescence)",
command=plot_Pemission_N_2, width = 26, height = 2)
   b19.grid(row = 7, column = 4, padx = 5, pady = 0)

   label_31 = Label(sub_window_1, text="N_2")
   label_31.grid(row = 6, column = 5, padx = 0, pady = 0)

   b20 = Button(sub_window_1, text="Figure ( Lifetime)", command=plot_lifetime_N_2,
width = 24, height = 2)
   b20.grid(row = 7, column = 5, padx = 5, pady = 0)


   label_32 = Label(sub_window_1, text="NO")
   label_32.grid(row = 8, column = 3, padx = 0, pady = 0)

   b21 = Button(sub_window_1, text="Figure (, Fluorescence)", command=plot_fluorNO,
width = 24, height = 2)
   b21.grid(row = 9, column = 3, padx = 5, pady = 0)
```

```
    label_33 = Label(sub_window_1, text="NO")
    label_33.grid(row = 8, column = 4, padx = 0, pady = 0)

    b22 = Button(sub_window_1, text="Figure (Pemission, Fluorescence)",
command=plot_Pemission_NO, width = 26, height = 2)
    b22.grid(row = 9, column = 4, padx = 5, pady = 0)

    label_34 = Label(sub_window_1, text="NO")
    label_34.grid(row = 8, column = 5, padx = 0, pady = 0)

    b23 = Button(sub_window_1, text="Figure (Lifetime)", command=plot_lifetime_NO, width
= 24, height = 2)
    b23.grid(row = 9, column = 5, padx = 5, pady = 0)

    label_35 = Label(sub_window_1, text="NO_2")
    label_35.grid(row = 10, column = 3, padx = 0, pady = 0)

    b24 = Button(sub_window_1, text="Figure (Fluorescence)", command=plot_fluorNO_2,
width = 24, height = 2)
    b24.grid(row = 11, column = 3, padx = 5, pady = 0)

    label_36 = Label(sub_window_1, text="NO_2")
    label_36.grid(row = 10, column = 4, padx = 0, pady = 0)

    b25 = Button(sub_window_1, text="Figure (Pemission, Fluorescence)",
command=plot_Pemission_NO_2, width = 26, height = 2)
    b25.grid(row = 11, column = 4, padx = 5, pady = 0)

    label_37 = Label(sub_window_1, text="NO_2")
    label_37.grid(row = 10, column = 5, padx = 0, pady = 0)

    b26 = Button(sub_window_1, text="Figure (خ», Lifetime)", command=plot_lifetime_NO_2,
width = 24, height = 2)
    b26.grid(row = 11, column = 5, padx = 5, pady = 0)

    label_38 = Label(sub_window_1, text="H_2O")
    label_38.grid(row = 12, column = 3, padx = 0, pady = 0)

    b27 = Button(sub_window_1, text="Figure ( Fluorescence)", command=plot_fluorH_2O,
width = 24, height = 2)
    b27.grid(row = 13, column = 3, padx = 5, pady = 0)

    label_39 = Label(sub_window_1, text="H_2O")
    label_39.grid(row = 12, column = 4, padx = 0, pady = 0)

    b28 = Button(sub_window_1, text="Figure (Pemission, Fluorescence)",
command=plot_Pemission_H_2O, width = 26, height = 2)
    b28.grid(row = 13, column = 4, padx = 5, pady = 0)

    label_40 = Label(sub_window_1, text="H_2O")
    label_40.grid(row = 12, column = 5, padx = 0, pady = 0)

    b29 = Button(sub_window_1, text="Figure (خ», Lifetime)", command=plot_lifetime_H_2O,
width = 24, height = 2)
    b29.grid(row = 13, column = 5, padx = 5, pady = 0)


    label_41 = Label(sub_window_1, text="CH")
    label_41.grid(row = 14, column = 3, padx = 0, pady = 0)
```

