Master's Thesis

Optimization of High Harmonic Generation Using 400 nm and 800 nm driving Laser Beam

Abdullah Abduljaleel

Humboldt Universität zu Berlin

Supervisors
Prof. Dr. Kurt Busch
Dr. Oleg Kornilov

Masterarbeit

Zur Erlangung des akademischen Masterstudiengang Optische Wissenschaften (Master program of optical sciences).

Thema der Arbeit : Optimierung der Hochharmonischen Erzeugung unter Verwendung von 400 nm und 800 nm treibendem Laserstrahl - (Optimization of High Harmonic Generation Using 400 nm and 800 nm driving Laser Beam)

eingereicht von: Name der Verfasserin / des Verfassers

Gutachter/innen:

Titel, Name

Titel, Name

Eingereicht am Institut für Physik der Humboldt-Universität zu Berlin am:

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Abdullah Abduljaleel,

Berlin, 20th of October, 2023

Zusammenfassung

Seit der Entdeckung der Hochharmonischen Erzeugung (HHG) und insbesondere nach der Verleihung des Nobelpreises für Physik im Jahr 2023 für die Herstellung von Attosekundenpulsen hat sich das Interesse an der Erzeugung kürzerer Pulse, die hauptsächlich durch HHG erzeugt werden, zunehmend verstärkt. Diese Aufmerksamkeit resultiert aus den leistungsstarken Werkzeugen, die ultrakurze Pulse bieten, um verschiedene komplexe Phänomene in der Natur zu untersuchen. Die HHG-Experimente und die Erzeugung ultrakurzer Pulse sind untrennbar mit der Herausforderung der Phasenanpassung zwischen den Ausbreitungseffekten verbunden. Lichtquellen bei 800 nm und 400 nm sind weit verbreitet und bieten Eigenschaften, die HHG-Experimente erheblich erleichtern.

In dieser Arbeit präsentieren wir zwei Experimente. Das erste Experiment untersucht die Phasenanpassung der HHG unter Verwendung einer bereits vorhandenen Einrichtung bei 800 nm. In diesem Experiment haben wir die Phasenanpassungsbedingungen bei verschiedenen Drücken des Umwandlungsmediums und verschiedenen Werten der Anregungspulsenergien untersucht. Im zweiten Experiment haben wir die Phasenanpassung der HHG unter Verwendung einer 400 nm Lichtquelle untersucht und mit der Phasenanpassung bei 800 nm verglichen. Dabei haben wir verschiedene Werte der Gruppenlaufzeitdispersion (GDD) und Werte der Dispersion dritter Ordnung (TOD) auf den Anregungslaserpuls untersucht, zusätzlich zur Untersuchung der Phasenanpassungsbedingungen und der Anregungspulsenergien.

Als Ergebnis zeigt diese Arbeit die optimalen Phasenanpassungsbedingungen sowohl für die 800 nm als auch für die 400 nm Lichtquellen. Durch Kurven, Phasenanpassungskarten und Diskussionen zeigen wir das Verhalten verschiedener Spektren mit den verschiedenen zuvor genannten Parametern und mögliche Gründe für dieses Verhalten. Das 400 nm Licht zeigt geringere Photonenergien und erfordert niedrigere Druckbedingungen im Vergleich zur 800 nm Lichtquelle.

Zusammenfassend liefert die 800 nm Lichtquelle höhere erhaltene Photonenergien. Die Parameter, die in den HHG-Experimenten bei 800 nm verwendet wurden, bieten flexible Freiheitsgrade zur Erzeugung von HHG im Vergleich zur 400 nm Lichtquelle, und die Experimente mit der 800 nm Lichtquelle zeigen eine stärkere Übereinstimmung mit den theoretischen Vorhersagen auf eine beobachtbarere Weise.

Abstract

Since the discovery of the high harmonic generation (HHG), and especially with the 2023 Nobel Prize in physics awarded for attoseconds pulses production, there has been a growing focus on the efforts to produce shorter pulses, primarily produced by the HHG. This attention is driven by the powerful tools that the ultrashort pulses provide in investigating different ambiguous phenomenon in nature. The HHG experiments and the production of ultrashort pulses are intrinsically linked to the challenge of phase matching among propagation effects. Light sources of 800 nm and 400 nm are ubiquitous and offer characteristics that greatly facilitate HHG experiments.

In this thesis we present two experiments. The first experiment investigates the phase matching of HHG using an already available 800 nm setup. In this experiment we investigated the phase matching conditions across various values pressure of conversion medium and various values of driving pulse energies. In the second experiment we investigated the phase matching of HHG using 400 nm source and we compare it with phase matching of the 800 nm again by investigating various values of the group delay dispersion (GDD) and values of the third order dispersion (TOD) applied to the driving laser pulse, additional to investigating the phase matching conditions and the driving pulse energies.

As result, this thesis demonstrate the optimum phases matching conditions for both the 800 nm and the 400 nm sources. Through curves, phase matching maps, and discussions we demonstrate the behavior of different spectra with various recent mentioned parameters the potential reason underling these behaviors. The 400 nm light demonstrates less photon energies and lower required pressure conditions with respect to the 800 nm source.

In conclusion, the 800 nm source provides higher obtained photon energies. The parameters used in the HHG from 800 nm experiments gives flexible degrees of freedom for achieving HHG with respect to 400 nm source and experiments of the 800 nm source show agreement with the theoretical predictions in more observable manner.

Acknowledgment

Expressing gratitude and not denying favours to those to whom they belong lie at the core of my beliefs as an individual. Prophet Muhammad, peace and blessings be upon him, said, "He who does not thank people does not thank God". Therefore, I cannot undertake the task of delivering this thesis or writing acknowledgments without expressing the following:

To Professor Dr. Kurt Busch, my first supervisor, the support I received, your care, and even your stringent yet constructive criticism have guided me through the entire master's program, not only this thesis. Your professional, positive, and inspiring presence in the physics department of Humboldt University has always meant a lot to me. It is a feeling that I often share with my peers and a memory I shall forever cherish.

To my supervisor, Dr. Oleg Kornilov, this thesis would have never been the way it is without your support, surveillance, comments, and guidance. Besides, I cannot write these words without mentioning that Dr. Oleg, who had enough patience and with whom I plotted my first sine function in Python. I feel deeply fortunate and happy for being under your supervision and for what I could learn from you both on the scientific level as a scientist and supervisor and on a personal level as an elder brother.

To my supervisors, receiving support beyond one's duties is not expected, except from genuinely humane and noble souls. It is something I always witnessed from you. I'm deeply thankful to you.

For Professor Dr. Misha Ivanov, your strongly encouraging and positive presence with your well-known smile made me perceive this master's program differently and was a great motivation for me in its own right. Thank you very much.

To my dear friends and colleagues whom I am blessed to accompany and learn from: Durvasa Gupta, Sobhy Kholaif, Viktor Bender, Graham Brown, Cem Güney, Lisa-Marie Koll, and Mikhail Osolodkov. This thesis would have never been completed without your immediate responses of assistance and readiness to help without hesitation when needed. It meant a lot to me and still does. My gratitude would never be enough. Your actions always remind me of how to respond when others ask for help. Therefore, I will pay it forward.

To my beloved wife, with whom I have been fortunate to share my life for the past year, your love, care, endless support, your beautiful persona, and your patience with someone like me during the course of this thesis have helped me reach this point.

To my beloved brothers and sister, whom I never appreciated enough and whom I dearly miss, without you, I would have never come this far.

To my beloved father and beloved mother, mere words are incapable of conveying what lies within my chest. The sentiments and feelings I hold are vast, while the expressions at my disposal are limited. It is you and your prayers that have made me the person I am today.

Contents

Introduction

Light, in its various forms, and lasers, in particular, are powerful tools that find applications ranging from everyday activities to critical domains such as medicine and space applications. Within the realm of lasers, one particular application is the generation of ultrashort pulses. In order to observe any rapid phenomena in nature, a controllable tool that operates at a speed equal to or exceeding that of the phenomenon in question is necessary. This ensures better temporal resolution. Among the fastest-known phenomena in nature that are ubiquitously subject to investigation, are the atomic and molecular structure vibrations as well as the evolution of the electronic states in atoms and molecules. Hence, there emerges the need to develop shorter, well-characterised, and controllable pulses as tools to investigate such phenomena.

To observe atomic and molecular structures whose vibrational periods lie in the femtoseconds range, femtosecond (fs) laser systems can be used [1]. The time scale of electron motion between the excited states of atoms and molecules, however, lies in the subfemtosecond range. The shortest obtainable pulse durations via state-of-the-art laser systems are limited to 2.6 fs [2]. The Ti:sapphire lasers are the usual sources for such ultrashort pulses. The durations of these ultrashort pulses are not sufficient to observe the electronic evolutions of the excited states. In order to observe such rapid evolutions, one needs to use ultrashort pulses in the attoseconds range and photon energies in the extreme ultraviolet (XUV) region [3].

The term 'ultrashort pulses' refers to optical pulses with durations on the order of 10^{-12} sec, i.e., picoseconds to 10^{-18} sec, i.e., attoseconds. The shortest achieved laser pulses are in the range of attoseconds. Pulses in this range were first produced at the beginning of the previous decade by Ferenc Krausz [4] and Pascal Salieres [5], both in the same year. To get a sense of the duration of such events on this order, other events with different durations can be seen in Figure 1.

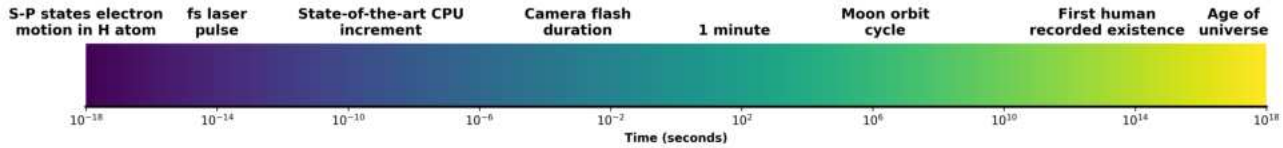


Figure 1: Time scale of various events, compared to ultrashort laser pulses.

One can think of these pulses in the quantum picture as a composition of quantised photons propagating at the speed of light in a vacuum, $c_o = 2.998 \times 10^8$ m/s, and carrying energy $E = hv$, where h is the Planck's constant and v is the frequency of the propagated photon. In the classical picture, however, one can think of a pulse as a superposition of a number of electromagnetic plane waves with different frequencies that satisfy the wave Equation, which is derived from the Maxwell equations [6, 7]. The monochromatic plane wave equation is introduced as a solution for the wave Equation ($E(x,t) = E_o \exp^{i(kx-wt)}$). The plane wave includes the amplitude E_o, the wavevector k, and the angular frequency w.

The generation of ultrashort pulses in the femtosecond range is done through optical techniques such as mode-locking and chirped pulse amplification (CPA). However, high harmonic generation is the means to generate pulses in the attosecond range [4, 8], which was awarded the Nobel Prize in 2023. Electronic devices, including superconductor circuits, are too slow to generate ultrashort pulses. The speed constraints of their electrical circuits prevent them from producing pulses in the femtosecond or attosecond range, restricting the generation to the nanosecond range. By means of mode-locking, where the phases of the propagating longitudinal modes in the laser cavity are locked to each other, constructive interference for these modes is yielded, enabling the generation of pulses in the femtosecond range. The first application of mode-locking was around 1963 [9]. In mode-locking, the spectral bandwidth of the pulse, denoted by $\Delta\omega$, is directly influenced by the number of locked modes. This, in turn, determines the duration of the generated pulse,

denoted by $\Delta\tau$. A shorter pulse duration corresponds to a greater bandwidth that consists of a large number of frequency components. Thus, having more phase-locked modes that constructively contribute to the pulse results in a shorter pulse duration, and vice versa. Both, the pulse bandwidth $\Delta\omega_p$ and the pulse duration $\Delta\tau_p$ form the time-bandwidth product ($\Delta\omega_p, \Delta\tau_p \gtrsim \frac{1}{2}$) [10]. The pulse can be understood as a superposition of multiple waves with different frequencies, which constructively interfere. This superposition with different frequency components can be described as ($E(x,t) = \int_{-\infty}^{+\infty} E(\omega)e^{i(\omega t - kx)}d\omega$) [11].

With well-characterised attosecond ultrashort pulses and pump-probe spectroscopy [12, 13], one can achieve a temporal resolution that allows for the real-time observation of the rapid electronic states evolution in atoms and molecules. One of the means to obtain ultrashort pulses in the attoseconds range with XUV photon energies is through high harmonic generation HHG. Ultrashort pulses in the femtoseconds range with high peak intensities $I = Power/Area$, on the order of approximately 10^{14} W/cm^2, are necessary for HHG. Intensities of this magnitude are obtainable through the focusing of ultrashort femtosecond pulses [14]. Unlike the free electron lasers (FEL), HHG is a table-top experiment and is accessible in relatively small laboratories without the need for large facilities. This feature has gained HHG considerable attention in recent years due to its potential applications in attosecond science and ultrafast spectroscopy.

High harmonic generation refers to the phenomenon of converting a driving laser beam into its higher harmonics through the nonlinear interaction between the laser field and a gaseous medium, as in our case. An important aspect of the HHG and its optimization is the parameters of the driving laser field, focusing geometry, and the conditions in the HHG conversion medium. The choice of the wavelength of the driving laser field significantly influences the characteristics of the generated high harmonics. In this thesis, the focus is on experimentally investigating HHG using an 800 nm driving laser source and comparing it with the HHG using a 400 nm driving laser source. The conversion efficiency of the high harmonics (HH) using a 400 nm driving laser field is expected to be higher than that using the 800 nm driving field [15].

The experimental investigation is conducted using the setup at Max Born Institute (MBI), including the laser source, HHG chamber, and the chamber that encapsulates the XUV spectrometer. The choice of these two wavelengths is motivated by their widespread availability. The 800 nm wavelength lies in the near-infrared (NIR) region, while the 400 nm wavelength lies in the ultraviolet (UV) region. Each of these two driving wavelengths yields a different high harmonic conversion efficiency. The widespread availability of these laser sources enables us to explore the influence of different laser parameters on the HHG process, such as the driving laser pulse energy and the pressure of the conversion medium. The aim is to optimize the generation of high harmonics through investigating the optimum phase matching conditions. This is accomplished by scanning the pressure for optimum harmonic intensity.

This thesis is organized into six sections: the 1^{st} section describes briefly the mathematical description of an ultrashort laser pulse. The 2^{nd} section describes the three-step model of HHG at the microscopic level. The 3^{rd} section describes phase matching as the main focus among the propagation effects. The 4^{th} section describes the setup and its parameters. The 5^{th} section provides an investigation into the phase matching conditions of HHG using a 25 fs, 800 nm driving laser pulse, focusing on two parameters: the pressure of the conversion medium, i.e., the gas, and the driving laser pulse energies. The first experiment in this section was done in an already available setup. The 6^{th} section provides the investigation of the HHG characteristics using an 800 nm and comparing its results with HHG using a 400 nm as driving beams, adding two additional parameters to optimize the driving pulse energy: the group delay dispersion (GDD) and the third-order dispersion (TOD), both applied to the laser pulse. The second experiment with its two parts, the 800 nm and the 400 nm, was done in a new optical layout in order to carry out the mentioned investigation.

1 Gaussian Pulse of the Driving Laser Field

The ultrashort laser pulse is the main means to generate high order harmonics by guiding it through the medium of HHG, a gas medium in our case. The following is the mathematical description of the Gaussian pulse in time domain:

$$E(w,t) = E_0\, e^{-\frac{t^2}{2\sigma^2}} \cdot e^{i(wt+\phi(t))} \tag{1}$$

In this equation, the first exponential term represents the Gaussian envelope of the wave, while the second term describes the oscillation of the electric field. Here, t is the time, σ is the parameter that determines the full width half maximum (FWHM), $\sigma = \frac{FWHM}{2\sqrt{2\ln 2}}$, which in turn determines the duration of the pulse. w_0 is the carrier angular frequency of the oscillating electric field, and $\phi(t)$ is the phase of the E field, which is zero for a transform-limited pulse, and this term describes the chirp for $\phi(t) \neq 0$. The Gaussian wave is a convenient form to describe a wavepacket that satisfies Maxwell's equation. In the following Figure 2, there is an illustration of a realistic Gaussian pulse:

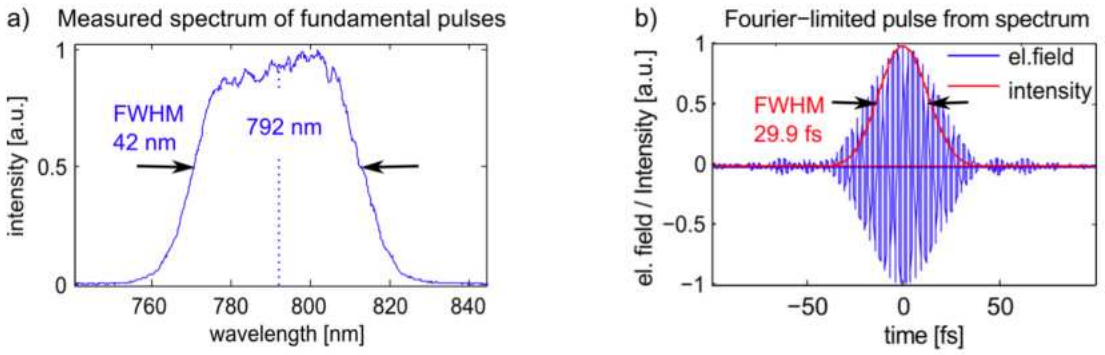


Figure 2: (a) Experimental spectrum of an ultra short laser pulse with FWHM = 42 nm, the carrier wavelength λ = 792 nm, (b) Fourier transform of the pulse duration corresponds to (a). The figures are imported from [16].

Since the laser pulse consists of many frequency components propagating in a medium with a frequency-dependent refractive index $n(\omega)$, the pulse will experience different kinds of dispersion effects like the group delay dispersion (GDD) which is the second-order dispersion, and the third order dispersion (TOD). Due to the broad spectrum of the ultrashort pulse (by definition), the frequency dependence can be neither linear nor quadratic; instead, higher-order dispersion like TOD will affect the pulse. The different kinds of dispersion are quantified by the Taylor series expansion of the wavenumber k as a function of the angular frequency ω.

$$\phi(t) = \phi_0 + \underbrace{\frac{d\phi}{dt}}_{\omega_0}(t-t_0) + \frac{1}{2}\underbrace{\frac{d^2\phi}{dt^2}}_{\text{GDD}}(t-t_0)^2 + \frac{1}{6}\underbrace{\frac{d^3\phi}{dt^3}}_{\text{TOD}}(t-t_0)^3 + \dots \tag{2}$$

ω_o is the carrier angular frequency. The GDD and the TOD are in time domain. Controlling the GDD can cause a linear chirp for the pulses, which results in the spreading out of the spectral components in time. This manipulation either stretches or compresses the pulse in time. Controlling the TOD, meanwhile, can reshape the pulse profile at the trailing and leading edges and induce a quadratic chirp, which gives rise to an asymmetric shape for the pulse.

2 High Harmonic Generation - Three-Step Model

The high harmonic generation (HHG) is a phenomenon in which the frequency of a driving laser beam is transformed into higher-order harmonics, resulting in odd multiples of the frequency of the driving laser field. This is achieved by employing various conversion media and utilizing an intense laser beam with the peak intensity on the order of approximately 10^{14}W/cm^2. This is done by focusing a laser pulse in a medium, resulting in generation of coherent frequencies up to the extreme ultraviolet (XUV) radiation and soft x-ray. The following is a schematic HHG spectrum obtained using the three-step model:

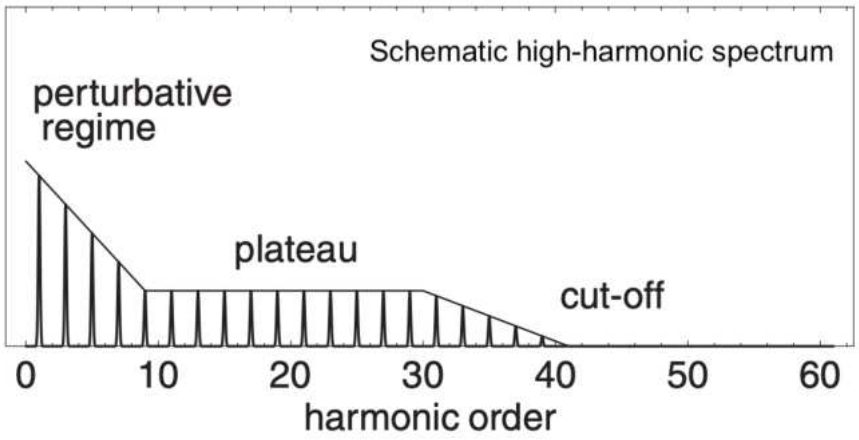


Figure 3: Schematic HHG spectrum for illustration purposes consists of three parts. 1) Perturbative regime: shows low-order harmonics. More details about it can be seen in the subsection 2.1. 2) The plateau: shows the intermediate harmonics. After the rapid decrease of efficiency in the perturbative regime, the intensity of the harmonics remain almost constant in intensity until they reach the 3) cut-off area, which shows the highest orders of harmonics and a rapid decrease in the efficiency. This figure is imported from [3].

Since the discovery of the high harmonic generation and the various efforts to explain it accurately, the quasi-classical three-step model [17] has provided an adequate explanation. The model describes the interaction between the atom and the laser field in three steps: 1) Tunnel Ionization, 2) Propagation, and 3) Recombination. The first step involves the ionization of an atom by a strong laser field. The intense electric field of the laser interacts with the bound electrons, causing them to tunnel out of their bound states and become free. In the second step once the electron is freed, it experiences acceleration in the laser field. The electron flies away from the ion by the electric field of the laser, gains kinetic energy, and then propagates back towards the parent ion. In the final step, the accelerated electron recombines with its parent ion. During recombination, the electron releases its excess energy by emitting a high-energy photon. In this model, the electron undergoes the three steps sequentially.