```
   b30 = Button(sub_window_1, text="Figure (Fluorescence)", command=plot_fluorCH,
width = 24, height = 2)
   b30.grid(row = 15, column = 3, padx = 5, pady = 0)

   label_42 = Label(sub_window_1, text="CH")
   label_42.grid(row = 14, column = 4, padx = 0, pady = 0)

   b31 = Button(sub_window_1, text="Figure (Pemission, Fluorescence)",
command=plot_Pemission_CH, width = 26, height = 2)
   b31.grid(row = 15, column = 4, padx = 5, pady = 0)

   label_43 = Label(sub_window_1, text="CH")
   label_43.grid(row = 14, column = 5, padx = 0, pady = 0)

   b32 = Button(sub_window_1, text="Figure (Lifetime)", command=plot_lifetime_CH, width
= 24, height = 2)
   b32.grid(row = 15, column = 5, padx = 5, pady = 0)

   # Menu sub_window_1
   menubar = Menu(sub_window_1)
   sub_window_1.configure(menu=menubar)

   filemenu = Menu(menubar, tearoff=0)
   quantum_eff = Menu(menubar, tearoff=0)
   helpmenu = Menu(menubar, tearoff=0)
   fluorescence = Menu(menubar, tearoff=0)
   pemission = Menu(menubar, tearoff=0)
   lifetime = Menu(menubar, tearoff=0)

   menubar.add_cascade(label="File",menu=filemenu)
   filemenu.add_command(label="---------------------------")
   filemenu.add_command(label="Exit", command=window.destroy)

   menubar.add_cascade(label="Quantum Efficiency",menu=quantum_eff)
   quantum_eff.add_command(label="Build figure (, Quantum Efficiency) for:")
   quantum_eff.add_command(label="-----------------------------------------")
   quantum_eff.add_command(label="Carbon dioxide", command=plot_lambda_qeffCO_2)
   quantum_eff.add_command(label="Carbon monoxide", command=plot_lambda_qeffCO)
   quantum_eff.add_command(label="Nitrogen", command=plot_lambda_qeffN_2)
   quantum_eff.add_command(label="Nitric oxide", command=plot_lambda_qeffNO)
   quantum_eff.add_command(label="Nitrogen dioxide", command=plot_lambda_qeffNO_2)
   quantum_eff.add_command(label="Water --> Vapor", command=plot_lambda_qeffH_2O)
   quantum_eff.add_command(label="Hydridocarbon", command=plot_lambda_qeffCH)

   menubar.add_cascade(label="Fluorescence",menu=fluorescence)
   fluorescence.add_command(label="Build figure  Fluorescence) for:")
   fluorescence.add_command(label="-----------------------------------------")
   fluorescence.add_command(label="Carbon dioxide", command=plot_fluorCO_2)
   fluorescence.add_command(label="Carbon monoxide", command=plot_fluorCO)
   fluorescence.add_command(label="Nitrogen", command=plot_fluorN_2)
   fluorescence.add_command(label="Nitric oxide", command=plot_fluorNO)
   fluorescence.add_command(label="Nitrogen dioxide", command=plot_fluorNO_2)
   fluorescence.add_command(label="Water --> Vapor", command=plot_fluorH_2O)
   fluorescence.add_command(label="Hydridocarbon", command=plot_fluorCH)

   menubar.add_cascade(label="Pemission",menu=pemission)
   pemission.add_command(label="Build figure (Pemission, Fluorescence) for:")
   pemission.add_command(label="-----------------------------------------")
   pemission.add_command(label="Carbon dioxide", command=plot_Pemission_CO_2)
   pemission.add_command(label="Carbon monoxide", command=plot_Pemission_CO)
```