2.1 Fisrt Step: Ionization

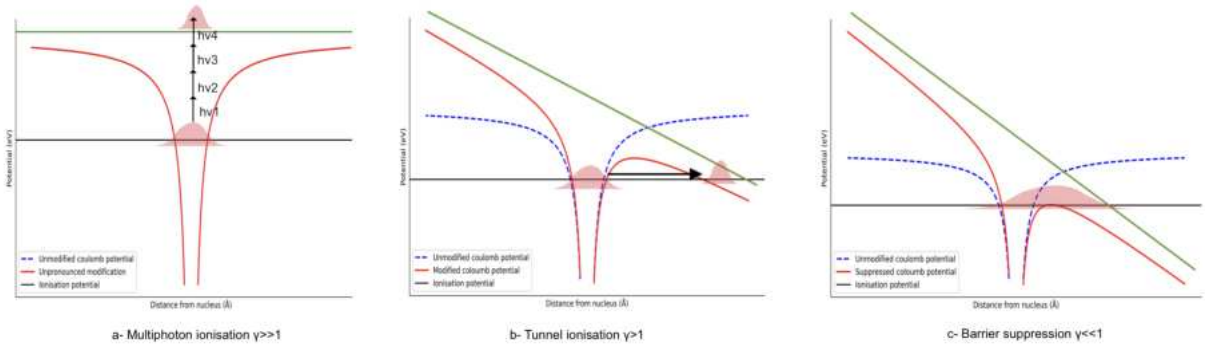
The interaction between the laser pulse and the medium can result in different dominant types of ionization, depending on the field's amplitude. One can determine in which regime one is working—multiphoton ionization, barrier suppression ionization, or tunnel ionization—by the Keldysh parameter [18, 19]. In the multiphoton ionization regime, the parameter is $\gamma >> 1$; in the tunnel ionization regime, it is $\gamma < 1$; and in the suppressed barrier regime, the Keldysh parameter is $\gamma << 1$.

$$\gamma_{Keldysh} = \sqrt{\frac{I_p}{2U_p}} \tag{3}$$

I_p is the ionization potential, which represents the amount of energy needed to remove an electron from an atom or molecule. U_p is the ponderomotive energy. This energy determines the kinetic energy of the propagating electron in the continuum. The ponderomotive energy is given by:

$$U_p = \frac{e^2\, {E_o}^2}{4\, m_e\, \omega^2} = \frac{e^2\, I}{2 m_e \epsilon_0\, c\, \omega^2} \tag{4}$$

e is the electron charge, E_0 is the electric field's amplitude, m_e is the electron mass, ω is the carrier frequency of the electric field, I is the intensity of the field, ϵ_0 is the electric permittivity of the vacuum, c is the speed of light in the vacuum.



a- Multiphoton ionisation γ>>1 b- Tunnel ionisation γ>1 c- Barrier suppression γ<<1

Figure 4: Schematic illustration that shows the types of ionization and the behavior of Coulomb potential accordingly. (a) Multiphoton ionization (MPI) occurs usually at relatively low intensities, where the strength of the laser field is not sufficient to cause a modification to the Coulomb potential barrier. In this type of ionization, a number of photons are absorbed by one atom resulting in the release of one electron. The Keldysh parameter in this case is $\gamma >> 1$. (b) Tunnel ionization: In this case the strength of the field is sufficient to modify the Coulomb potential as shown in the figure, making it probable for an electron to tunnel through the potential to the continuum. In this case, the Keldysh parameter is $\gamma < 1$. (c) Barrier suppression: In this case, the Coulomb potential in significantly modified by a sufficiently strong laser field. This leads to releasing the electron directly to the continuum with out tunneling. In this case, the Keldysh parameter is $\gamma << 1$.

In the case where the Keldysh parameter is $\gamma >> 1$, the dominant type of ionization is the multiphoton ionization regime. It is the regime of small field strength. In such scenario, a minimal number of photons are involved in a single ionization event. This regime is known as the perturbative regime. It describes the number of the weak amplitude photons that are absorbed by a single atom resulting in the emission of a single photon in the low-energy part of the HH spectrum, thereby generating the lower harmonics [20]. However, this is not of the main interest since HHG does not operate in this regime. These photons with weak field amplitude are in the leading and the trailing edge of the pulse, see the leading the trailing edges of a pulse in Figure 6. This scenario corresponds to the first case of ionization, as shown in Figure 4.

If the field strength of the ionizing pulse is sufficiently high, in a way that the potential barrier is bent that it falls below the ground state of the electron, and there is no energy barrier I_p preventing the electron form leaking into the continuum, the ionization will operate in the regime of barrier suppression[21], as shown in Figure 4[c]. According to Equation (3), the Keldysh parameter in this case is $\gamma << 1$.

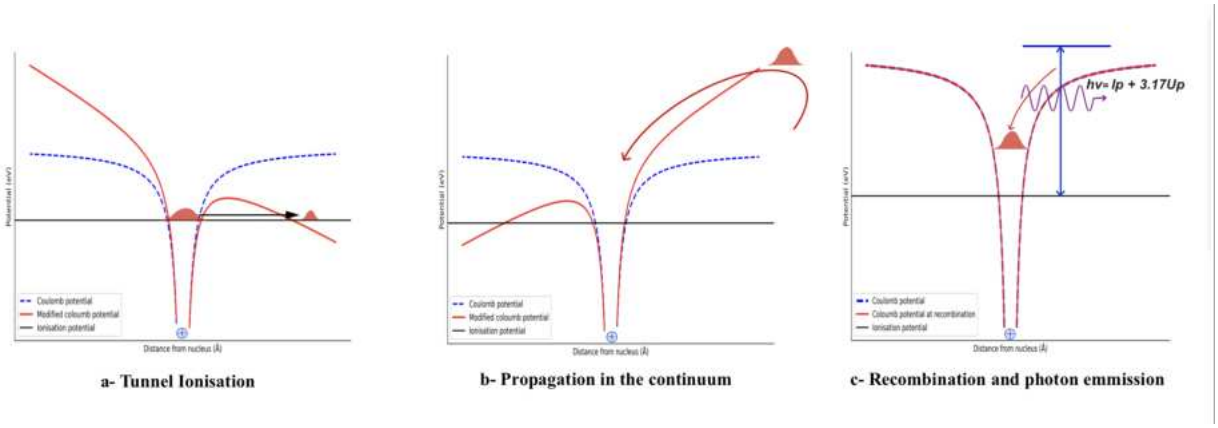


Figure 5: The three-step model of high harmonic generation. The model starts by (a) Well characterised ultrashort laser pulse modifies the atom's Coulomb potential, making it probable for the electron to tunnel through the potential to the continuum. (b) The electron propagates in the continuum, its trajectory is governed by the strong electric field of the laser. The laser field reverses its direction after half a cycle π in which the electron propagates back towards the parent ion with the possibility to recombine with the parent ion. (c) The electron recombines with the parent ion leading to a burst of light. This light is resulted from the release of the ionization potential energy and the excess kinetic energy acquired by the electron during its acceleration in the continuum.

In the regime where the Keldysh parameter $\gamma < 1$, the driving laser field is strong enough to distort the Coulomb potential. This makes it probable for the electron's wave function to leak into the continuum. The electric field in this regime is neither so high that it suppresses the potential barrier—stronger than the field that binds the electron to the atom—nor so low as to operate in the MPI regime. The value of $\gamma \leq 0.5$ from Equation (3) can be used to determine whether the electron is ejected through tunneling or not [22]. In the case of tunneling, the transmission probability can be calculated using Equation (5). Both barrier suppression ionization and tunneling ionization are considered as quasi-static [23]. In the quasi-static approximation, the electron is ionised within one laser cycle. For tunnel ionization through the Coulomb potential with the presence of a strong alternating electric field, as shown in Figure 4[b], an approximation must be introduced; this is known as the WKB approximation [24, 25]. The transmission probability through this potential is:

$$T \approx \exp\left(\frac{-2}{\hbar}\int_{r1}^{r2}\sqrt{2m(V(r)-E)}\right) \tag{5}$$

V(r) is the potential barrier, which in our case is the Coulomb potential modified by the electric field. With a driving laser wavelength of $\lambda \approx 800nm$, the electron can adapt to changes in the Coulomb potential quickly enough to achieve successful tunneling within a half laser cycle. Such tunneling ionization within a half laser cycle is considered quasi-static, where the field of the driving pulse during the tunneling is approximated as a static field. If the frequency of the driving laser field is higher, the motion of the electron would be governed by the average of many cycles of the laser field. Consequently, the quasi-static approximation in a single cycle would no longer be valid. Additionally, in longer wavelengths, the tunnel ionization probability would decrease. The ionization probability is a carefully derived mathematical expression—compared to the WKB approximation—for atoms in strong fields. The expression is modeled by Ammosov, Delone, and Krainov in their well-known ADK paper [26].

The Figure 6 below shows the tunnel ionization probability (w_{ADK}) in atomic units, using the hydrogen atom for simplicity. The ionization probability is plotted as a function of time. The figure emphasizes that the peaks of the ionization probabilities occur around the peak of the driving laser pulse. For this simulation,

an intensity of 2×10^{14} W/cm^2 is used. The ionization probabilities are calculated using a driving pulse with a 25 fs duration and are presented for two different wavelengths: 800 nm and 400 nm, as shown in Figures 6 and 7.

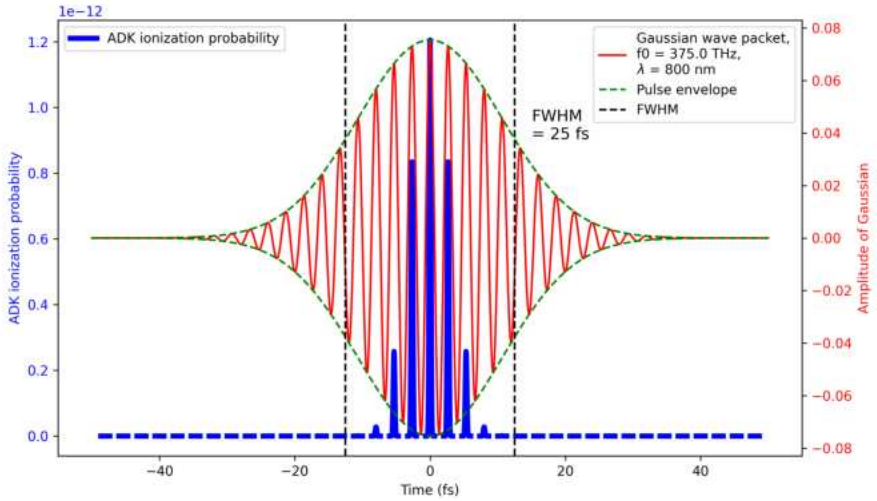


Figure 6: The blue line in the figure represents the simulation of the tunnel ionization probability for the hydrogen atom in atomic units, as described in the ADK paper [26]. The red line is the Gaussian pulse with an intensity of 2×10^{14} W/cm^2 used to generate the ionization probability. See Appendix C for further details regarding the pulse and the probability simulation.

Figure 7 shows that the extent of ionization probability on the horizontal axis is constant within the FWHM range when compared with Figure 6. This is due to the similar intensity value used in both simulations at 2×10^{14} W/cm^2. Only an increase in the field's amplitude leads to an increase in the ionization probability area along the x-axis, as shown in Figure 7. As mentioned in the subsection 2.1, the field strength can determine the dominant type of ionization. The frequency determines the number of the laser cycles during which ionization occurs, as demonstrated in the latter figure. Therefore, the density of the probability lines is increased within the same area.

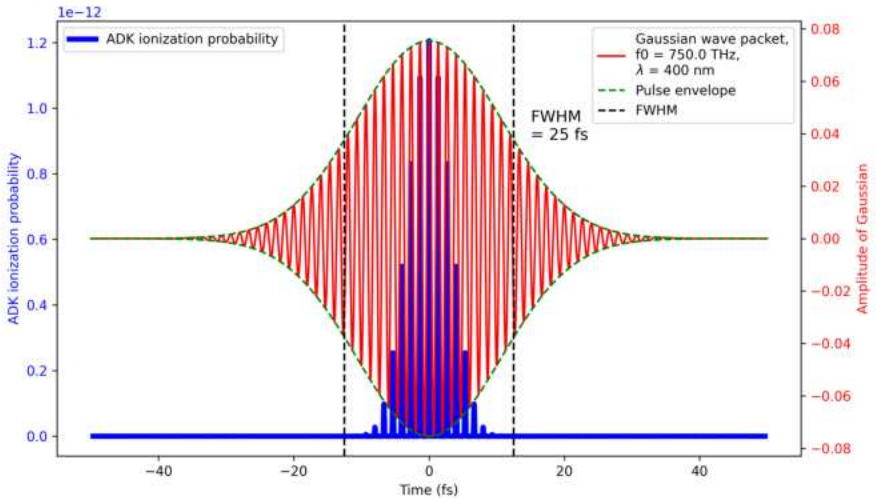


Figure 7: The same simulation of 6 with the same parameters but for 400 nm carrier wavelength. The oscillatory term used for the pulse is $cos(\omega t + \phi)$. The simulation is done in atomic units.

2.2 Second Step: Electron Propagation and Long - Short Trajectories

Now, after ionization, the electron is no longer bound to the atom; it is free in the continuum. Its motion is governed by the driving periodic electric field $E(t) = E_0, \cos(\omega t + \phi)$. The second step investigates the electron's propagation in the electric field and explores the different electron trajectories that can result from various birth times and different phases of the driving laser pulse. Consequently, this step investigates the different kinetic energies acquired by the electron during its propagation through these various trajectories. The semiclassical three-step model suggests that the motion of the electron starts at zero velocity, $\nu_0 = 0$. The second step starts with the velocity of the electron [4]:

$$\nu(t) = \nu_o + \int_0^t a(t)\, dt = \nu_o + \int_0^t -\frac{e}{m_e} E(t)\, dt = -\frac{E_o\, e}{m_e\, \omega}(sin\,(\omega\, t + \phi) - sin\,\phi) \tag{6}$$

ν_o is omitted from the right-hand side of Equation (6), because $\nu_{t=0} = 0$. m_e is the mass of the electron, and $a(t)$ is the acceleration of the electron. The trajectory of the electron is determined by:

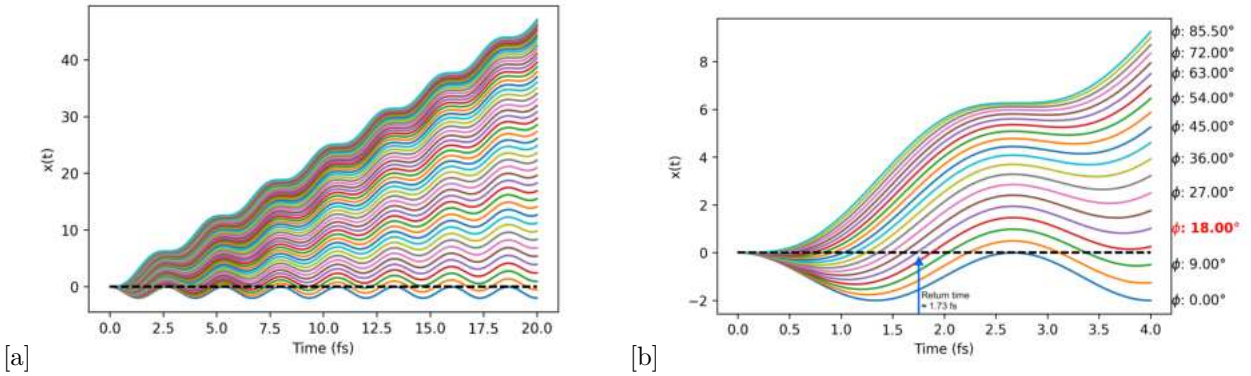
$$x(t) = x_o + \int_0^t \nu(t)\, dt \tag{7}$$

$$x(t) = x_o + \frac{e\, E_o}{m_e\, w^2}\,(cos(w\, t + \phi) - cos\,\phi) + \frac{e\, E_o\, t}{m_e\, w}\, sin\,\phi \tag{8}$$

The latter equation is solved numerically for $x(t) = 0$, i.e., the recombination position. This model shows good agreement with the quantum mechanical approach [27]. The aim here is to calculate the highest kinetic energy, $\frac{1}{2} m \nu^2$; we need to calculate the highest velocity that the electron may have at $x(t) = 0$. This velocity can be obtained through the optimal trajectory that corresponds to the optimum phase $\phi \approx 18°$ of the driving field at $x(t) = 0$ [4, 22]. The different phases of the driving field correspond to different moments of possible electron birth. The three-step model predicts that the electron can acquire kinetic energy at the moment of

collision equal to $3.17, U_p$, which agrees with the experimental results [28]. U_P, as aforementioned, is defined by the electric field.

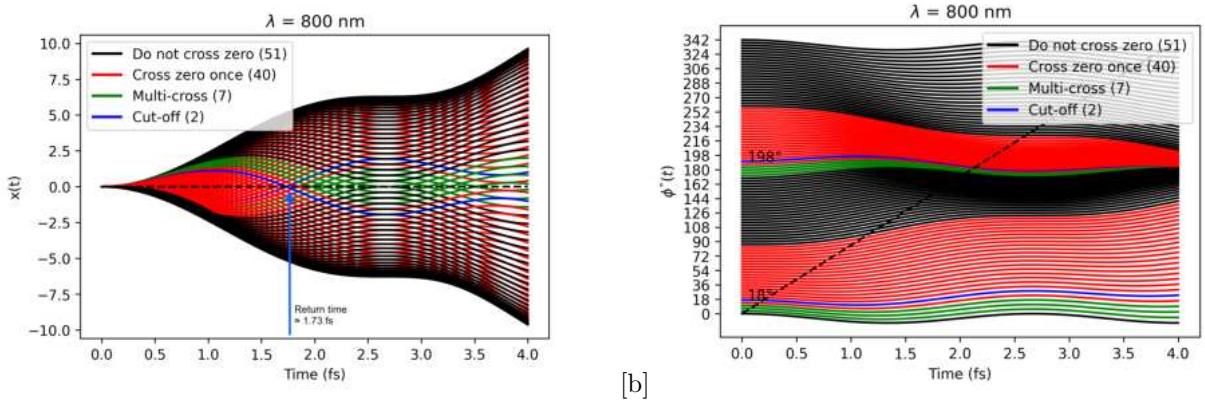
For E-field phases larger and smaller than the optimum phase, there emerge the trajectories that are called *long* trajectories (for they take a relatively long route in the continuum), and *short* trajectories (for they take a relatively short route in the continuum). Through both of these trajectories, the electron acquires different kinetic energies smaller than $3.17, U_p$. In principle, any kinetic energy acquired by the electron from a long trajectory can be acquired from a short trajectory as well. More details on this point can be found in Figures 12, 9, and 16. The emission resulting from these different long and short trajectories is observed in the plateau region of the HH spectrum which decreases at the cutoff, as shown in Figure 3.



[a] [b]

Figure 8: (a) A simulation of 50 different trajectories of the electrons as a function of time. These trajectories correspond to various phases of a quarter cycle $0 < \phi < \frac{\pi}{2}$. (b) is the same plot but on a smaller time scale in order to take a closer look at the trajectories around the recombination position $x(t) = 0$.

Figure 8[a] traces only 20 lines of trajectories in order to reduce the density of the colored pixels. This figure highlights the trajectory that corresponds to the cutoff phase. The electron in this trajectory spends approximately 1.73 fs until it returns to the parent ion. This figure also shows that for the trajectories corresponding to $\phi, < 18°$, the electron spends a longer time in the continuum until it comes back to the parent ion. These trajectories are referred to as the long trajectories. On the other hand, the trajectories corresponding to $\phi, >, 18°$ are the short trajectories, in which the electron spends a shorter time until it comes back to the parent ion.

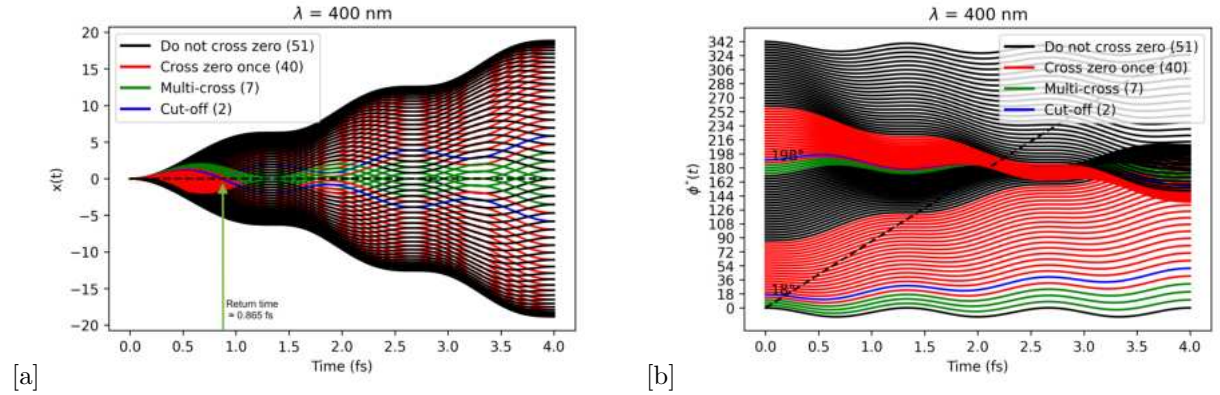


[a] [b]

Figure 9: (a) Simulation of 100 different electron trajectories as a function of time, generated using an 800 nm wavelength. (b) This figure presents the same 100 trajectories as in Figure (a), with the various phases distributed equidistantly along the vertical axis. The simulation is done in SI units. For details of the simulation, see Appendix C.

These trajectories of Figure 9[a] correspond to a span of phases of a full laser cycle, $0 < \phi < 2\pi$. The trajectories start at the moment of ionization ($x_{t=0} = 0$). The vertical axis of the latter figure refers to the distance from the parent ion, while the horizontal axis refers to the time spent by the electron after its birth until it comes back to the parent ion, $x_t = 0$. Through Equation (8), it is possible to highlight the trajectory that corresponds to the cutoff phase in each half cycle. It is indicated by a blue line in the latter figure. This information is shown in Figure 11. The electrons in the cutoff trajectory spend time of approximately 1.73 fs until they return back to the parent ion ($x_{t=1.73fs} = 0$). The electrons in these two trajectories pass the zero point $x_t = 0$ only once, as Figure 9[a] shows.

The trajectories shown in Figure 9[b] are unfolded with an offset in order to observe clearly the different windows of phases that yield different kinds of trajectories. The figure shows three kinds of trajectories that correspond to three kinds of phase windows: those that do not cross zero, those that cross zero once, and those that have multi-crosses. These three windows repeat themselves in each half cycle π. The latter figure shows at which phases the cut-off trajectory lie with respect to the mentioned kinds of phases. The long trajectories have relatively smaller phase window than the short trajectories. This information is shown as well in Figure 12. However the trajectories with multi-crosses are only among the long trajectories. Figure 9[b] shows as well that more than half of the trajectories do not return the electron to the parent ion. This significantly contributes in the formation of the plasma. The percentage of the kinds of the trajectories mentioned in the latter figure remains roughly the same around the investigated wavelength 800 nm, unless the wavelength is altered by roughly two orders of magnitude higher or lower. If the wavelength of the driving field is relatively long compared to 800 nm, the electron would have a longer route of propagation due to the spread of the pulse envelope. This makes the electron's return to the parent ion less likely. The simulation of other wavelength is beyond the scope of this thesis. The ratios of the phase types at this wavelength are 51% of the trajectories do not lead the electron back to the parent ion, 42% of the trajectories cross the zero line once (including the cut-off trajectories), and only 7% show the possibility of multi-crossing). In general, the curves of the latter figure show that less than half of the trajectories lie in a phase range of the laser cycle that can be considered as an effective window of the phase, i.e., yields a trajectory that guides the electron back to the parent ion.



[a] [b]

Figure 10: (a) The same simulation of 100 different trajectories of electrons as a function of time but for the wavelength 400 nm wavelength. The simulation was conducted in SI units. For more details, see Appendix C.

Figure 10[a] shows that the behavior of the trajectories is similar to that in the recent simulation for the 800 nm. Figure 10 shows that the ratio of the types of phases is similar to those of the trajectories at the wavelength 800 nm, as shown in Figure 9. For the wavelength 400 nm, the trajectories at this wavelength require a shorter time ≈ 1.3 fs, to complete a full laser cycle, which is half of the latter period for the 800 nm. Additionally, the time of the electron return at the cut-off trajectory is ($x_{t=0.865fs} = 0$). The shorter trajectories at the 400 nm wavelength is a factor to the electrons gaining less kinetic energy compared to the

800 nm. Figure 10[a] shows that within the frame work of a single full laser cycle 2π, the electrons wobbles in their trajectories in an oscillatory motion in the transverse plane of the plot while being driven away—by the trajectories of the electric field—of the parent ion along the vertical axis.

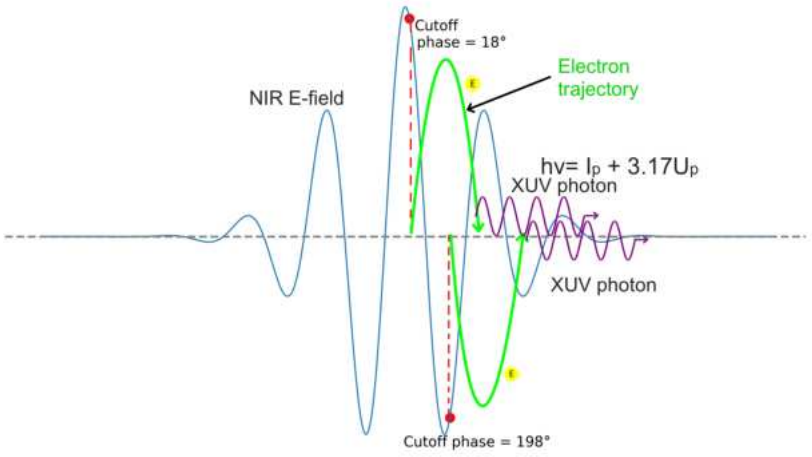


Figure 11: Illustration of a few-cycle laser pulse and its cut-off phase.

2.2.1 Cut-off Law

Due to energy conservation, all the kinetic energy acquired by the electron during its propagation will be converted to photon energy, $\hbar\omega$, once the electron recombines with its parent ion. One objective of this step is to determine the phase of the trajectory in which the electron acquires the highest energy at the moment of recombination with the parent ion. The ionization energy, I_p, that was needed to eject the electron from its orbit, will also contribute to $\hbar\omega$:

$$\hbar\omega = E_{kin} + I_p \tag{9}$$

The maximum kinetic energy is $3.17\,U_p$, where U_p is the ponderomotive energy. It depends on the amplitude of the electric field E_o and the frequency ω, see Equation (4). The electrons that are born at the optimal phase, as described in the subsection 2.2, yield at the recombination the highest energy of harmonic photons. Since there are two peaks of the electric field in each cycle, and the optimum phase is at approximately 18°, the atom will encounter two peaks of the electric field in each complete cycle 2π and the process described by three-step model repeats itself every half cycle, as in Figures 12, and 11. In Figure 8[b], is shown the trajectory that corresponds to this optimal phase in red font.

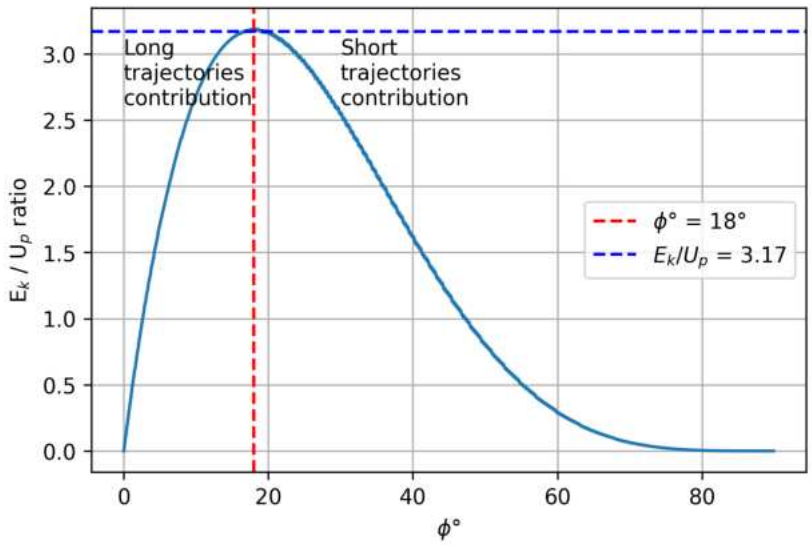


Figure 12: The simulation illustrates the cut-off relation. On the vertical axis are the ratios of E_k/U_p. The horizontal axis denotes the various phases of a half laser cycle π in degrees. The black line indicates that See the calculation in Appendix C.

The highest prefactor is 3.17, corresponds to the kinetic energy (E_k) of an electron propagating along the optimal trajectory at the moment of recombination as shown in Figure 12. This results in a photon with energy $\hbar\omega = 3.17U_p + I_p$. The value of E_k reaches the cut-off value $3.17\,U_p$ at the moment of the electron return to the parent ion. The electron takes $\approx$ 1.73 fs from the moment of birth to return to the parent ion in the optimal trajectory, as indicated in the 8. The phase associated with this optimal trajectory is $\phi \approx 18°$. Figure 12 also demonstrates that any ponderomotive energy with a prefactor lower than 3.17 can be produced from both a long and a short trajectory.

When an electron tunnels into the continuum at this particular phase of the electric field, it propagates in a trajectory that allows it to gain maximum kinetic energy due to acceleration, as shown in Figure 11, this trajectory then guides the electron back to its parent ion.

2.3 Third Step: Recombination

The intensity spectrum of HHG can be determined by utilizing the re-collision dipole moment, which can be represented in either the quantum or classical picture.

The re-collision dipole is represented in the classical picture due to the correspondence principle (i.e. the systems described in quantum theory with large quantum numbers reproduce classical physics), which will give the spectrum of HH. The description of this part is in the frame of semiclassical three-step model that is based on P.B.Corkum's work [29] that works together with the ADK paper [26] utilising the quasi-static approximation.

The emission of the HH is determined by calculating the expectation value of the dipole operator:

$$\langle \Psi | e\vec{z} | \Psi \rangle = \langle \Psi_g | e\,\vec{z} | \Psi_c \rangle + \langle \Psi_c | e\,\vec{z} | \Psi_c \rangle + c.c. \tag{10}$$

Where Ψ_g is the wave function of the ground state, and Ψ_c is the continuum state wave function, e is the electron charge, $\vec{z}$ is the dipole vector in the z-axis. The terms $\langle \Psi_g | e\,\vec{z} | \Psi_c \rangle + c.c.$ are the dominant ones which manifest the harmonic spectrum that is shown in Figure 13. Given that we are dealing with large orbits, the continuum wave function Ψ_c, which has a large extent compared to any ground state, can be approximated as a plane wave. A detailed explanation can be found in the references [30–32]. The plane wave approximation [29] of the continuum state is shown in the following Equation:

$$\Psi_c(z \approx 0, t) = e^{i(k\vec{z} - \omega t)} \tag{11}$$

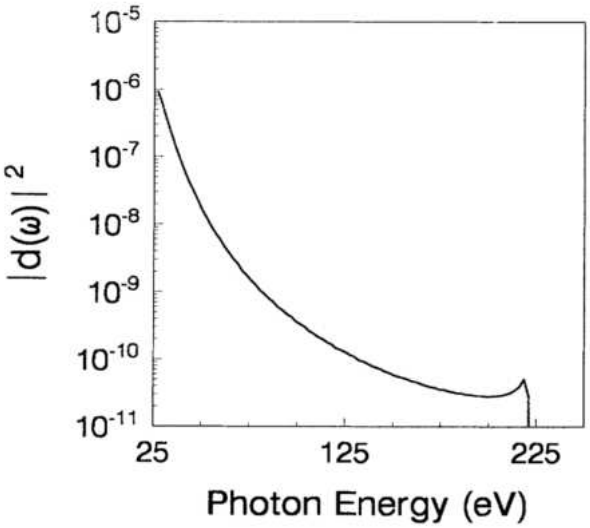


Figure 13: The square dipole moment of the ion-electron re-collision calculated in the Fourier domain. The driving laser used has a wavelength of 1.06 μm and an intensity of $6{\times}10^{14}$ W/cm^2.The helium gas is the conversion medium. This figure is imported from [29].

The second step of the three-step model shows that the extent of the photon energy produced by an 800 nm driving field is greater than the extent of photon energy produced by the 400 nm driving field. Furthermore, Figure 13 shows that with greater wavelength, such as the 1.06 μ, m theta is used in Figure 13, higher photon energy values can be produced, up to 225 eV. The following spectrum of the HH is calculated through the Fourier transform of the time-dependent dipole moment.

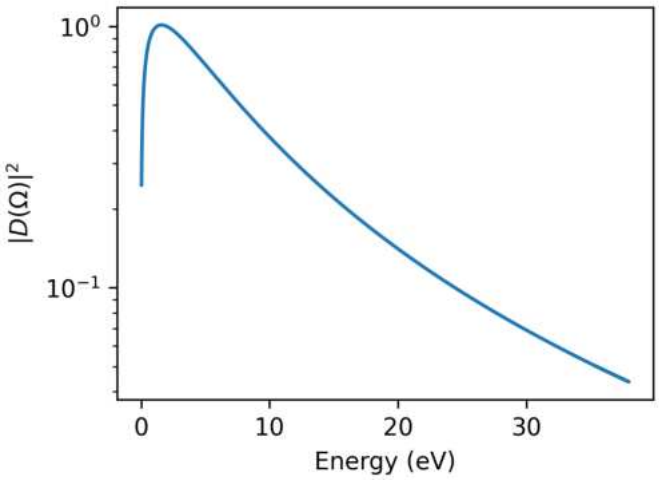


Figure 14: Simulation of the time dependent square dipole moment of the ion-electron recombination in the frequency domain resulting from the recombination. At the vertical axis, the figure shows the strength of the dipole moment calculated for a driving laser field with a carrier wavelength 800 nm and intensity 2×10^{14} W/cm$^-$2 for the ionization potential of argon atom 15.76 eV. The horizontal axis shows the resulted photon energies of the dipole moment. The simulation is done in atomic units. For more details see Appendix C. This figure and its code are adapted from [32], with the authors permission.

3 Propagation Effects in HHG

After viewing the HHG scenario of a single atom in a microscopic scale through the three-step model, which investigates the tunnel ionization regime, propagation of the electron in the continuum, and recombination, the three-step model predicts the shape of high harmonic spectrum fairly well. A crucial part, known as the phase matching that needs to be established carefully in order to observe the HH emission in the macroscopic scale. The phase matching is achieved by matching the phase of the driving laser pulse to generate HH and the phase of co-propagating generated high harmonics. Increasing the phase matching conditions over a longer distance of the conversion medium yields a quadratic increase in the intensity of the HH emission. If the phase matching condition is not fulfilled, the HH will propagate in different phase velocities along the propagation axis and the intensity of the HH will increase and decrease periodically. This means that the phases of the HH are interfering both constructively and destructively in a periodic manner [22]. In this case of intensity periodicity, the distance between two points where the HH are completely out of phase is called the coherence length L_c. The intensity behavior of the HH in the two cases of matched and mismatched conditions is shown in the following figure:

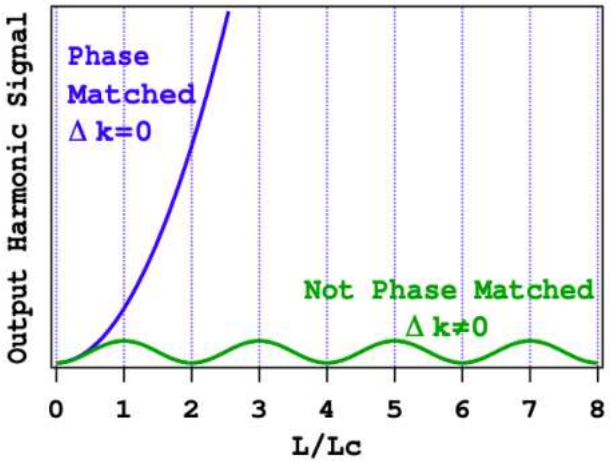


Figure 15: The intensity of the HHG spectrum as a function of the conversion medium length L and the coherence length L_c in the case of phase matching and its absence. This figure is imported from [22].

The propagation of HH will encounter four kinds of dispersion, each of them leads to the modification of the wavevector. Since a gas is used as a conversion medium, a neutral dispersion will modify the wavevector by the value $\Delta k_{neutral}$. Due to the plasma created by the intense driving laser pulse, the phase will be modified due to the dispersion of this plasma by the value Δk_{plasma}. The geometrical phase of a plane wave is always zero [33], however due to the focus of the driving laser beam, the beam will acquire a geometrical phase that also contributes to the modification of the wavevector by the amount $\Delta k_{geometric}$. Additionally, due to the difference between the generated photon phase at the moment of recombination and the phase of the driving laser field in a single atom scenario, an atomic phase mismatch with the value $\Delta k_{tot\,atomic\,DP}$ must be considered. The vacuum phase mismatch is zero, $\Delta k_{vacuum} = 0$, therefore it is neglected. The total wavevector mismatch equation then becomes:

$$\Delta K_{tot} = \Delta k_{neutral} + \Delta k_{plasma} + \Delta k_{geom} + \Delta k_{tot\,atomic\,DP} \tag{12}$$

Figure 16 shows that photons with different wavevectors are born from different electron trajectories. Once

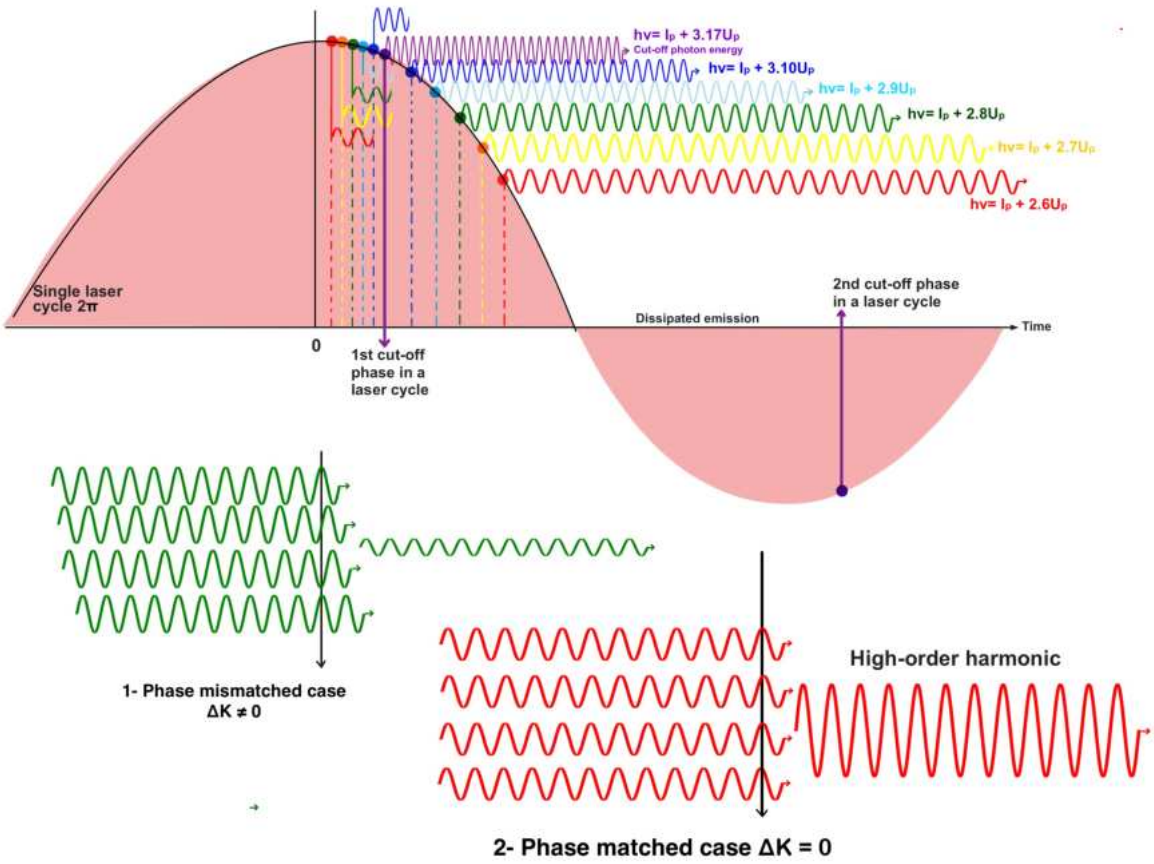


Figure 16: Illustration of the mismatched wavevector (1^{st} case, $\Delta K_{tot} \neq 0$) and the matched wavevector (2^{nd} case, $\Delta K_{tot} \approx 0$ of the emitted photons due to the electron recombination after propagating in the continuum and coming back to the parent ion at the HHG process. The factors of the ponderomotive energies U_p are not real and are written approximately for illustration purposes.

it is managed to phase match the wavevector of these photons, it interferes constructively and yields a coherent emission. The latter figure shows as well that there are two cut-off phases in each complete cycle. This indicates that an atom can undergo the three-step model twice in each laser cycle. The prefactor of the U_p is at its maximum at the cut-off phase and decreases once the phase shifts to before and post the cut-off phase yielding the long and short trajectories as shown in Figure 16.

3.1 Neutral Dispersion

Due to the frequency-dependent refractive index ($n(w)$) of the HHG conversion medium, the equation of the neutral dispersion wavevector mismatch will be:

$$\Delta k_{neutral}(w) = [n(w_0) - n(m_{th}\, w_0)]\, \frac{m_{th}\, w_0}{c} \tag{13}$$

w_0 is the frequency of the driving laser, and m_{th} is the high harmonic order. The neutral dispersion mismatch depends on the frequency of the driving laser w_0 and the frequency of the generated high harmonics ($m_{th}\, w_0$). Due to the fact that the refractive index value $n(w_0)$ is greater than 1 in the case of visible and near IR laser pulse and the $n(m_{th}\, w_0)$ is smaller than 1 at the XUV wavelength of the generated HH [34], the sign for this

wavevector mismatch contribution is positive $\Delta k_{neutral} > 0$. The calculation of the refractive index values ($n(w)$) of a medium at different wavelengths is done through the Clausius-Massotti relation. The details of the relation can be found in the following reference: [35].

3.2 Plasma Dispersion

As illustrated in Figures 9 and 10, for the cases of 800 nm and 400 nm as carrier wavelengths of the driving laser field, the majority of trajectories do not guide the electrons back to their parent ions at $x(t) = 0$. This creates a plasma in the conversion medium with a refractive index $n_{plasma}(w)$ given by:

$$n_{plasma}(w) = \sqrt{1 - \left(\frac{w_p}{w}\right)^2} = \sqrt{1 - \left(\frac{N_e}{N_c(w)}\right)} \tag{14}$$

The w_p is the frequency of the plasma and is given by $w_p = \sqrt{\frac{q^2\, N_e}{\varepsilon_o\, m_e}}$, N_e is the free electron density that forms the plasma, q is the electron charge, ϵ_o is the electric permittivity of vacuum and m_e is the electron mass. N_c is the critical density of the plasma and is given by $N_c = \frac{\varepsilon_o\, m_e\, w^2}{e^2}$. The created refractive index created by the plasma will contribute to modifying the wavevector by the value $\Delta k_{plasma}(w)$, which is given by:

$$\Delta k_{plasma}(w) = (n_{plasma}(w) - 1)\frac{m_{th}\, w}{c} = \frac{-w_p^2}{2c\, w} \tag{15}$$

The refractive index of the plasma is $n_{plasma}(w) < 1$, so the wave vector mismatch of the plasma $\Delta k_{plasma}(w)$ will be negative.

3.3 Geometrical Phase (Gouy Phase)

The Gouy phase shift appears over the waist of the focused laser beam. In order to obtain the Gouy phase mismatch, we start from the local wavevector $k(z)$ at a given point in the space, which can be expressed as $k(z) = \nabla\Phi(z)$, where ∇ represents the gradient operator and $\Phi(z)$ is the phase of the wave at position z. From the wave equation in vacuum for a focused beam, we obtain an additional spatial phase that is given by:

$$\Phi_{foc}(z) = arctan\left(\frac{z}{z_0}\right) \tag{16}$$

z is the position along the propagation axis, and z_0 is the Rayliegh length, which is the point where the beam waist transverse area doubles in size by definition. The Gouy phase shift is manifested by a retardation of the phase of the wavefront that occurs during the passage through the focus area [10, 36].

$$k_{foc}(z) = \frac{d\Phi_{foc}(z)}{dz} = \frac{1}{z_0} \tag{17}$$

The wavevector mismatch of the Gouy phase will be:

$$\Delta k_{foc} = \frac{(m-1)}{z_0} \tag{18}$$

Thereupon, we know that the Gouy phase mismatch contribution is positive $\Delta k_{foc} > 0$.