```
    pemission.add_command(label="Nitrogen", command=plot_Pemission_N_2)
    pemission.add_command(label="Nitric oxide", command=plot_Pemission_NO)
    pemission.add_command(label="Nitrogen dioxide", command=plot_Pemission_NO_2)
    pemission.add_command(label="Water --> Vapor", command=plot_Pemission_H_2O)
    pemission.add_command(label="Hydridocarbon", command=plot_Pemission_CH)

    menubar.add_cascade(label="Lifetime",menu=lifetime)
    lifetime.add_command(label="Build figure (خ», Lifetime) for:")
    lifetime.add_command(label="----------------------------------------")
    lifetime.add_command(label="Carbon dioxide", command=plot_lifetime_CO_2)
    lifetime.add_command(label="Carbon monoxide", command=plot_lifetime_CO)
    lifetime.add_command(label="Nitrogen", command=plot_lifetime_N_2)
    lifetime.add_command(label="Nitric oxide", command=plot_lifetime_NO)
    lifetime.add_command(label="Nitrogen dioxide", command=plot_lifetime_NO_2)
    lifetime.add_command(label="Water --> Vapor", command=plot_lifetime_H_2O)
    lifetime.add_command(label="Hydridocarbon", command=plot_lifetime_CH)

    menubar.add_cascade(label="Help",menu=helpmenu)
    helpmenu.add_command(label="About the Program", command=sub_window_2)


b1 = Button(window, text="Entry Data & Build Figures", command=sub_window_1, width =
24, height = 2)
b1.grid(row = 1, column = 2, padx = 20, pady = 90)


def sub_window_2():
    sub_window_2 = Toplevel(window)
    sub_window_2.wm_title("Laser-induced fluorescence calculations - About the Program")
    sub_window_2.geometry('570x200')
    sub_window_2.configure(background="#E5E5CC")

    sub_window_2.tk.call('wm', 'iconphoto', sub_window_2._w, icon)

    label_1 = Label(sub_window_2, text="Laser-induced fluorescence calculations",
anchor="center")
    label_1.grid(row = 0, column = 0, padx = 150, pady = 10)
    label_1.configure(background="#E5E5CC")

    label_2 = Label(sub_window_2, text="Version 1.0.0", anchor="center")
    label_2.grid(row = 1, column = 0, padx = 150, pady = 0)
    label_2.configure(background="#E5E5CC")

    label_3 = Label(sub_window_2, text="Author: Mustafa Mohamed Badreldin Mirghani",
fg="green", cursor="dotbox", anchor="center")
    label_3.grid(row = 2, column = 0, padx = 150, pady = 10)
    label_3.configure(background="#E5E5CC")

    def callback(event):
        webbrowser.open_new(r"http://www.mutawirsoft.com")

    label_4 = Label(sub_window_2, text="Built by Montasir Mirghani", fg="blue",
cursor="hand2", anchor="center")
    label_4.grid(row = 3, column = 0, padx = 150, pady = 0)
    label_4.bind("<Button-1>", callback)
    label_4.configure(background="#E5E5CC")

    label_5 = Label(sub_window_2, text="All rights reserved - anchor="center")
    label_5.grid(row = 4, column = 0, padx = 150, pady = 10)
    label_5.configure(background="#E5E5CC")
    # Menu sub_window_2
    menubar = Menu(sub_window_2)
    sub_window_2.configure(menu=menubar)

    filemenu = Menu(menubar, tearoff=0, background="#E5E5CC")

    menubar.add_cascade(label="File",menu=filemenu)
    filemenu.add_command(label="---------------------------")
    filemenu.add_command(label="Exit", command=window.destroy)

b2 = Button(window, text="About the Program", command=sub_window_2, width = 24,
height = 2)
b2.grid(row = 1, column = 3, padx = 0, pady = 90)


def exit_from_app():
    window.destroy()

b3 = Button(window, text="Exit", command=exit_from_app, width = 24, height = 2)
b3.grid(row = 1, column = 4, padx = 20, pady = 90)

# Main Menu
menubar = Menu(window)
window.configure(menu=menubar)

filemenu = Menu(menubar, tearoff=0, background="#E5E5CC")
helpmenu = Menu(menubar, tearoff=0, background="#E5E5CC")

menubar.add_cascade(label="File",menu=filemenu)
filemenu.add_command(label="Entry Data & Build Figures", command=sub_window_1)
filemenu.add_command(label="---------------------------")
filemenu.add_command(label="Exit", command=window.destroy)

menubar.add_cascade(label="Help",menu=helpmenu)
helpmenu.add_command(label="About the Program", command=sub_window_2)

window.mainloop()
```