3.4 The Intensity Dependent Dipole (Atomic) Phase

The emitted photon phase depends on the return time of the electron to the parent ion. At the moment of electron recombination with the parent ion, the phase of the emitted photon is different from the phase of the driving laser wave. The electron trajectory varies with different intensities of the driving laser field. This indicates that the time of the emitted photon's birth will differ, therefore, its phase differs. This causes a mismatch between the driving field's phase and the phase of the HH. This mismatch is low for small intensities. It is called the atomic dipole phase mismatch because it is not related to the propagation, but rather it is related to what happens in a single atom. The total atomic phase of the HH is expressed in the following equation:

$$\Phi_{tot\,atomic\,DP} = m_{th}\, w\, t_r - \frac{1}{\hbar}\, S(t_b,\, t_r) \tag{19}$$

Where m_{th} is the order of the harmonic, w is the angular frequency, t_b and t_r are the time of birth (release) and the time of recombination respectively, $S(t_b,\, t_r)$ is the quasi-classical action [37]. The relation 19, establishes a relationship between the intensity of the driving field and the total phase of the HH field. This relation consists of two components: the phase of the driving laser field at the time of recombination t_r, and the atomic phase, i.e. the dipole phase (also referred as the intrinsic phase) $S(t_b,\, t_r)$. The dipole phase is a result of the accumulated phase from the propagated electron wave function in the continuum. The quasi-classical action (the dipole phase) of Equation (19) is given by:

$$S(t,\, t_r) = \int_{t_b}^{t_r} (\frac{p(t_b,\, t_r)^2}{2\, m_e} + I_p) dt \tag{20}$$

$p(t_b,\, t_r)$ is the classical momentum of the electron at the t_r time of recombination. m_e is the mass of the electron, I_p is the ionization potential of the atom. In the relation of the ponderomotive energy U_p 4, one can see the relationship between U_p and the intensity I, where $U_p \propto I$. In the three-step model, as shown in Figure 16, different trajectories yield different kinetic and ponderomotive energies $E_{kin} \propto U_p$, and consequently different m_{th} harmonic. The electron wave function propagating in the $x(t)$ trajectory will be accumulating its own phase. Therefore, we find a dependence between this phase and the intensity of the driving beam. By combining the trajectory Equation (8), with the total atomic phase Equation (19), the atomic dipole phase equation is obtained:

$$\Phi_{tot\,atomic\,DP} = \frac{2\, U_p}{\hbar w} \int_{\phi_b}^{\phi_r} d\phi\, (sin(\phi) - sin(\phi_b))^2 + \frac{I_p}{\hbar w}(\phi_r - \phi_b) \tag{21}$$

$\phi_b = w t_b$ is the phase at release (the birth time of the electron), $\phi_r = w t_r$ is the phase at the time of recombination. In Figure 18 the evolution of the dipole phase of the short trajectories and the long trajectories with respect to different intensities for the 31^{st} harmonic is shown. The dipole phase of the short trajectories from which the 31^{st} harmonic is generated show almost no change with varying intensities, while

the long trajectories from which the 31^{st} harmonic is generated show constant ratio with respect to different intensities [38, 39]. The change of the total harmonic phase with respect to the intensity can be shown by:

$$\frac{\partial \Phi_{tot\,atomic\,DP}}{\partial I} = \frac{2\,K}{\hbar w}\,\frac{\partial U_p}{\partial I} = \frac{K\,e^2}{\hbar\,m\,\epsilon_0\,c\,w^3} \tag{22}$$

K is a dimensionless constant that depends on the trajectory of the electron. It is the solution for the integration in Equation (21). Equation 22 is solved numerically in the following Figure 17.

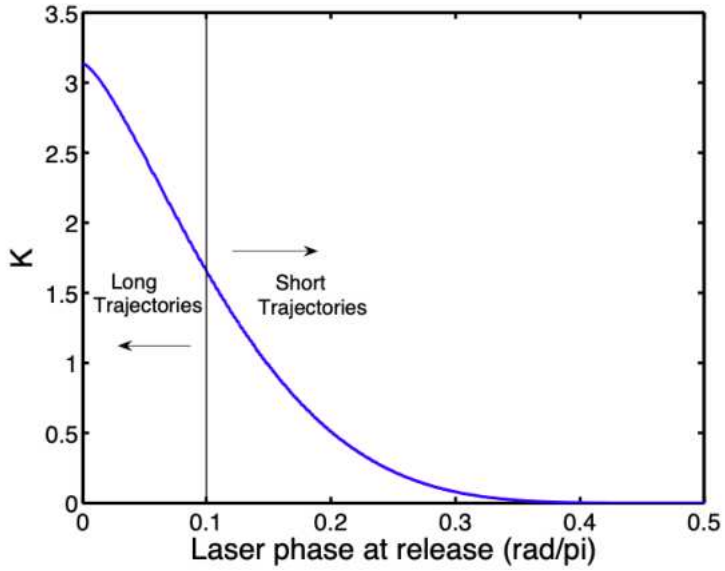


Figure 17: K values as a function of the phase birth ϕ_b, where the black line corresponds to $\phi = 0.1\pi\,rad \approx 18°$, i.e., the cut-off phase. The values of the release phase $\phi_b = wt_b$ (at the moment of birth) that are smaller than the cut-off phase correspond to the long trajectory phases, whereas the values greater than $0.1\pi\,rad$ correspond to the short trajectory phases. This figure is imported from [22].

The K values that correspond to the long trajectories are higher than the K values of the short trajectories. It is possible to select which trajectories (long or short) contribute more to the HHG emission. This is achieved through placing the gas jet after the focus point of the driving beam for greater short trajectories contributions, whereas more contributions from the long trajectories are obtained by placing the gas jet before the focus point of the driving beam [40]. This makes the atomic phase compensated by the negative Gouy phase.

Experimentally controlling the pressure of the HHG conversion medium in the HHG cell, as will follow, impacts the plasma dispersion mismatch Δk_{plasma} and the neutral dispersion mismatch $\Delta k_{neutral}$ due to their dependence on the density of the medium. After determining the signs of the different contributions that modify the wavevector mismatch in Equation (12), one can control the plasma dispersion contribution by regulating the gas pressure. Subsequently, we control the density-(*number of atoms/volume*)-to increase or decrease the negativity of the plasma mismatch contribution in order to level it up with the other terms of the equation, aiming to have $\Delta K_{tot} = 0$.

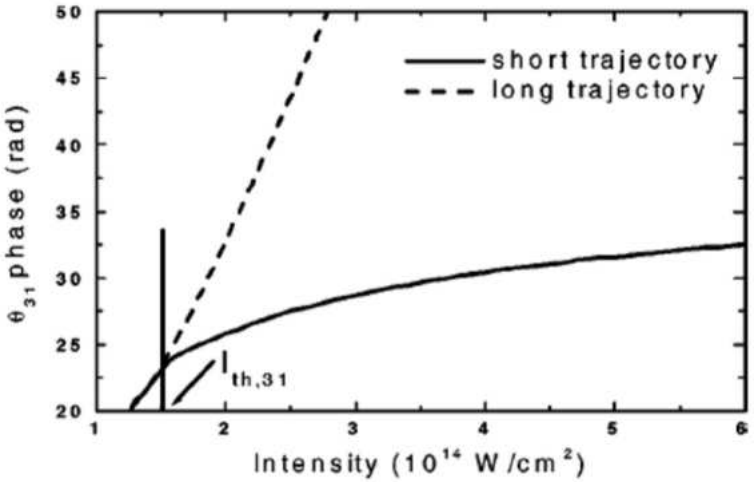


Figure 18: The figure show the sensitivity of the long trajectory dipole phase of the 31^{st} harmonic of argon gas to the laser intensity, while the short trajectory of the same harmonic is almost insensitive to the laser intensity. The slopes of the two trajectories are determined by the dimensionless K values. This figure is imported from [18].

4 Experimental Setup

This section describes the experimental setup that was used to investigate the phase matching conditions. The section includes: an overview of the different parts of the experimental beamline, a description of the used vacuum system and its characteristics, a brief description of the Beta-Barium Borate Crystal (BBO) that was used to double the frequency of the driving laser pulse, a setup description of the 400 nm experiment, a description of the HHG source (the HHG chamber), a description of the gas inlet and the HHG cell through which the NIR beam passes to generate HH, a description of the XUV spectrometer in which the HH are observed as vertical lines by the CCD camera.

4.1 Overview of the Experimental Beamline

In the two experiments, a state-of-the-art chirped pulse amplification commercial laser system (Aurora, Amplitude Technologies) was used. The laser provides ultrashort pulses in the NIR range with a 795 nm carrier wavelength. The laser system has a 1 KHz repetition rate with an output energy up to 2.16 mJ that was used for the experiment. The pulse duration produced by the laser is 25 fs.

The arm of the laser directs a short NIR pulse to the HHG chamber. After the HHG chamber, the generated XUV radiation co-propagates with the NIR towards the XUV spectrometer. All the co-propagating driving laser pulses are filtered out later, before the entrance of the XUV spectrometer, with an XUV-transmissive aluminum filter that allows only 20-70 eV to pass through.

Figure 19, shows the arrangement of the different parts of the setup. The setup consists of the aforementioned laser system, a second harmonic generation setup chamber, an HHG chamber that has a gas pressure-regulated HHG cell, and the XUV spectrometer.

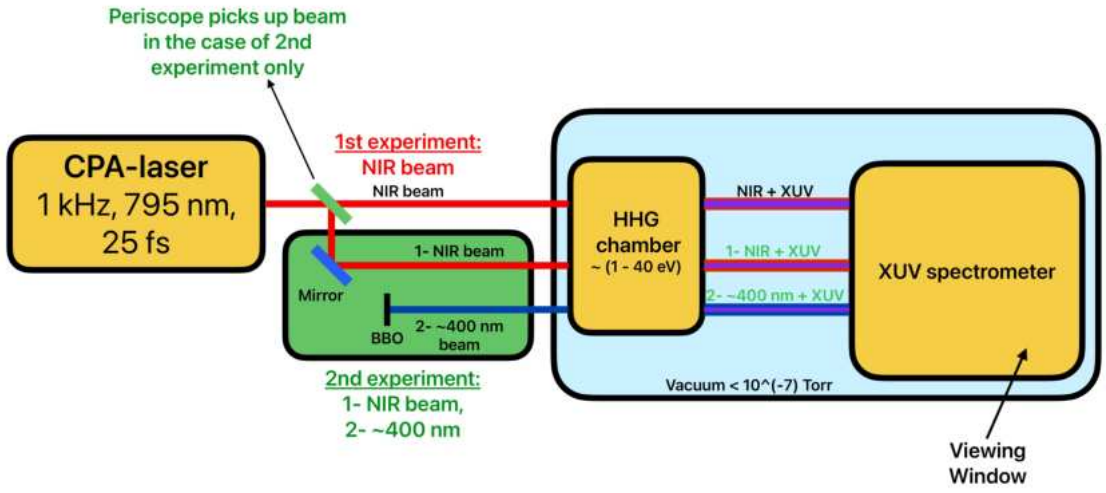


Figure 19: A setup overview for the entire beamline with the laser system is presented. The first experiment (written with red font) was conducted using only the NIR beam for an already available setup. The second experiment (written with green font) is referred in the green. This experiment investigates the phase matching using an NIR beam, and it investigates the phase matching using a 400 nm beam with two additional parameters.

4.2 Vacuum System

The beamline, starting from right before the HHG chamber, is contained in the vacuum system. This would enhance the stability of the experimental room through the protection from external environmental conditions. The high photon energies in the XUV range are highly absorbed by different materials. Therefore the layout of the XUV radiation is encapsulated in this vacuum system to prevent absorption of the XUV radiation. The laser system, the source of the NIR beam, is not contained in the vacuum system, as indicated in Figure 19.

The beamline generally consists of two major parts, the HHG chamber and the chamber housing the XUV spectrometer, as shown in Figure 19. In order to reduce the gas leakage from the HHG chamber into the rest of the beam line, a deferential pumping is introduced between the HHG chamber and the chamber housing the XUV spectrometer. Note that the gas pressure in the HHG cell only is varied later for the phase matching purposes. The low pressure, which is approximately 10^{-7} Torr, in the vacuum chamber is intended to remain fixed to stabilize the beam from external factors. However, the pressure in the vacuum chamber changes once the gas of the conversion medium flows into the HHG cell. Separate vacuum gauges are used to monitor the pressure in the different chambers within the vacuum system.

4.3 Setup of 400 nm Experiment

In the first measurement setup of the second experiment, a new optical layout was configured. A periscope was used to alter the altitude of the beam, adjusting it to match the altitude of the HHG chamber entrance. A half-wave plate is introduced after the periscope to restore the original horizontal polarisation state of the beam. This layout was used to generate HH using the NIR beam. This layout allows for a comparison between the HH results generated using 800 nm and 400 nm beams.

Next, In order to obtain a 400 nm wavelength beam, a BBO crystal is used to convert the 800 nm beam into 400 nm beam through frequency doubling. Two mirrors were introduced that are reflective for the 400 nm wavelength range. These mirrors have a wavelength reflection range of (380-420 nm) and a 45° angle of reflection, and they are transparent for the 800 nm wavelengths range. A half wave plat was placed after the periscope to restore the linear horizontal polarisation state. The angle and the wavelength range of reflection for each mirror help to filter out as much of the NIR beam as possible after it passes through the BBO crystal, as shown in the setup Figure 20.

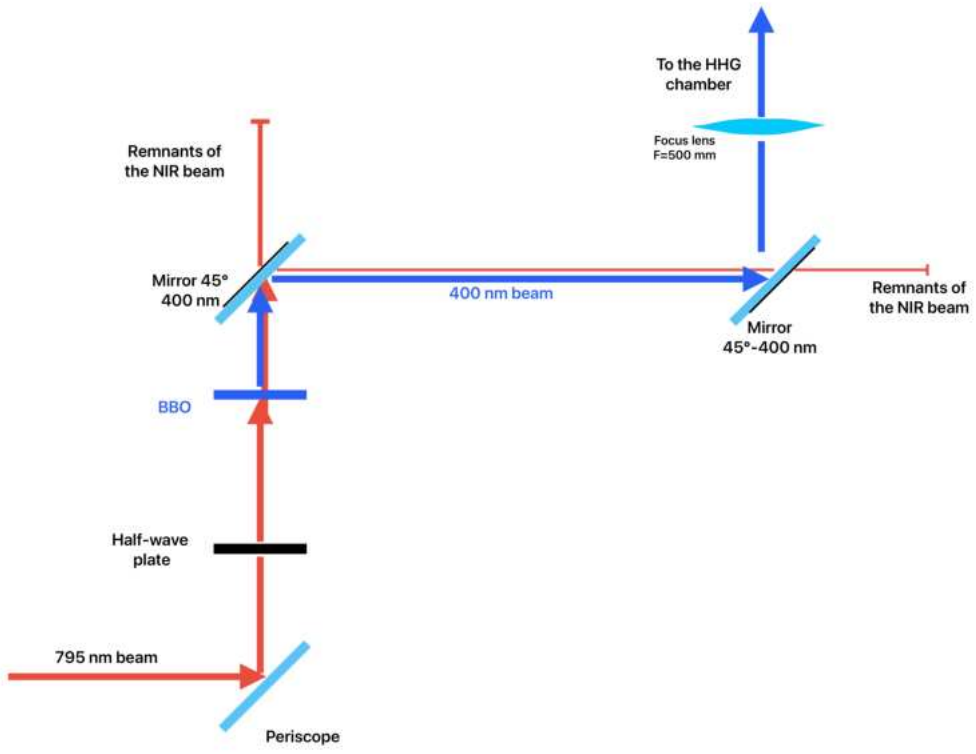


Figure 20: The optical layout of Figure 20 viewed from the top.

4.4 High Harmonic Generation Source and its Optical Layout

In the first experiment, the carrier wavelength of the NIR pulses is $\lambda_o = 795$ nm, and they have a duration of approximately 25 fs FWHM. The pulses are focused and directed to the HHG chamber to produce the HHG spectrum using a focus mirror (in the fist experiment) with a 62.5 cm focal length. In the second experiment a focusing mirror was replaced with a lens with a 50 cm focal length. This replacement is due to the new setup and the rearrangement of the beam optical layout in the second experiment, as shown in Figure 20. The focused NIR beam using a lens is directed to the HHG cell in the first measurement, and in the second measurement, the frequency of the beam is doubled with a BBO crystal and focused with the lens, then directed to the HHG cell.

Figure 22 shows the two cases of the second experiment (800 nm and the 400nm). A foil with a 100 μm thickness is glued on both the rear and the front beam passages of the HHG cell. In order to change the foil that is glued on the beam passages of the cell, the HHG cell is dissembled and lifted vertically along the Y-axis. Adjusting the position of the HHG cell along the Z-axis is possible, as shown in the latter reference, which plays an important role for the phase matching process, as will be discussed later. Note that for the first experiment which was conducted using an already available setup, focusing the NIR beam was achieved through a focusing mirror, unlike the second experiment, as shown in this figure.

The gas pressure inside the cell initially is ≈ 40 mbar while the argon gas flows into the HHG cell. The pressure at the HHG chamber, however, should be $\leq 10^{-4}$ mbar. This value will change as the pressure in the cell is changed due to the argon gas flow in the HHG chamber. To confine the volume of the gas, two aluminum foils with 100 μm thickness are glued on the rear and the front of the HHG cell, as shown in Figure 22. In the two experiments, the two foils on the rear and the front of the HHG cell are drilled by the

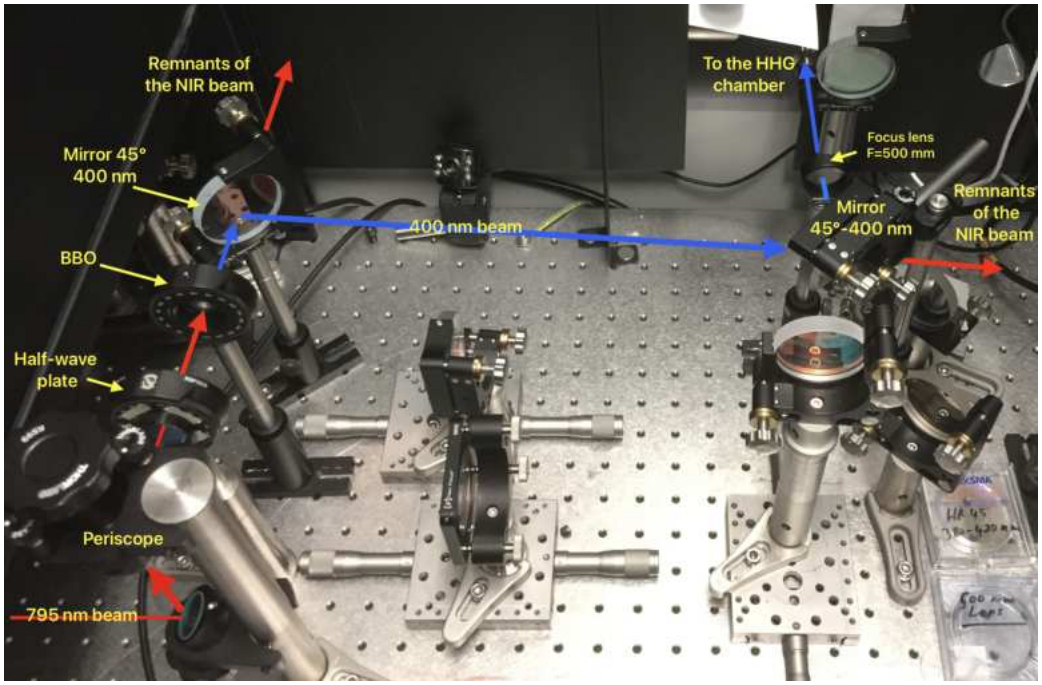


Figure 21: The optical layout that is used to obtain 400 nm from an 800 nm source and guiding it to the HHG chamber.

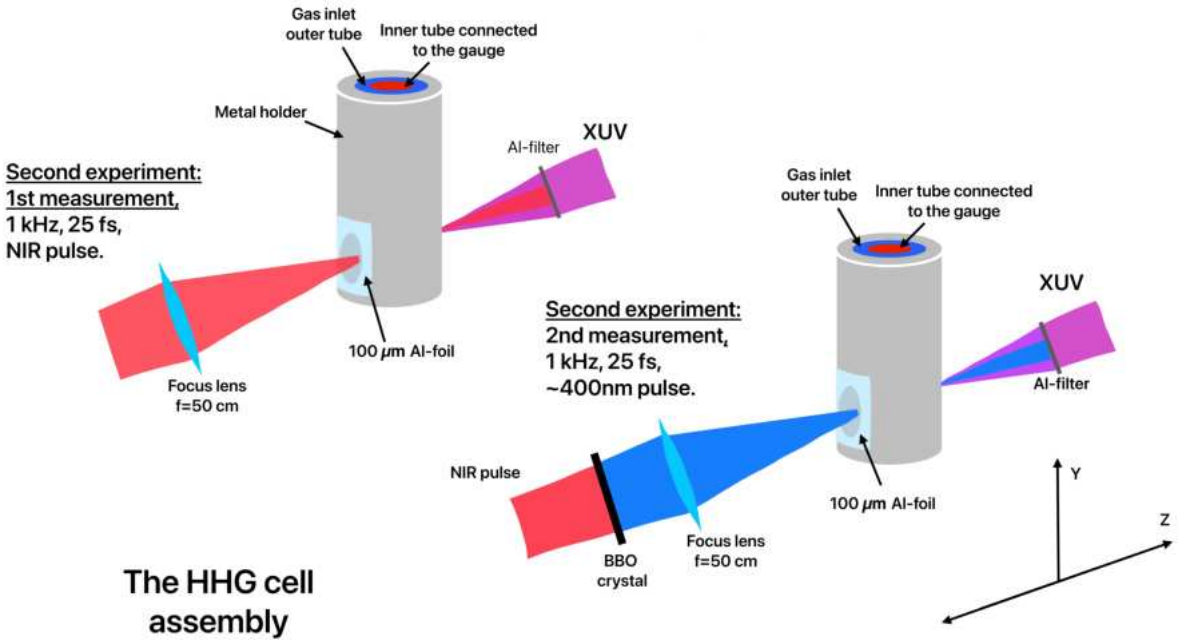


Figure 22: A schematic illustration of the HHG cell and the gas inlet in the case of the NIR beam and the 400 nm beam directed through the cell, resulting in an XUV radiation. The gas flows into the HHG cell using a gas jet.

laser beam, creating an approximately 100 μm diameter holes. By regulating the gas pressure in the HHG cell, the total phase mismatch can be adjusted ΔK_{tot}, so to increase the negativity of the Δk_{plasma} term balances the other terms in order to achieve $\Delta K_{tot} = 0$.

4.5 Gas Inlet of the HHG

The gas cell must be aligned in all three spatial dimensions. The gas cell is sealed off from the vacuum chamber with a 100 μm aluminum foil. This foil is drilled by the train of pulses of the laser beam. Once the HHG cell is mounted, it will be automatically aligned, directing the XUV radiation to the XUV spectrometer. The cell is mounted on the end of a straight tube that feeds the gas into the HHG cell. This tube attached to a translation stage that allows the cell to move along the horizontal axis, as shown in Figure 22, with ± 5 cm range of motion. The gas inlet line comprises two tubes inside each other. The inner tube is connected to the gas pressure gauge to provide a means to measure the gas pressure directly within gas cell. The gas flows into the HHG cell through the outer tube. The gas supply system includes a commercial control system from MKS, which is designed to maintain the pressure in the HHG cell during the measurement.

4.6 XUV Spectrometer

In all of the experiments, the energy of the NIR and the 400 nm beams that propagate through the HHG cell are largely unabsorbed. Therefore, these beams are filtered out with a 100 nm XUV-transmissive aluminum filter, as shown in Figure 22. The energy window of the aluminum filter ranges from 20-70 eV [41]. This filter is placed before the entrance of the beams to the XUV spectrometer.

The XUV spectrometer occupies an area of approximately 15 cm $\times$ 10 cm, and is placed in a chamber after the HHG chamber as shown in Figure 23. The beam path starting from the entrance of the spectrometer begins with a fixed slit that has width of 50 μm width. Following the slit is a grating that diffracts the HH, using a grating with 100 nm bar spacing. The diffraction pattern of the HH is captured by the microchannel plate (MCP) that is combined with the phosphor screen. The process of observing the HH—starting from the impact of the HH photons and ending with with the observation of the HH on the phosphor screen—is explained in details in Figure 24. A CCD camera is placed at the viewing window, as indicated in Figure 19, in order to record the spectrum displayed on the phosphor screen within the vacuum chamber. The detailed description of the spectrometer can be found in [42].

The MCP consists of a stack of two plates of micro fiber channels fused together to form a thin disc with a high electric resistance. Each plate has its own channels in a slightly tilted angle with respect to the surface of the MCP. The two MCPs together form a 'chevron' configuration, as seen in Figure 24. Once an energetic photon strikes the MCP, an electron is generated. An energetic photon in our case is in the XUV range of approximately 20 - 70 eV and is generated using an NIR driving beam. The electron—after being freed inside the micro channel—is accelerated by a static electric field, generated by the applied voltage of approximately 2.3 kV on the MCP stack. Thus, due to acceleration of the electrons by the static electric field, they hit the walls of the micro channel again, due to the titled design; as a result, their number is multiplied, as illustrated in the following Figure 24.

The thickness of each of the MCPs is approximately 1 mm. A voltage of around 1.15 kV is applied on each of the MCPs in order to generate a static electric field, and 4.3 kV is applied on the phosphor screen in order to make the phosphor molecules fluorescent. Both sides of the MCP stack and the phosphor screen are coated with conductive, transparent material. The detection starts with an energetic photon that creates a free electron. This free electron will gain energy as it is guided and accelerated in the micro channel by the applied static electric field. The accelerated electron will collide with the wall of the micro channel due to the tilted design of the channels causing to free additional electrons. The newly freed electrons will undergo the same process. The multiplication of these electrons over a 1-mm length of a single channel results in approximately 10^3 electrons. At the end of the second channel, the multiplication yield approximately 10^6

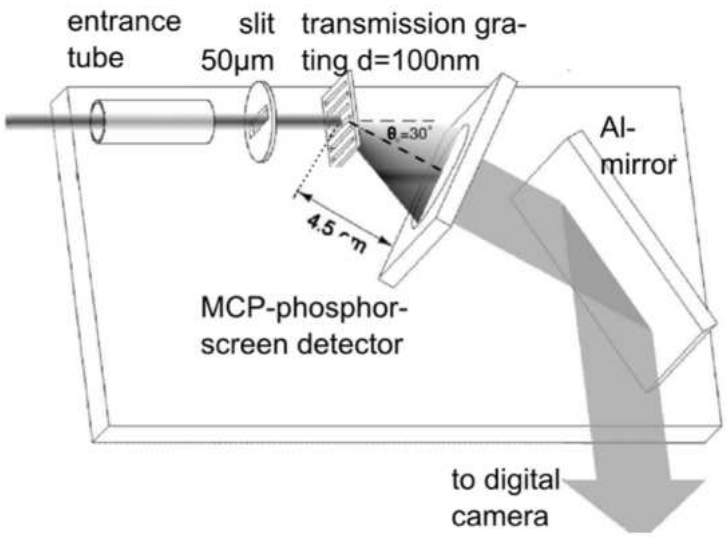


Figure 23: The XUV spectrometer, used to observe the HH spectrum that propagates directly from the HHG source, is located in a separate chamber, as shown in Figure 19. The spectrometer consists of a 50 μm slit, diffraction grating, MCP, phosphor screen, mirror, and a CCD camera. This figure is imported from [42].

electrons in the single bored tunnel across the two plates. Illustration of this paragraph can be found in Figure 24.

The MCP is followed by the phosphor screen detector with a maximum applied voltage of 5 kV. Once the phosphor screen is hit with the electrons pulse, florescent phosphor molecules are excited. Each florescent spot corresponds to an incident XUV photon on the opposite side of the MCP.

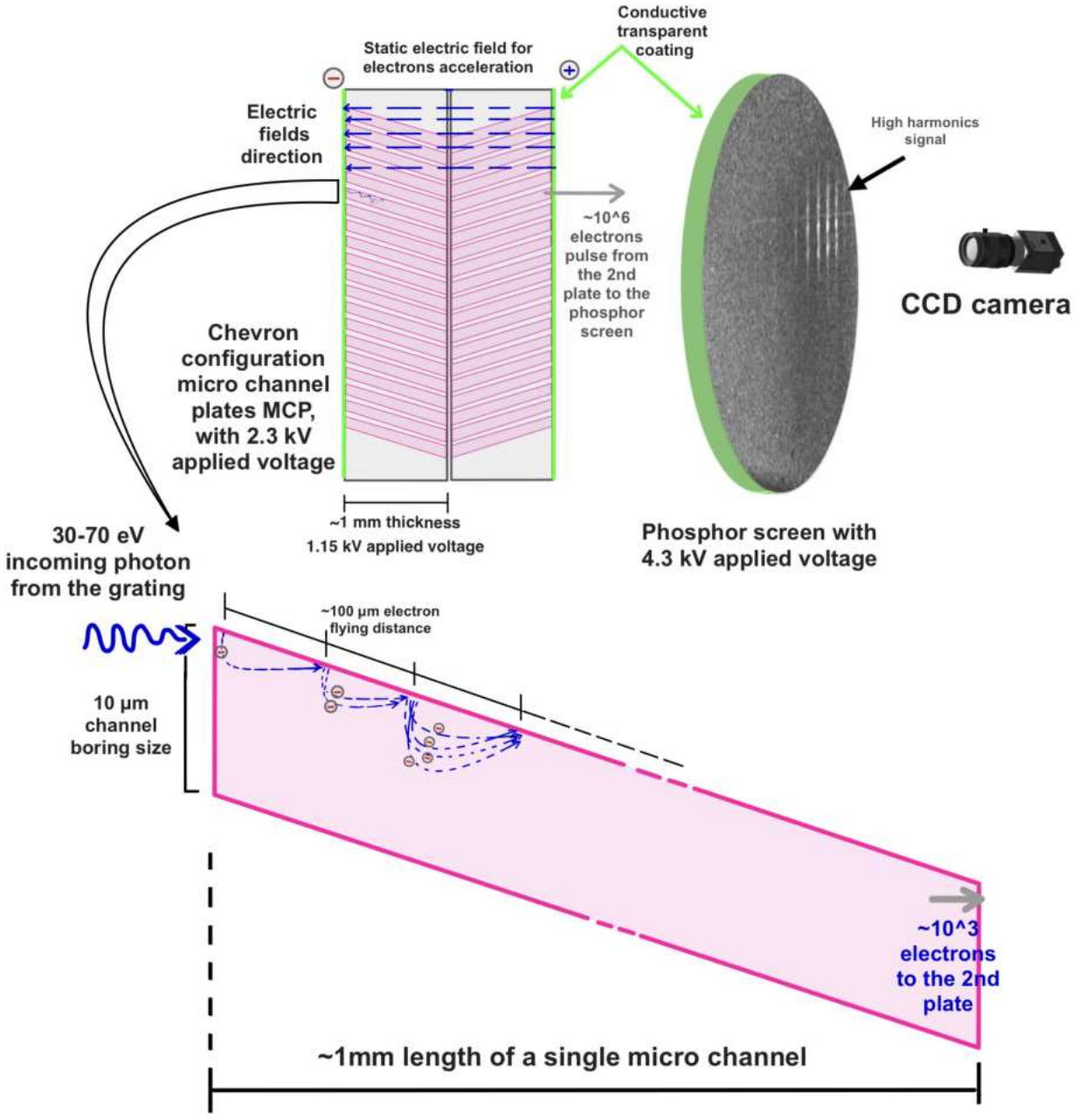


Figure 24: An illustration of the MCP and the phosphor screen which detect and display the XUV of the HHG radiation.

5 First Experiment: Phase Matching Investigation with an NIR Beam

5.1 Setup Characterisation

In this section, and in subsequent sections, the XUV spectrometer will be referred to as the 'MCP-phosphor detector' as well, to indicate the specific parts of the spectrometer that are being discussed.

The preparation of the setup begins by turning on the MCP-phosphor detector and setting it up to the working range. To do this, a positive voltage needs to be gradually applied to the MCP and the phosphor screen using a power supply. Due to the fragility of the MCP and the phosphor screen against rapid discharge, the application of the voltage has to be gradual. We started by keeping the MCP at zero voltage while slowly increasing the voltage up to $\approx$ 100 V on the phosphor screen. Then the voltage was slowly increased on the MCP up to $\approx$ 200 V while keeping the voltage on the phosphor screen fixed at $\approx$ 100 V, and so on, until the voltage on the MCP reaches the working range $\approx$ 2.3 kV, which is required to generated an electric field. At this moment, the voltage on the phosphor screen should be also $\approx$ 2.3 kV. Applying 100 V on the MCP means that a 50 V will be applied to each of the plates of the MCP.

Once the voltage on the MCP has been set to the working range, the voltage applied to the phosphor screen can be increased gradually up to its working range $\approx$ 4.3 kV in order to achieve fluorescence. According to the manufacturers of the MCP stack and phosphor screen, the slower this process of voltage application is for both parts, the safer it will be in order to avoid any discharge between them which would cause a permanent damage. The same method, in reverse, has to be used in order to switch the MCP-phosphor detector off. Furthermore, it is important to keep in mind that the **difference** in the applied voltage between the MCP and the phosphor screen should not exceed 2.5 kV while performing the steps of switching it on or off.

The next step of the setup preparation is the examination of the spectrometer by detecting the dark counts on the MCP-phosphor detector by observing glowing dots on the screen. The dark counts are detected despite that the XUV spectrometer is placed in a well-isolated thick aluminum chamber without any radiation penetrating it, given that there is no nearby source of energetic radiation. However, the dark counts can only be observed through high amplification, achieved by applying high voltages to the MCP-Phosphor detector [43], as explained previously. These dots are used as means to examine the functionality of the MCP and the phosphor screen in the XUV spectrometer. Furthermore, the dots—which can be observed with bare eyes in dark conditions—are used to examine how focused the CCD camera is on the objective. At this point the gauge measuring the pressure in the chamber, where the XUV-Spectrometer is placed, is switched off. This is done to avoid noise that could appear on the MCP-phosphor detector due to free-flying electron from the gauge. More information about this point in the next subsection 5.2.

With wearing the appropriate goggles that block the deployed NIR light, before removing the beam blocker, the laser beam has to be aligned to ensure that it is delivered to the 6 mm-long HHG cell, through which the argon gas flows. Using the CCD camera that is placed at the viewing point, as shown in Figure 19, we checked to see whether the beam passes through the HHG cell or not. Prior to that, the HHG cell had been sealed from the rear and the front with a 100 μm thickness aluminum foil to confine the argon gas inside the cell. By moving the HHG cell along the z-axis, one can adjust the position of the cell to ensure that the waist of the beam (the focus) is right before the HHG cell. This is where the positive Gouy phase shift occurs, allowing us to use the short trajectories, which compensate for the Gouy phase mismatch. Next, we tried to generate HH by regulating the gas pressure in the HHG cell and the energy of the driving laser pulse.

5.2 Data Collection

While setting the MCP-phosphor detector, an undesired glow was noticed across the phosphor screen. It turned out that the glow was caused by the XUV radiation that hits some metal in front of the MCP-phosphor detector, which in turn caused electrons from the metal to fly and hit the MCP. A negative voltage ≈ -30 V was applied to the front of the MCP-phosphor detector to repel these flying electrons. The outcome of this latter step is shown in the images of the phosphor screen 25.

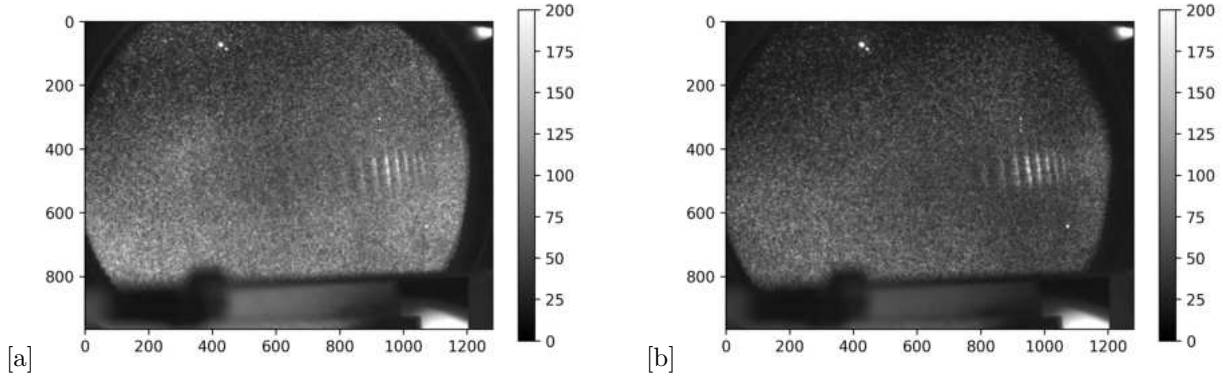


Figure 25: Two shots of the phosphor screen taken by the CCD camera. (a) An image of the phosphor screen before the removal of the glow. (b) An image of the phosphor screen after removing the glow.

Before the start of the data collection, it is necessary to set the proper exposure time of the CCD camera in order to collect a sufficiently visible signal from the HHG. Next, we examined the phase-matching of the HHG at different gas pressure values and driving pulse energies. For this, 4 different driving laser pulse energies were examined: 0.4, 0.84, 1.28, and 1.73 mJ. For each energy value, 13 different gas pressure values ranging from 5 to 65 Torr with +5 Torr step.

Throughout the measurements, the images that were taken by the CCD camera appeared to be misaligned. In the case of misalignment, the mean values of the columns will not be read properly, resulting in a distorted spectrum. As seen in Figure 25, there is a slight rotation due to the misalignment of the CCD camera, causing the HH lines to deviate from vertical position. By aligning the image and rotating it by -7°, the visibility is improved for a more accurate interpretation of the HH's mean values[1]. After the rotation of the image so that the HH lines appear vertical, the areas for the HH and noise along the vertical axis are determined by the dashed red line and the blue line. The mean values of these areas are presented in Figure 26(b) with the difference between them in an orange line. The red dashed line represents the area where the HH spectrum mean values are the highest. the area within the blue dashed line represents the background. This area is chosen for the noise for it is roughly homogeneous and where there is no undesired glowing across the image, so the subtraction from the spectrum would not cancel any valuable information. The red dots in Figure 26(b) represent the positions of the HH peaks along the horizontal axis, on which further calculations are performed. These red dots are found using the 'find_peaks' function of the 'scipy.signal' module. Within this module, one can control which peaks are detected by specifying a lower limit for the width of the peaks, a lower limit for the height of the peak, and the minimum distance between the adjacent detected peaks.

The Figures 27, are treated by specifying the area where the spectrum and the background noise lies and subtracting the mean values of the background from the mean value of the spectrum, see code Appendix D. The results are then displayed in Figure 28. Figure 28, shows the integrated spectra of the high harmonics, which are corresponding to the intensity of the HH. The positions of these peaks along the horizontal axis

[1]Note that rotation functions of different libraries in python may give different results of rotated images, which gives different results of peaks along the x-axis; consequently, affects significantly the calibration. For details see code Appendix D.

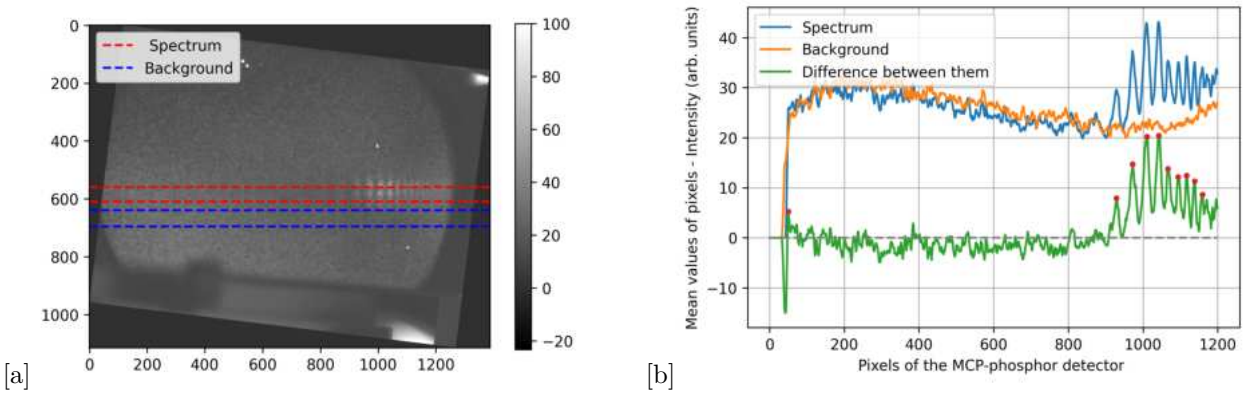


Figure 26: (a) The entire screen of the detector displays HH generated using 1.73 mJ driving pulse energy at 35 Torr gas pressure. (b) The corresponding spectrum, back ground, and the difference between them.

are subjected to curve-fitting procedure in order to assign each harmonic to its order as will be discussed in the next step.

In the first experiment, the same specified areas of the spectrum and the background in Figure 26[b] are used for all other measurements with different driving pulse energies (0.4, 0.84, and 1.28 mJ) at various gas pressure values. Using the same areas of the spectrum and the background for all other measurements in this experiment ensures the consistency of the calibration procedure and the intensity profile ratio among the different measurements to have consistent data interpretation across the various measurements. On these presented spectra, further calculations are performed, including the determination of the peaks positions, on which the calibration procedure is performed, as will follow next, and the heights of the HH peaks, which are presented in the subsequent phase matching maps.

The spectra shown in Figure 28, represent the results of the measurements within the range of pressures where phase matching is anticipated. The resulting HH from the higher than 55 Torr or lower than 20 Torr gas pressure values are not presented for brevity. However, their relative intensity values are presented later in the figures that shows the phase matching maps 33. Using different driving laser pulse energies helps in understanding the evolution of the HHG at different gas pressures and helps as well in determining the phase matching conditions at which the most intense HH emission can be obtained for given source parameters. Next, is the determination of the HH orders, which is done in the following calibration procedure using the curve-fitting.

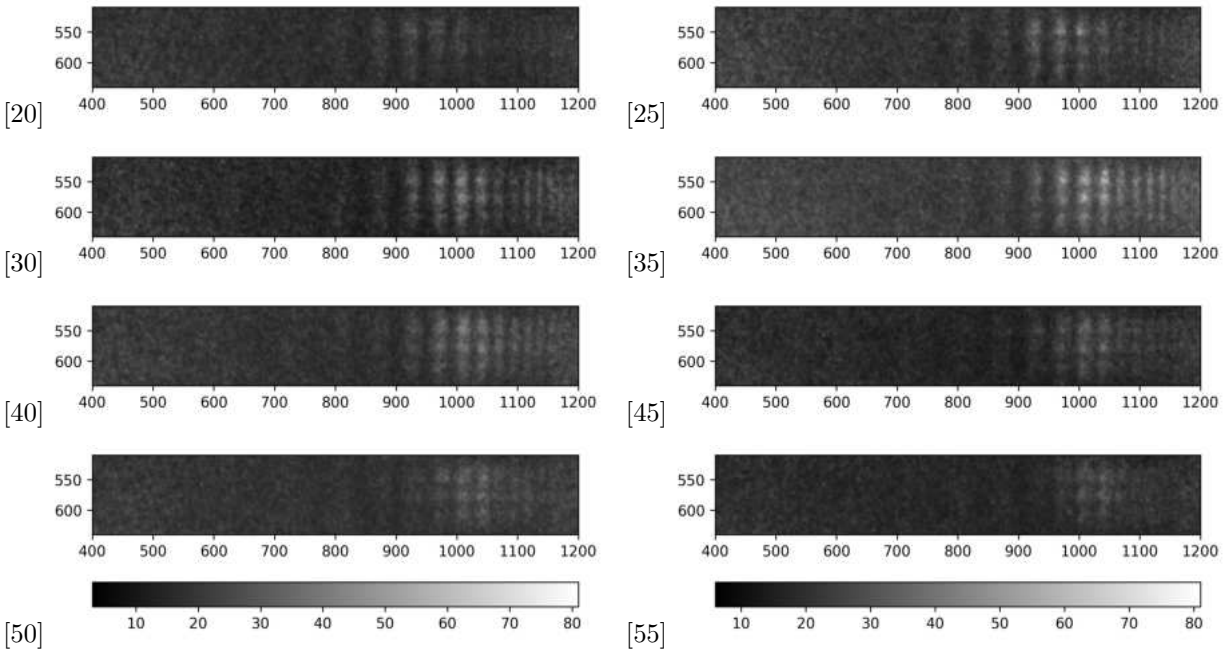


Figure 27: A number of the observed HH spectra by the CCD camera. The HH spectra are generated using a pulse energy of 1.73 mJ at 8 different gas pressure values (ranging from 20-55 Torr in +5 steps). This information is written in the lower left corner of each figure. The images show how the different HH spectra appear at different gas pressures.

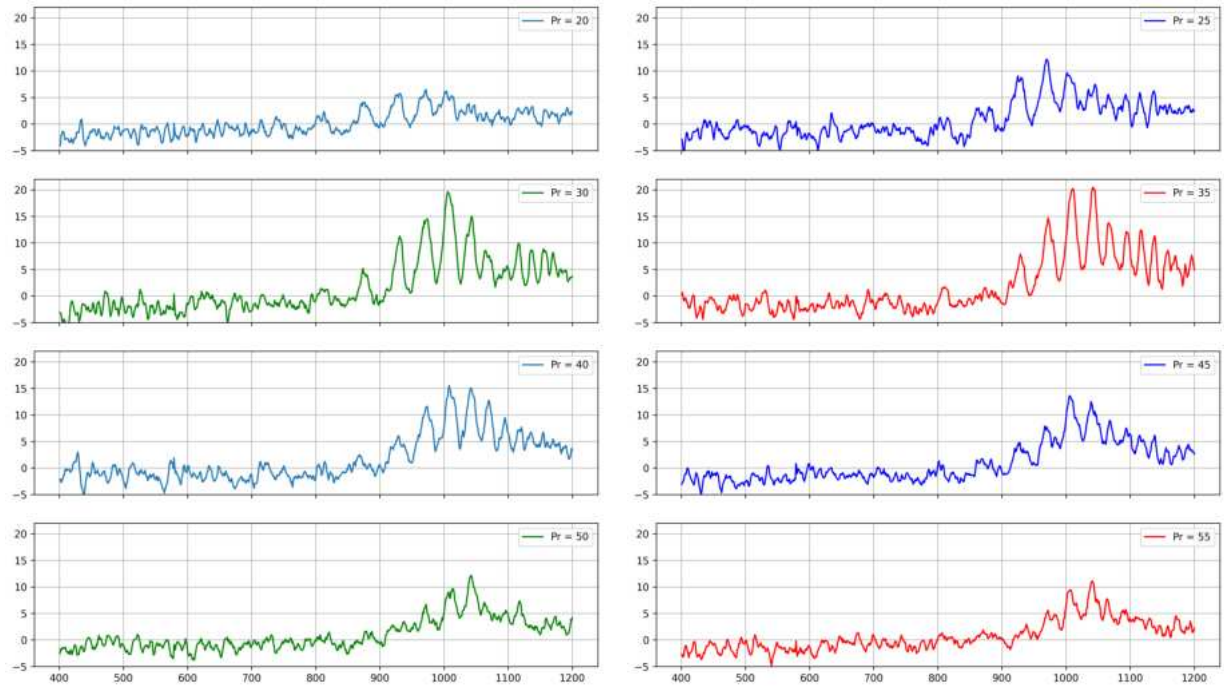


Figure 28: The spectra of the HH resulting from using the driving laser pulse energy of 1.73 mJ at different gas pressures (ranging from 20-55 Torr with +5 Torr step). The images are presented after the subtraction of the background from the spectrum of each image. More information about this point is available in the code Appendix D.

5.3 Calibration of the First NIR Experiment

In order to assign each harmonic to its corresponding order, a calibration procedure is required [42]. The aim of this procedure is to determine the relationship between the wavelength of each harmonic and the corresponding pixel (u_m) on the MCP-phosphor detector's screen. The calibration procedure is performed using the following equation:

$$\lambda_m = d\, sin\left[arctan(\frac{u_m - u_0}{A}) + \theta_0\right] \tag{23}$$

In this equation, λ_m is the wavelength of the m_{th} harmonic, and d is the period of the grating—which is 100 nm in our case. u_m is the position of the m_{th} harmonic on the horizontal axis of the detector's coordinate system, while u_0 is the position of the origin of the coordinate system. The coordinate system of the MCP-phosphor detector is shown in Figure 29. $\theta_0 = 30$ is the angle between the normal of the MCP plane and the direction of the beam, A is a constant that depends on the magnification.

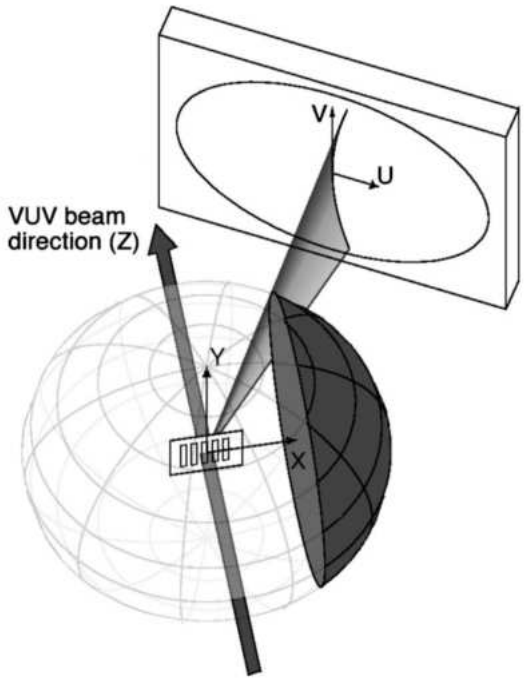


Figure 29: The geometry of the vacuum XUV beam diffraction illustrates the components of Equation (23). This figure is adapted from [42].

The parameters A and u_0 are unknown in the equation above. These parameters will differ with every new alignment because of changing alignment of the CCD camera. Once we manage to determine these missing parameters, it will be possible to identify the wavelengths of the observed HH on the detector's screen using Equation (23). Consequently, this enables us to assign the observed HH to their orders $N = \lambda_{(fund)}/\lambda_{(m)}$. The same calibration should be used for all the measurements of the same experiment as long as the same CCD camera alignment is being used.

The calibration procedure starts by selecting a number of HHs whose intensity peaks are distinct, as there may be 'false' peaks at the beginning or the end of the spectrum. These false peaks appear as a result of subtraction between a negative peak of the background line and the spectrum line; see the left-hand side

of the position 800 on the horizontal axis of Figure 26. Then, these positions of these peaks are manually recorded with respect to the horizontal axis of the detector's screen u_m. For example, the peaks which are obtained from the most intense spectrum, as shown in Figure 26, and used for the following calibration procedure are: $u_m = (873, 928, 971, 1008, 1041, 1066, 1094, 1116)$.

In fact we know that the HH orders are odd and spaced by 2, as mentioned in the section 2. Then these positions of the HH peaks, which are eight peaks positions for this calibration, are assigned manually to a sequence of eight harmonics wavelengths that start from the order, e.g., N=19, using the formula $([1/N, 1/(N+2), 1/(N+4), 1/(N+6), 1/(N+8), 1/(N+10), 1/(N+12), 1/(N+14)]) \times 795 \cdot 10^{-9}$ m). This sequence of harmonics as the example is plotted in the dashed blue line in each of the following Figures 30[c]. This blue dashed line represents the experimental data that was collected from the detector's screen via the CCD camera. The number of peaks positions must equal the number of the manually assigned harmonics wavelengths in the aforementioned formula. The model, on which the curve fitting is performed and the wavelengths of the HH determined, is given by Equation (23). This model uses the positions of the HH peaks u_m to determine the HH wavelengths. This model is plotted in a red dashed line in each of Figures 30. The sequence of HH is shifted manually upward or downward to obtain different HH sequences by changing the value N. It is then plotted in the blue dashed line in Figures 30. This sequence of HH wavelengths is compared with the wavelengths model in Equation 23. This model used the positions of HH peaks u_m obtained from the phosphor screen. As a criteria for comparison between the assigned wavelengths and the wavelengths obtained from the model, the sum of the square residuals (SSR) is used to determine which cure fit is the best. SSR $= \lambda_{assigned} - \lambda_{model}$. The smaller the SSR, the closer the wavelengths from the model are to the wavelengths of the assigned HH. Then based on the peaks positions obtained from the phosphor screen, the curve fitting is performed. The unknown parameters, A and u_0, can be determined through the curve fitting; consequently, the orders of the harmonics can be determined as well. The paper that describes geometry of the XUV spectrometer [42], suggests that the 15_{th} HH order appears roughly in the middle of the phosphor screen. The according to this information we specify roughly from which sequence N of HH wavelengths the investigation starts, in orders to determine the orders of the obtained peaks of the spectrum. This procedure is done once per experiment and CCD camera alignment. Then The unknown parameters u_0 and A of Equation (23) that are obtained by the curve fitting are (701.54068202, -1254.99749996]) respectively.

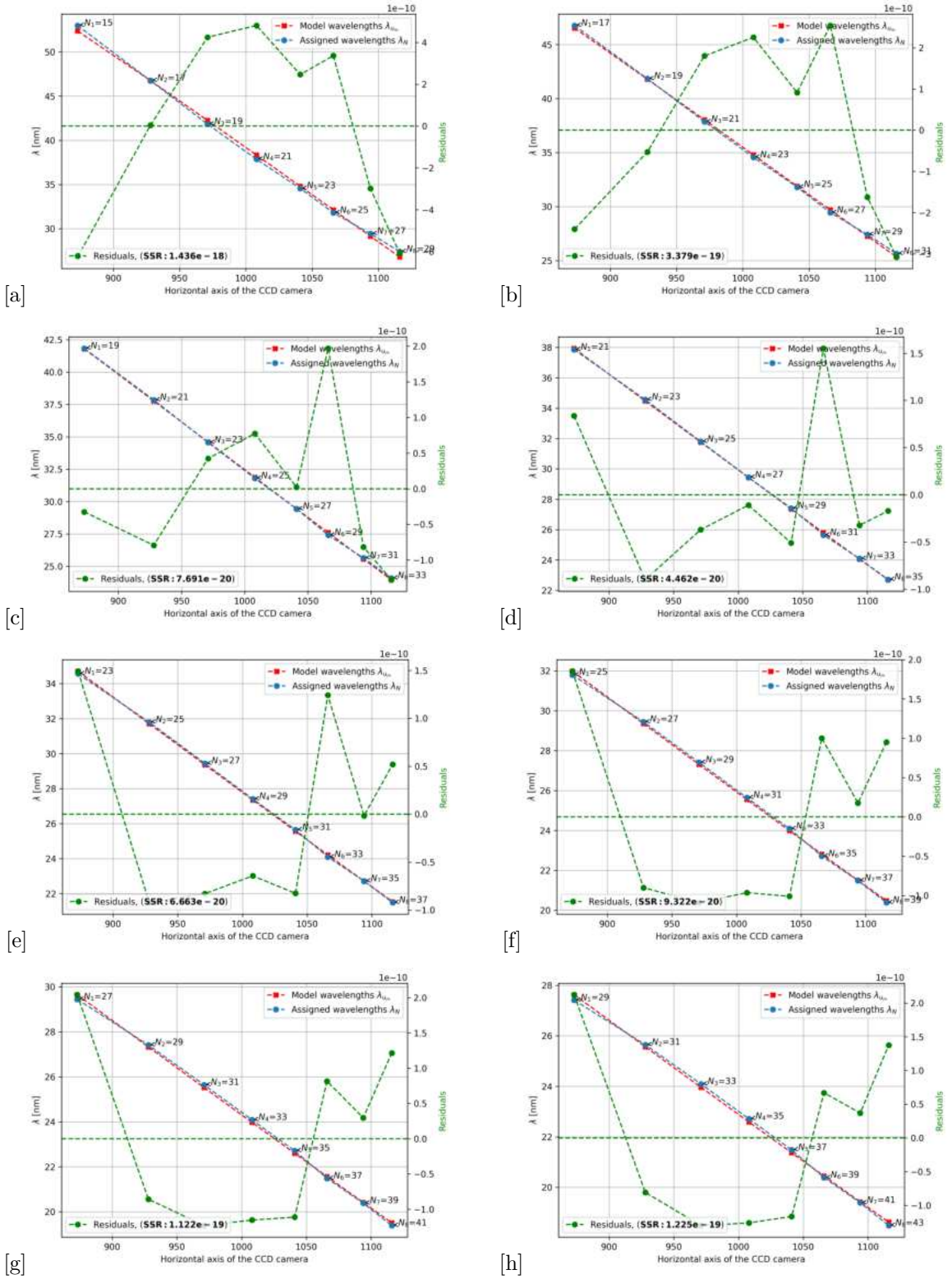


Figure 30: A curve fitting (represented by blue and red dots) is performed for different sequences of HH adjusted manually by N to determine the orders of the HH. The vertical axis represents the wavelengths of the examined HH. The horizontal axis represents the positions of the HH on the detector's screen. The green dots represent the corresponding residuals for each pair of blue-red dots. The residuals are measured with respect to the green dashed zero line. The SSR shows that the plot [d] has the best cure fitting. Therefore the correct sequence of HH starts at N=21 at the position 873 on the horizontal axis of the image taken by the CCD camera.

Then according to the results of calibration shown in Figures 30, the spectrum is plotted against wavelength and the harmonics order respectively is shown in Figures 26.

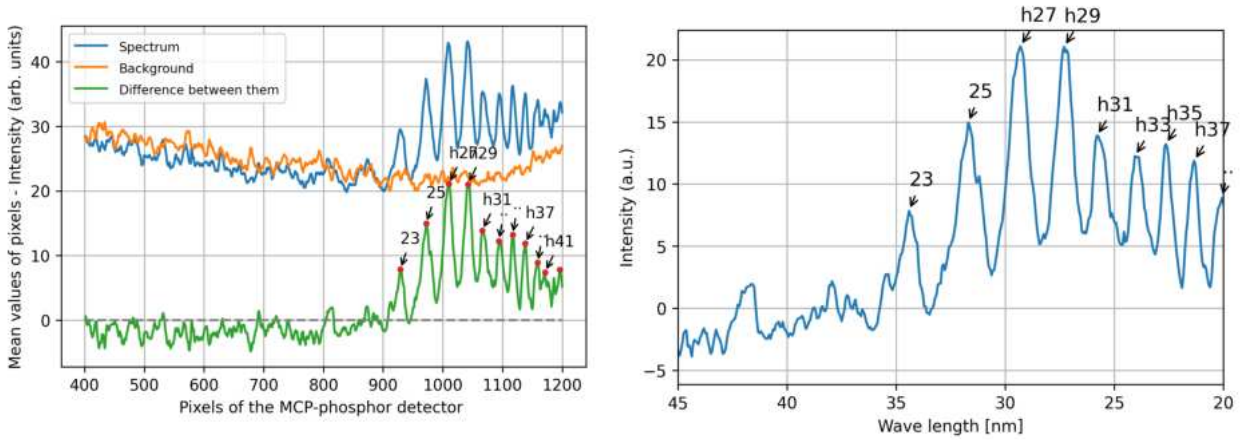


Figure 31: The HH, along with their positions on the detector's screen and wavelengths, are determined after the curve fitting procedure. This HH spectrum is generated using the pulse energy 1.73 mJ at a gas pressure of 35 Torr. For the figure on the right-hand side, the peak of the 21^{st} HH is not indicated, merely to avoid the detection of other undesired peaks, as explained in Figure 26.

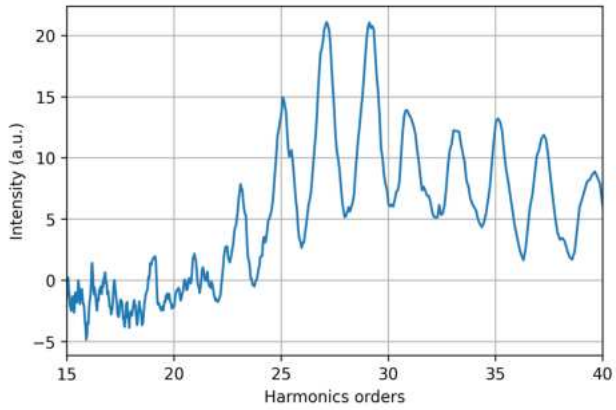


Figure 32: The HH spectrum are plotted as a function of their orders $N = \lambda_{(fund)}/\lambda_{(m)}$ after the determination of the HH wavelengths through the calibration procedure. The values of $\lambda_{(m)}$ are obtained from Equation 23. This HH spectrum is obtained using an NIR driving laser pulse of 1.73 mJ energy at a gas pressure 35 Torr.

To verify the results of this calibration, another calibration procedure for a spectrum obtained from a different driving pulse energy 0.84 mJ at the same gas pressure (35 Torr) can be found in Appendix A. There, the process is performed using the same parameters of $u_0 and A$.

5.4 Phase Matching Maps for HHG using 800 nm (First Experiment)

The first experiment was conducted through examining the phase-matching conditions of the HHG chamber for four different driving laser pulse energies (0.4, 0.84, 1.28, and 1.73 mJ). Each pulse energy value studied across 13 gas pressure values, ranging from 5 to 65 Torr with +5 step, in order to asses their impact on the phase matching process. As aforementioned, the control of the gas pressure—namely the density of the plasma—will be the mean to level out the other mismatch wavevectors and achieve $\Delta K_{tot} \approx 0$. In the next

subsection, there will be discussion of the results regarding the various utilised driving laser pulse energies and the different gas pressure values for each pulse energy.

After determining the orders of the HH through the calibration procedure, It is possible to plot the intensities of individual harmonics versus the HH order and all gas pressure values for the four values of pulse energy in this experiment. The phase matching maps shown in Figure 33, give an overview over all the experimental data of all the measurements for this experiment. The phase matching maps show that the phase matching conditions become optimal between 30 and 40 Torr. At these gas pressure values with the different driving laser pulse energies the longest span of the generated harmonic orders and the most intense emissions can be obtained. The most intense harmonic at different gas pressure values for all the pulse energies (0.4, 0.84, and 1.28 mJ) is the 27^{th}. With the 0.4 mJ pulse energy, however, the intense harmonics at almost all of the gas pressure values is the 25^{th}.

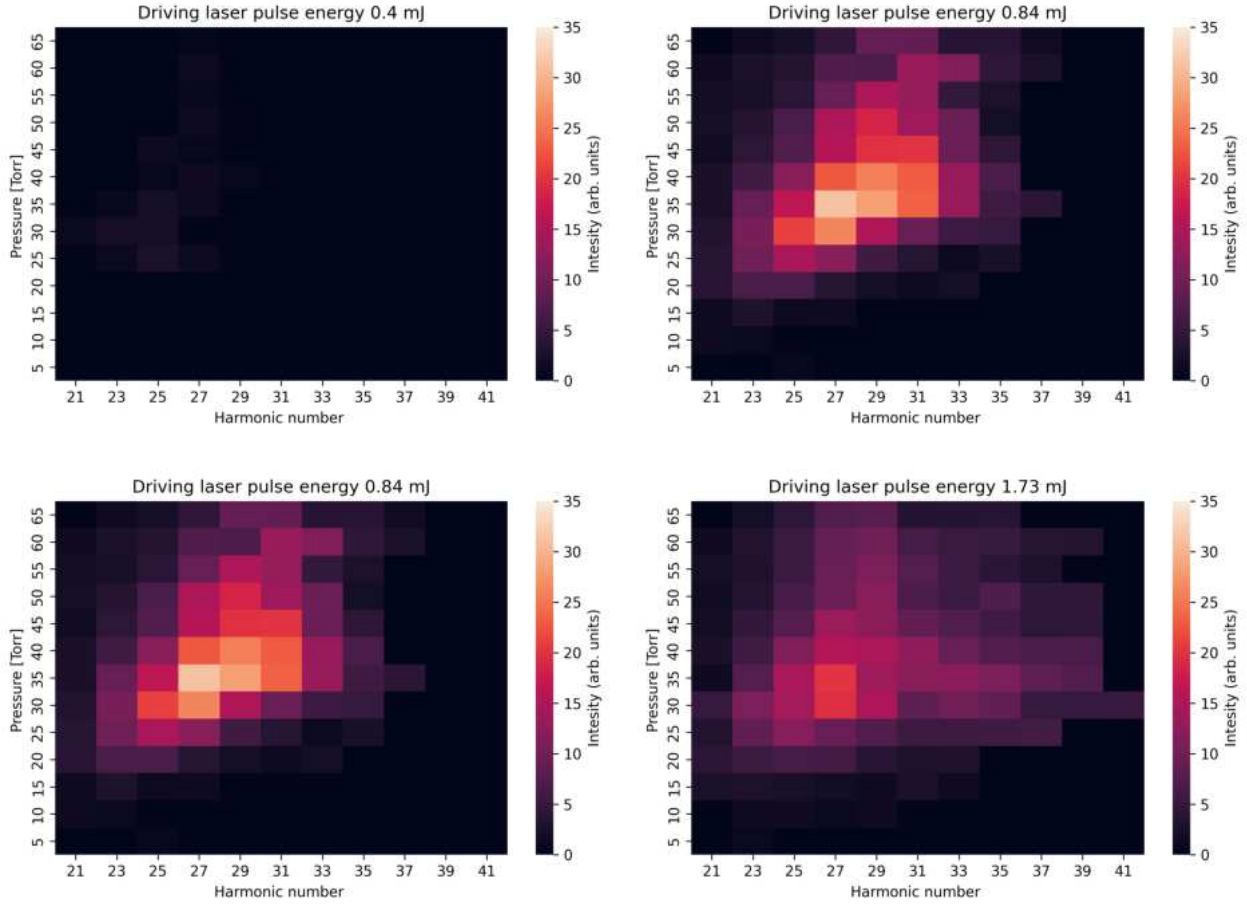


Figure 33: Heat maps for the generated HH at the gas pressure values (5-65 Torr) using different driving laser pulse energies (0.4, 0.84, 1.28, and 1.73 mJ). The phase matching maps show that the most intense obtained HH for all energy pulse values is the 27^{st}.

Note, that the intensity calibration of the XUV spectrometer is not implemented, in reality, a different HH may be more intense than what is observed

5.5 Discussion

Looking at the maps in Figure 33, a number of observations will be discussed regarding: 1) The observation of gas pressure values at which the most intense HH were obtained and the intensities of the HH that indicate the optimum achieved phase matching. 2) The observation of the few HH when using 0.4 mJ pulse energy. 3) The agreement of the generated HH in comparison to previous works. 4) The agreement between the experimentally generated HH and the theoretical predictions. 6) The observation of split in some HH peaks as shown in Figure 36. 7) The variations in the positions of harmonic peaks when different driving laser pulse energies are used as shown in the table 1.

Looking at Figures 33, one can see that the most intense emission occurs for orders 25^{th} and 27^{th} with the pulse energies of 0.84, 1.28, and 1.73 mJ at the gas pressure values is between (30-35 Torr). The highest emission with the driving laser pulse energy 1.73 mJ occurs with the order 27^{th} at 35 Torr gas pressure value. Therefore, the phase matching is roughly achieved between (30-40 Torr). We can see that the intensity of the HH is decreased in the case of 1.73 mJ pulse energy in comparison with the other driving laser pulse energies. This may be due to the increase in the density of the plasma which is proportional to the dispersion. This tells that it is hard to have HH while the majority of the conversion medium is ionised.

The intensity of HH using the pulse energy 0.4 mJ are the least with respect to the other driving pulse energies. The lowest used driving pulse energy 0.4 mJ is inadequate to ionise enough atoms to produce a significant number of photons that can be observed by the detector. This can explain the low intensity of the generated HH in Figures 34, and the heat map Figure 33 for the pulse energy 0.4 mJ. The insufficient pulse energy leads to a diminished intensity of HH. We can see that the only harmonic that appears almost across all of the gas pressure values for this pulse energy 0.4 mJ is the 27^{th} harmonic which is the most intense as well with respect to the other driving laser pulse energies, as the phase matching maps in Figure 33 show. Th same can be applied for the other energy values. We notice as well with 0.4 mJ pulse energy that the only gas pressure value at which two consequent high harmonics generated are (25, 30, and 35 Torr). This confirms again that the phase matching conditions occur at this range of gas pressure. The following Figures 34, the spectrum versus the orders are presented at gas pressures (30-40 Torr) where the phase matching, and consequently the most intense HH are expected.

It can be seen in Figure 34 that there is a slight shift in the positions of the HH using this pulse energy. The first observed HH at gas pressure 30 Torr is not aligned exactly at the 25^{th} order, but rather slightly shifted. This behavior is explained with respect to the other spectra in the table 1 and Figure 37.

In order to verify how plausible the former presented results are, we can compare it to a well known works which uses roughly similar conditions [44]. This work shows the generation of HH with argon gas under (10-20 Torr) gas pressure values:

The work [44] shows 9 odd HH orders between 11^{th} and the 27^{th} using argon gas as a conversion medium and using 40 fs linearly polarized 800 nm Ti:sa pulsed laser with 1kHz repetition rate with using a lens of 1-m focal length. The 15^{th} reaches the peak intensity at $\approx$10 Torr gas pressure value where the phase matching occurs. While the work of this thesis, the obtained HH are between 21-41 using $\approx$ 25 fs with wavelength 795 nm of a Ti:sa linearly polarized pulsed laser with 1 kHz repetition rate, and with using a focusing lens with of 500 mm focal length, as shown in the phase matching maps Figures 33. In this work the peak intensity appears with the harmonics 27^{th} and 29^{th}, with all the driving laser pulse energies at (30-40 Torr) except for 0.4 mJ. In the first experiment of this thesis, the extent of the observed HH using the argon gas extends from the 21^{st} HH to the 41^{st} HH in the case of 1.73 mJ driving pulse energy, eleven odd HH were observed. The cut-off energy of the comparing work is approximately 43 eV, while the cut-off in this thesis exceeds the 60 eV for the first experiment. This cut-off energy is in the case of 1.73 mJ pulse energy is higher cut-off

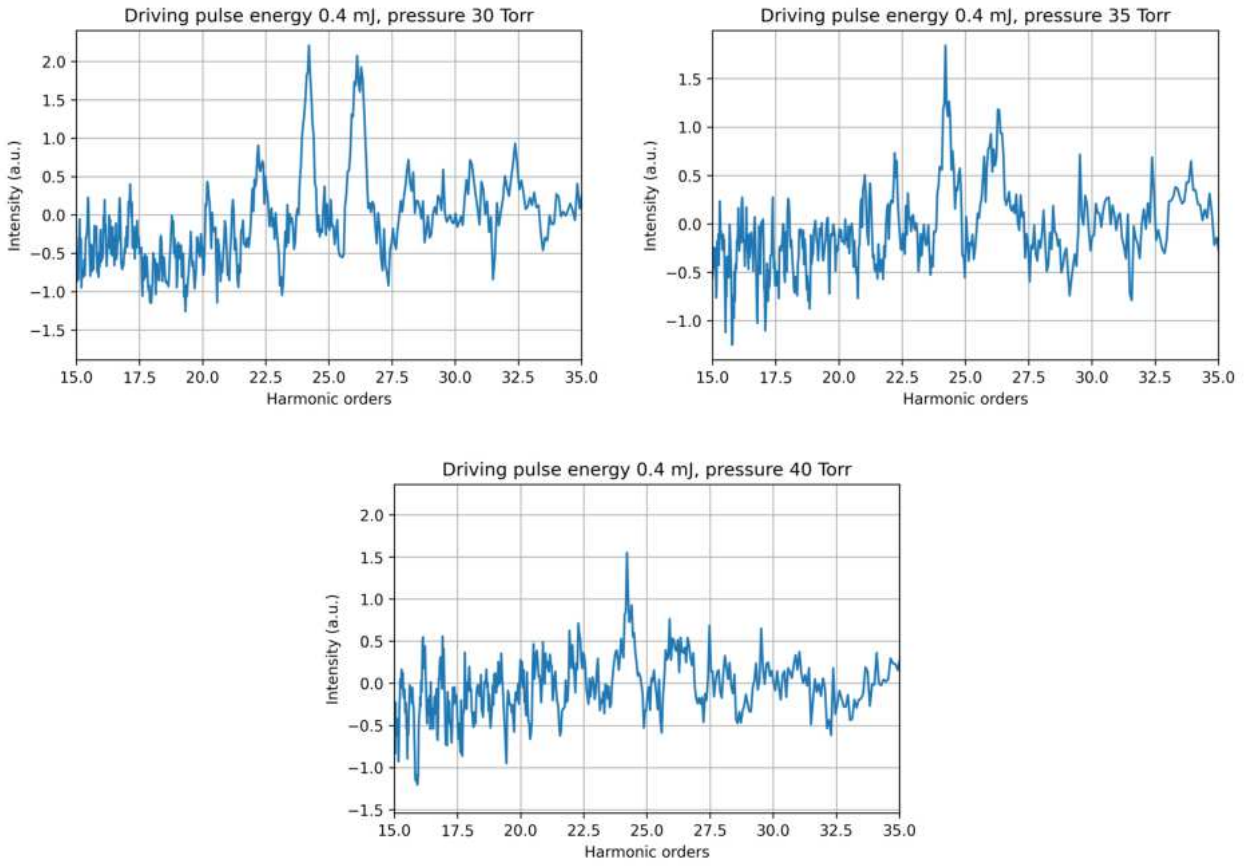


Figure 34: The spectra for HH obtained using driving laser pulse energy 0.4 mJ at gas pressure values (30, 35, and 40 Torr) where the phase matching conditions are optimum. The HH are presented in this range of orders to observe clearly, to which order the harmonics are assigned.

using lower pulse energies, as shown in Figure 36.

The phase matching conditions for the comparing work requires less gas pressure with respect to our case. The difference in the optimum gas pressure (less gas pressure for the phase matching in the comparing matter) can be due to the usage of a longer laser pulse duration which has ≈15 cycle (for 40 fs, 800 nm laser source). This means that the conversion medium atoms encounter more cycles per pulse whose effective phase windows can subject the gas atoms to the process of the three-step model, as explained in Figure 9. Consequently, the parameters of the pulse of the work [44], namely the pulse duration, can lead to larger amount of plasma in the gas cell with respect to the parameters of our experiment, which would require less gas pressure ≈ 10 Torr in their case order to achieve the phase matching, as explained in the section 3.

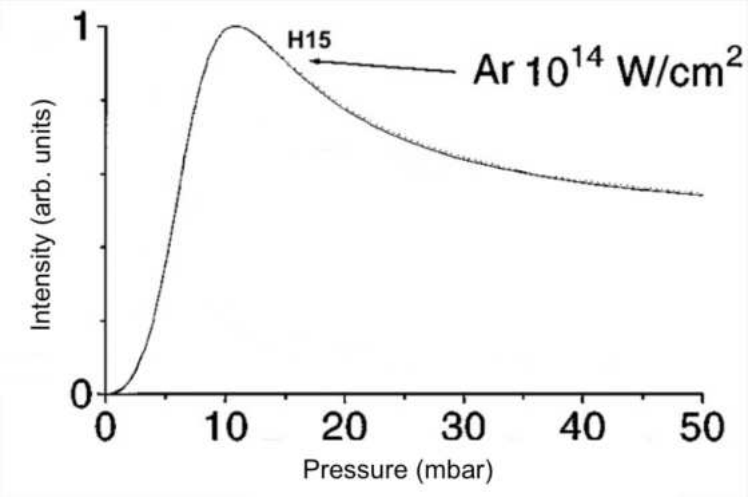


Figure 35: The intensity evolution of the 15^{th} harmonic at different gas pressure values. The harmonic is generated with 40 fs pulse of a linearly polarized Ti:sa laser using argon gas as a conversion medium. The emission of the harmonic shows its peak around 10 Torr. This figure is adapted from [44].

With increased driving pulse energy ranging from (0.4, 0,84, 1.28 up to 1.73 mJ), we notice that the cut-off energy increases as well with increasing the driving pulse energy. The cut-off exceeds the 60 eV when using 1.73mJ. The energy values of the harmonic spectrum show a good agreement with the three-step model. The highest cut-off value in this experiment resulted from 1.73 mJ input pulse energy, as shown in Figure 36

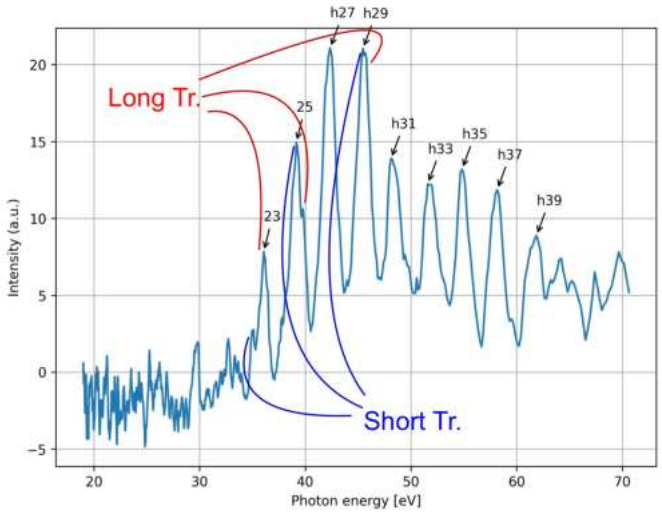


Figure 36: The photon energy values of the resulted high harmonic spectrum generated using the pulse energy 1.73 mJ under 35 Torr gas pressure. We notice in the energy spectrum the split of the peaks.

We notice that there is a pronounced difference in the range of the generated HH and their intensities when comparing the results of the driving laser pulse energies of 0.84 mJ and 1.73 mJ at the gas pressure values. This can be due to the overall intensity increase once the pulse energy is increased. The overall amplitudes of every single cycle will increase as well. As a result, this extends the window of the phases at which electrons trajectories return back to the parent ion; consequently, this increases the plateau of the HH spectrum. More explanation on this point can be found in subsection 2.2 and Figure 9. This can explain the extension in the orders of the generated HH using 1.73 mJ pulse energy with respect to the HH generated using 0.84 mJ pulse energy. By increasing the pulse energy, there is an unavoidable increase in the generation of plasma, which, as described in section 3.2, leads to a higher phase mismatch with the pulse. Additionally. the atomic phase mismatch is low for small intensities of the driving laser, but it increases once the driving pulse energy is increased for a fixed beam focus, at least for long trajectories as Figure 18 suggests. This leads to greater phase mismatch between the NIR beam and the generated XUV harmonics which affects destructively the resulting HH. More details on this point can be found in the subsection 3.4. The increase in the mismatch can explain the decrease in the most intense harmonic generated using pulse energy of 1.73 mJ with respect to the most intense harmonic generated using 0.84 mJ and 1.28 mJ pulse energies.

There is a pronounced split in peaks of each harmonic can be seen in Figure 36. As mentioned in the section 2.2 and shown in Figure 12, that any kinetic energy of the HH can result from two different phases pre and post the optimum phase $\approx 18°$ that yields the cut-off energy. The pre and the post-phases correspond to the short and long trajectories consequently as explained. This means that two different trajectories contribute in generating a single peak. This can explain why for some HH there is a pronounced split in the peak of the harmonic. More details on this point can be found in [22]. This shows again an agreement with the theoretical prediction with respect to contribution of the long and short trajectories in generating the same HH.

Observing the variation of the peaks positions, i.e. the wavelengths, for the same HH order as the driving pulse energy varies. The increase in the overall energy of the pulse can induce the self-phase modulation (SPM). The SPM occurs due to the modification of the refractive index which modified by increasing in the intensity. $n = n_0 + n_2 I$, n_2 is the non-linear refractive index. This causes different frequency components of the pulse to travel at different speeds, leading to spectral broadening and chirping for the pulse. This can explain the slight variation in the wavelengths as the pulse energy increases as shown in the table 1.

$21^{st}H$	$23^{rd}H$	$25^{th}H$	$27^{th}H$	29^{th}	$31^{st}H$	$33^{rd}H$	$35^{th}H$	Pulse energy (mJ)	Gas pressure (Torr)
873	928	971	1008	1041	1066	1094	1116	1.73	35
869	925	969	1005	1038	1065	1092	-	1.28	35
-	914	959	998	1031	1061	1086	1108	0.84	35
-	915	959	999	1031	1060	-	-	1.73	40
-	925	965	1006	1040	1066	1089	1113	1.28	40
-	915	962	997	1032	1061	1087	-	0.84	40

Table 1: This table shows the positions of the HH peaks u_m. The data are taken from three driving laser pulse energies (0.84, 1.28, 1.73 mJ) at the pressure values (30, 35 Torr) where the phase matching of the HHG occurs. Another illustration for variation of these peaks in Figure 37.

The curves in Figure 37 show a slight shift to higher order (lower wavelength) for a each HH once the energy of the driving pulse is increased. Furthermore, The 25^{th} HH obtained from the lowest used pulse energy 0.4 mJ shows shifting towards lower orders, namely towards higher wavelength, as shown in Figure 34. The highest obtained relative intensity using the energy 0.4 mJ is with respect to the other spectra is insignificant, therefore, it is not presented here in Figure 37. The curves resulting from different energies also show the variation of the cut-off for each pulse energy. Some cut-off peaks of the different spectra were not listed in

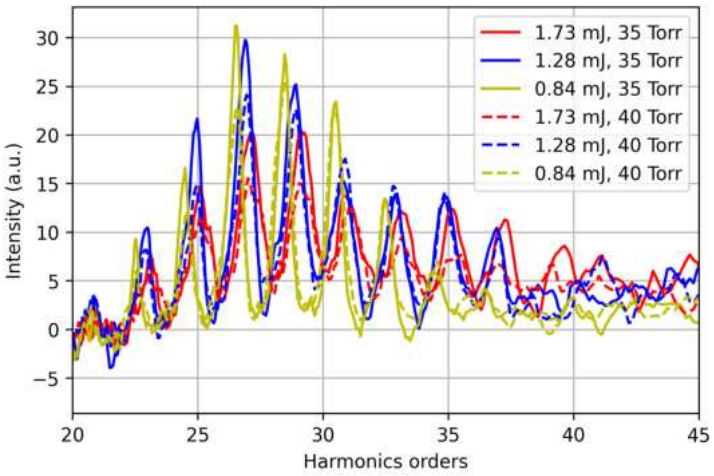


Figure 37: Different generated HH spectra using the driving pulse energies: 0.84, 1.28, and 1.73 mJ, each of them at the gas pressure 35 and 40 Torr, where the phase matching is roughly achieved in this experiment.

the table 1, but can be seen observable in Figure 37.

6 Generation of High Harmonics Using the 400 nm Fundamental wavelength and Comparison with the 800 nm (The Second Experiment)

The next section describes HHG with 400 nm light and compare it with the HHG by 800 nm. For this purpose, the setup was modified as described in subsection 4.3 and 4.1. To have a direct comparison, the same setup was used for both the 400 nm light and again with the 800 nm. The latter was done by removing the BBO and replacing the 400 nm reflecting mirrors with the 800 nm ones.

6.1 High Harmonic Generation Using 800 nm Source

Most of the parts of the setup of this experiment are the same as the recent experiment as mentioned in the subsection 4.4. The same HHG chamber, vacuum system, and the XUV spectrometer were used, but a new optical layout. The steps of the switching on the MCP-phosphor detector were followed, as mentioned in the subsection 5.1, by applying the voltage gradually on each of the MCP and the phosphor screen and setting them up to the working range. Examining the functionality of the detector was used through observing the dark counts. In this part of the experiment, three different driving pulse energies were used to generate HH: 1.44, 1.8, 2.16 mJ. Each of these energies examined across 10 gas pressure values ranging from 13 to 31 Torr with +2 Torr step.

Since this experiment has its own alignment, a new calibration for the HH order must be done. The 400 nm measurements yield only two harmonics in the best cases, as will be shown later. The positions of these harmonics are insufficient to run the calibration procedure. Therefore HH were generated using the 800 nm beam using the same setup and alignment of the 400 nm. The aim of generation the HH using the NIR is to be used as a reference for assigning the HH resulting from the 400 nm beam to their orders. In this part of the second experiment, we took ten shots per measurement by the CCD camera in order to calculate their average, a sample of an averaged and a single images can be found in Figure 38, all the shots that were taken for this prat of the experiment can be found the Appendix B with more details. The HH spectrum that was used for the calibration is shown in Figures 38.

Throughout the first measurements of the second experiment, we took 10 shots per measurement in order to calculate the average of these ten shots per measurement. The regions of the spectrum and the background of the 38[b] image determined in the same manner as explained in Figure 26. The peaks dots are determined also as shown in Figure 26.

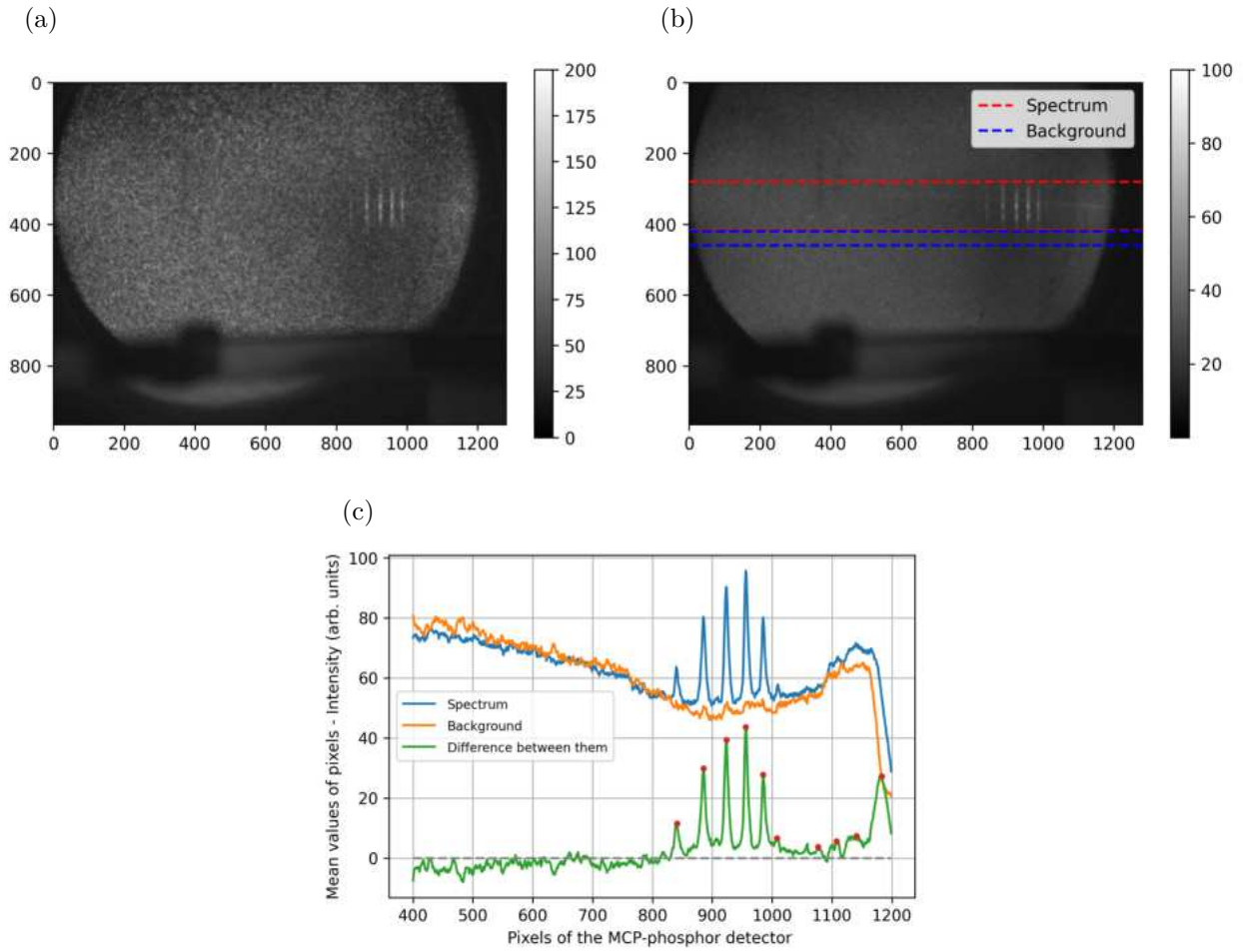


Figure 38: These HHs are generated using 800 nm driving laser beam with 1.8 mJ pulse energy at gas pressure of 21 Torr in the HHG gas cell. (a) A single image and (b) an averaged images of a single measurement of the HHG using an 800 nm driving beam collected from the detector's screen via a CCD camera. (c) Is the result of the mean values of the spectrum, the background, and the difference between them. The mean values are taken from the areas between the red and the blue dashed lines.

The calibration procedure of the NIR beam is done in the same steps that were done for the first experiment using Equation (23), see the subsection 5.3. The positions of the HH used for the calibration and are taken from the curves in Figure 38 are: u_m = (840, 884, 923, 956, 986). The unknown values determined by curve fitting (u_0 and A) are: (670.01296871 -1253.31351353) respectively. The results of the calibration are shown in the following and the subsequent Figures:

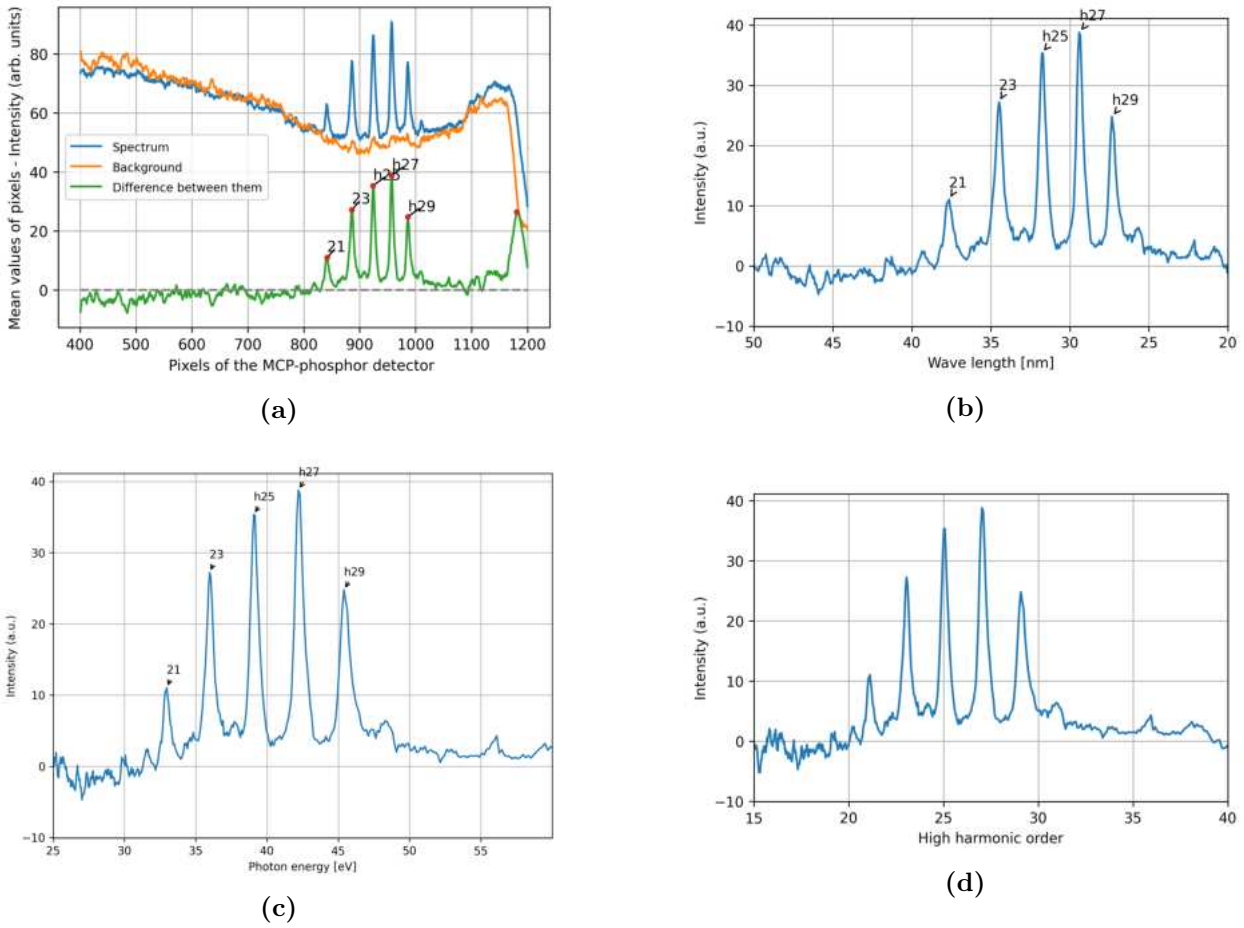


Figure 39: This spectrum is generated using 800 nm source and 1.8 mJ pulse energy at 21 Torr gas pressure. (a) The generated HH with respect to their positions on the detector's screen and (b) with respect to the wavelength after determining their orders. (c) HH spectrum versus the photon energy. The cut-off energy in this measurement is $\approx$ 48 eV with the consideration that the peak at the 31^{st} order is a HH. (d) HH versus their orders.

6.2 Phase Matching Maps for HHG using 800 nm (Second Experiment)

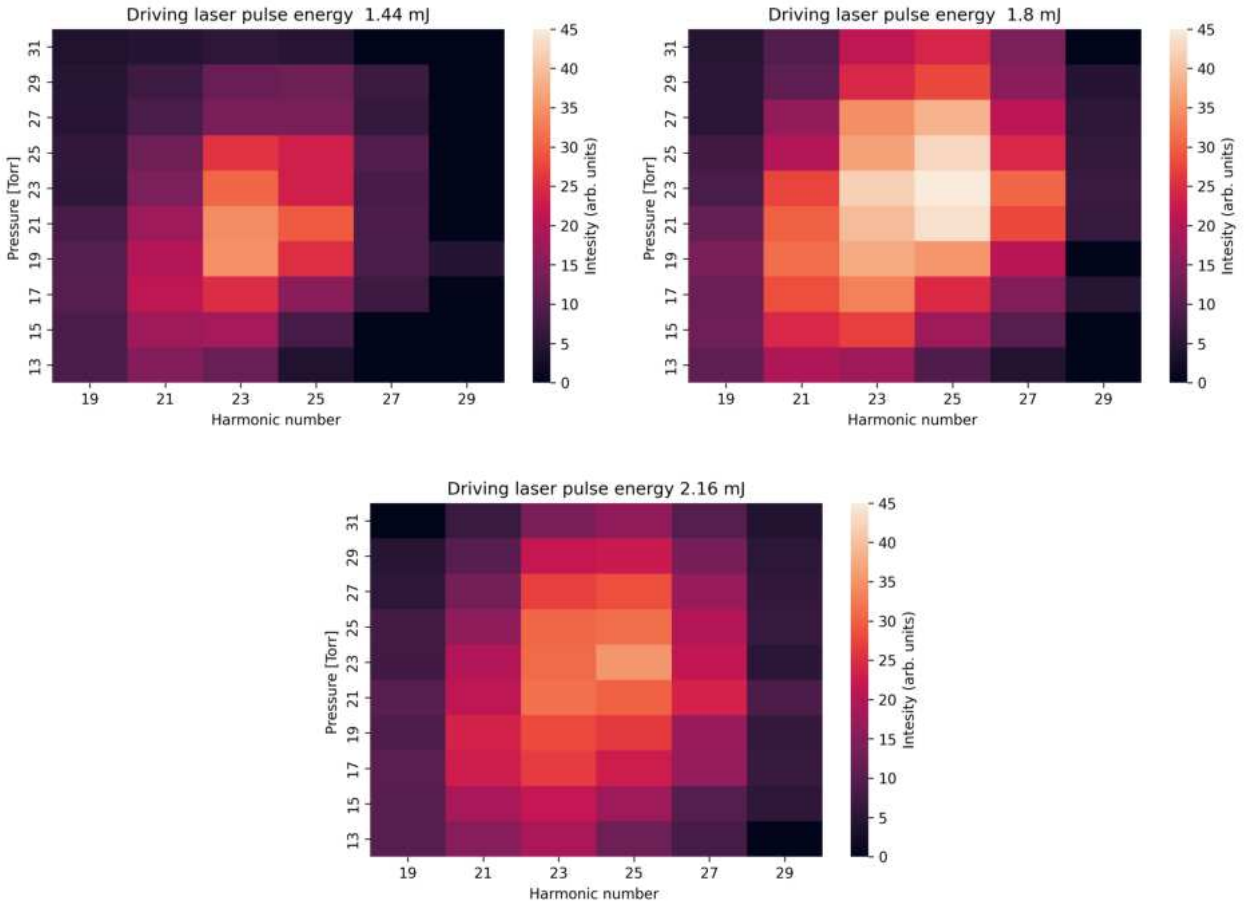


Figure 40: The phase matching maps of the HH obtained using (1.44, 1.8, and 2.16 mJ) driving pulse energy with 10 different HHG cell gas pressure (13 to 31 Torr) with +2 step. The phase matching maps show the relative intensities of the HH and their orders.

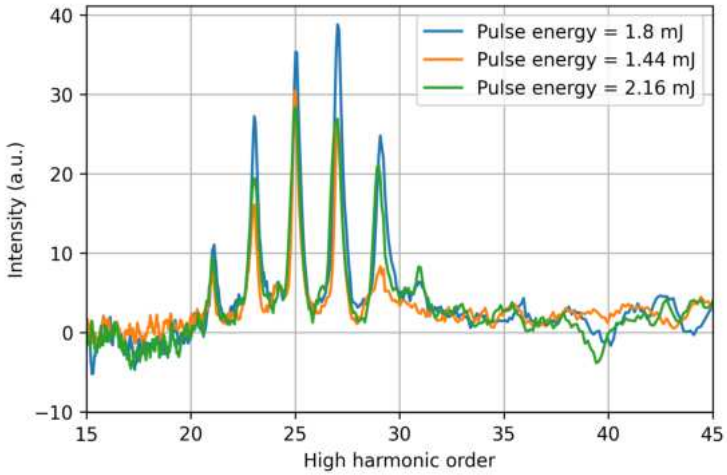


Figure 41: Spectra of the HH generated using three different driving pulse energy values: 1.44 mJ, 1.8 mJ, and 2.16 mJ. All the spectra are obtained at gas pressure 21 Torr.

6.3 Discussion

In this discussion, we will address the facts that did not appear in the first experiment, and we will try to explain them.

The intensity of the HH generated from 1.8 mJ is the highest, as shown in Figure 41, while the intensities of the HH are lower when using 2.16 mJ pulse energy. This may be agree with the results of the first experiment, however, the third used pulse energy 1.44 mJ, which is the lowest value of the driving pulse energy yields HH intensities between the other two, which is a new observation in this experiment. The paper that was published with the used MCP indicates that the intensity of the resulting HH are not necessarily a correct representation for the actual values the HH intensities. The latter paper presents and discusses several factors that contribute in the interpretation of the MCP to the generated HH. More details on this point can be found in the latter mentioned reference. However, the cut-off of the highest used driving pulse energy is consistent with the expected results, as Figure 41 shows, for there expected to be higher cut-off energy of the spectrum with using higher driving pulse energy. Note, that the 33^{rd} HH in the latter mentioned figure for the orange and the blue curves are not observed, so the peaks in this region and beyond are noise. The 33^{rd} HH in the green curve—namely the one resulting from the highest used pulse energy—is observed, and the observed peak of the green curve at the 35^{th} order can be a HH as well.

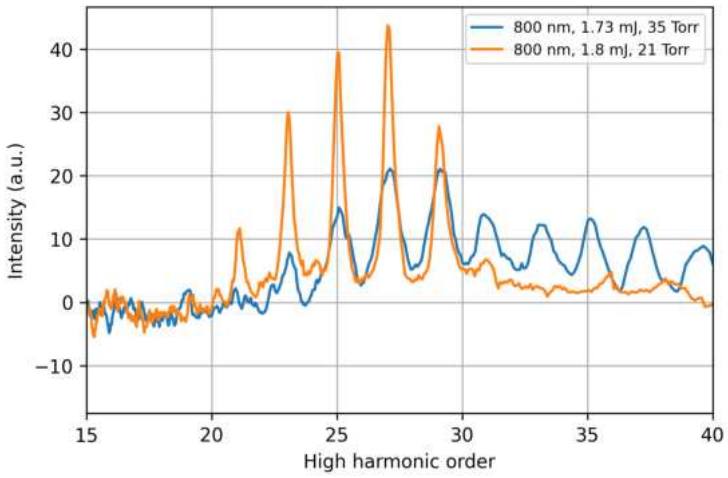


Figure 42: Spectra of the HH generated using three different driving pulse energy values: 1.8 mJ, and 1.73 mJ. from the first and the second experiment. Note that each spectrum has its own calibration, as aforementioned.

The number of the HH seem to be constant across the 10 different gas pressure values, as the phase matching maps in Figure 40 show. With the first used driving pulse energy 1.4 mJ, the HHs are four orders across the 10 pressure values except for the gas pressure value 19 Torr. We notice that the HHs of the first experiment are broader than the HHs of the second experiment. This can be observed in Figure 42, as well as the difference in the broadness of the HH lines can be seen clearly in Figures 27 and Figure 38. The broadness of the of the resulting HH lines can tell that band of the NIR pulses used in the first experiment were broader—shorter pulse duration—than the pulses of second experiment; consequently, the pulses produce a broader HH in the frequency domain. The relative band narrowness of the pulse in the frequency domain can be the reason as well to the decreased orders of the observed HH with respect to the first experiment.

With respect to the peak shifting, one notices in Figure 42, that although the spectra are from different experiments, but the driving pulse energies roughly the same, the peaks are well aligned. However the peaks of the different spectra of the second NIR experiment in Figure 41 show a slighter shifting when increasing the pulse energy than the different spectra from the first experiment shown in figure 37.

6.4 High Harmonic Generation Using 400 nm Source

In this part of the experiment, the SHG was used to convert the 800 nm driving beam into 400 nm using a BBO crystal. The measurements of the second experiment—HHG from 800 nm and 400 nm—were conducted during the same time period. Therefore, the setup was not changed. The parameters of the pressure in the vacuum system and the HHG chamber remained the same. The voltages applied on the MCP and the phosphor screen remained without changing as well. The alignment of the beam was not changed. The same XUV spectrometer was used. The lens that was used for focusing the 400 nm beam is also the same, 500 mm focal length. The only components that were changed are the mirrors shown in Figure 20, which were replaced mirrors reflective for the 400 nm wavelengths range and transparent for the 800 nm range. Additionally, placing a BBO crystal before the first 400 nm range reflective mirror, as shown in the latter Figure. Next, we tried to observe the harmonics by regulating the gas pressure in the HHG cell, the energies of the driving laser pulse, and by optimising the pulse energies by regulating the GDD and the TOD.

In this part of the experiment, 75 measurements were taken, each with different parameters of the pulse (pulse energy, GDD, and TOD), and the gas pressure. Across all of the measurements, ten shots for each result observed by the CCD camera were taken in order to take their average later, as shown in the following figures:

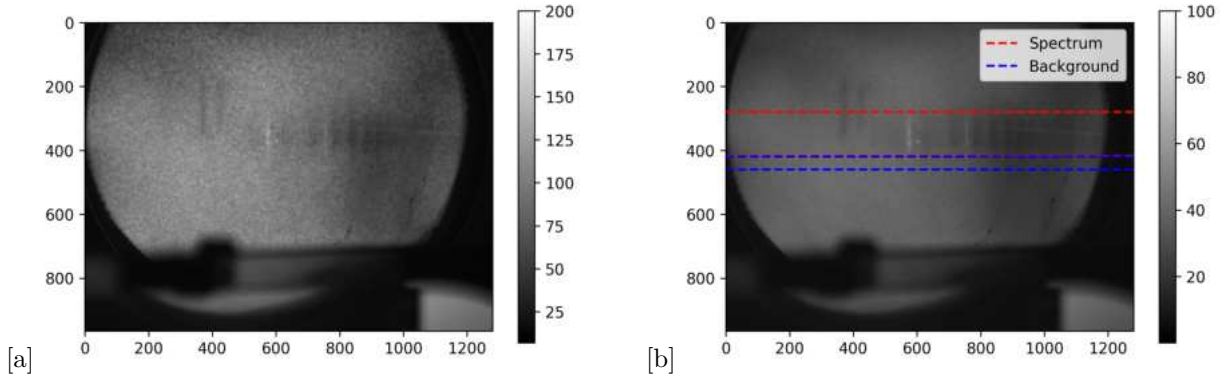
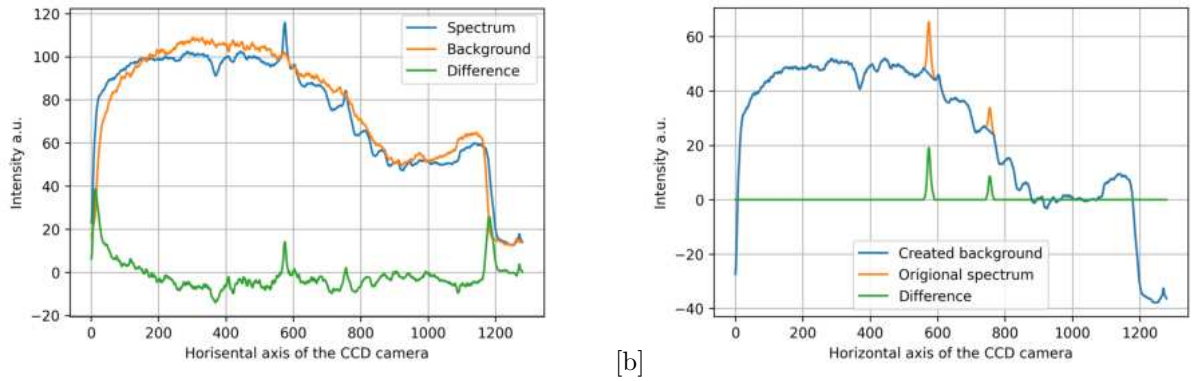


Figure 43: HHG signal using a 400 nm driving beam collected directly from the detector's screen by the CCD camera. (a) a single shot of the harmonics signal. (b) The average of ten images for a single measurement. These high harmonics are generated using 400 nm driving laser beam with 0.46 mJ pulse energy and 10 Torr gas pressure in the gas cell.

In Figure 43, the regions between the red and the blue dashed lines are the region where the mean values of the spectrum and the background are determined and shown in Figure 44. As shown in both figures that there are undesired dark and bright vertical regions both in the spectrum region and the background region. This would hinder extracting the harmonic data if the same recent method of extracting the HHG data using the NIR was used. More explanation on this point and how to overcome this obstacle can be found in the next Figure 44.

This spectrum in Figure 44 is the most intense in the this part of the experiment. The curves in Figure 44[a] show the mean values of the spectrum, the background, and the difference between them of a single measurement. These mean values are determined in the specified regions between the red dashed lines and the blue dashed lines of Figure 43. This range of pixels along the vertical axis is similar to the range of the NIR measurement, as shown in Figure 38, to preserve consistency in the interpretation of the data. Unlike the latter reference, the background of this spectrum is inhomogeneous, which hinders reading the heights of the generated HH, as the green line of Figure 44[a] shows. Therefore, another background is reconstructed. This starts by picking the most intense spectrum in this experiment and interpolate it in the following



[a] [b]

Figure 44: The curve in Figure [a] shows the harmonics generated using the 400 nm beam, 0.46 mJ pulse energy, and 10 Torr gas pressure. [b] is the exact same curve as [a], but interpolated at the region of the peaks.

way: the areas corresponding to the regions of the HH peaks along the x-axis are determined. Then an interpolation function is used to create a straight line between the two y-values (at the start and the end of each peak) as a function of the chosen x-values, where the peaks are. Then a straight line between these two values is drawn, resulting in a tilted straight line as shown in the blue line of Figure 44[b]. The mean values of the region 1020 to 1080 along the horizontal axis is subtracted for all the spectra. The result—i.e., the blue line of Figure 44[b], which also underlies the blue line—is averaged and used for all the other spectra by multiplying it by the scaling factor. This is due to the fact that different measurements of this experiments have different intensity profiles of the spectrum, and consequently, different backgrounds. For more details, see Appendix C for more details.

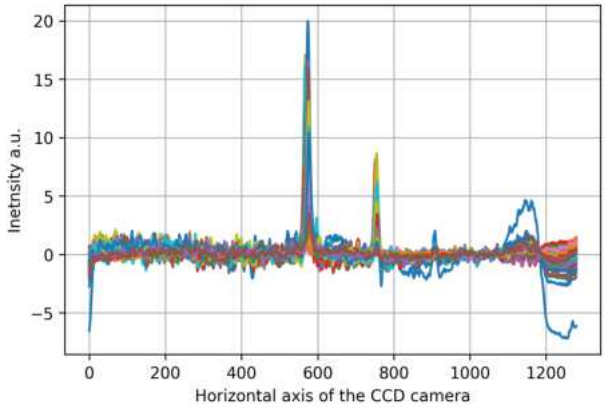
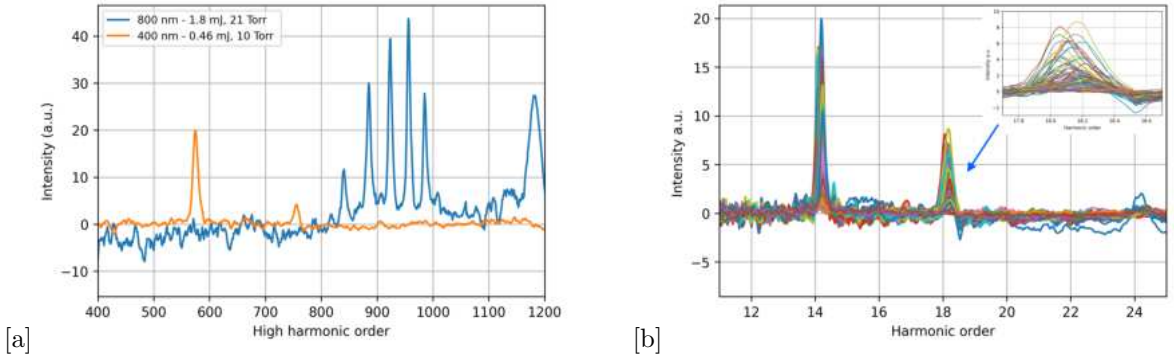


Figure 45: The resulting spectra after subtracting each constructed background–using interpolation in Figure 44[b]–from its corresponding spectrum. The peak intensities are collected form 75 spectra. The heat map in Figure 47 are made according to these peaks.

In addition to examining the effects of gas pressure and the driving laser pulse energies—the previously considered parameters—the GDD and the third order dispersion TOD were incorporated to optimise the driving pulse energies used with 400 beam.

To determine the orders of these harmonics yield of the 400 nm, the peaks of the most intense spectrum resulting from the 400 nm are integrated with the former discussed calibration procedure in subsection 6.1, since it is the same experiment and the same alignment, as aforementioned. Consequently, the same unknown parameters (u0 and A) of Equation (23) are used. The peak positions are: $u_m = (574, 755)$. The calibration result is shown in the following figure:



[a] [b]

Figure 46: [a] shows the resulting HH from the 800 nm and the 400 nm of the second experiment. The figure also shows where do the peaks of the harmonics using 400 nm lie with respect to the pixels of the image. Figure [b] shows all the spectra resulting from the 400 nm beam plotted versus the orders obtained from the second NIR calibration procedure.

The input beam was mainly 800 nm, that yields odd harmonics. By definition, when obtaining HH using 800 nm beam, the carrier wavelength of the fundamental 800 nm is divided by each of the resulting HH wavelength to obtain the HH orders $N = 800\,nm/\lambda_{(m)}$. The same is done for the harmonics wavelengths resulted from the 400 nm, given that the calibration is already done and the unknown parameters of Equation (23) are determined through curve fitting. Figure 48[a] demonstrates that the harmonics appear with orders 18^{th} and 22^{nd}, for a calibration that was done mainly for the 800 nm. Then to determine the orders of these harmonics in the 'language' of the 400 nm, the 400 nm must be divided by the harmonics wavelengths, $N = \lambda_{(fund)} = 400\,nm/\lambda_{(m)}$. This means that the obtained harmonics orders are 7^{th} and 9^{th}.

6.5 Heat Maps of the 400 nm High Harmonics Yield

The following phase matching maps incorporate two additional parameters to the parameters previously used, 5 different values of the GDD parameter and only three values for the TOD parameter incorporated in the measurements to optimise the driving pulse energy. The phase matching maps present the effects on the HH peaks intensities of five different driving laser pulse energies ≈(0.7, 0.56, 0.48, 0.41, and 0.35 mJ). Each value of these energies is investigated with five different GDD values of the driving pulse: (100, -100, -300, -500, and -700 fs^2) respectively. Each pair of "driving pulse energy—GDD value" is examined with five different gas pressure values in the HHG chamber. The entire group of the GDD and the driving pulse energy values are repeated three rounds, each round with a specific TOD value of the driving pulse. The examined values of the TOD are (0, 10000, and 20000 fs^{-3}). In total, 75 measurements were done. Each measurement with its own parameters, as shown in the following table:

The data of the following heat map is collected from the spectra shown in Figure 46 after doing the interpolation for each of them. The heat map is arranged exactly as the order of the table above. Each cell of energy in the table corresponds to a single block, (2×5) peaks, of peak intensities.

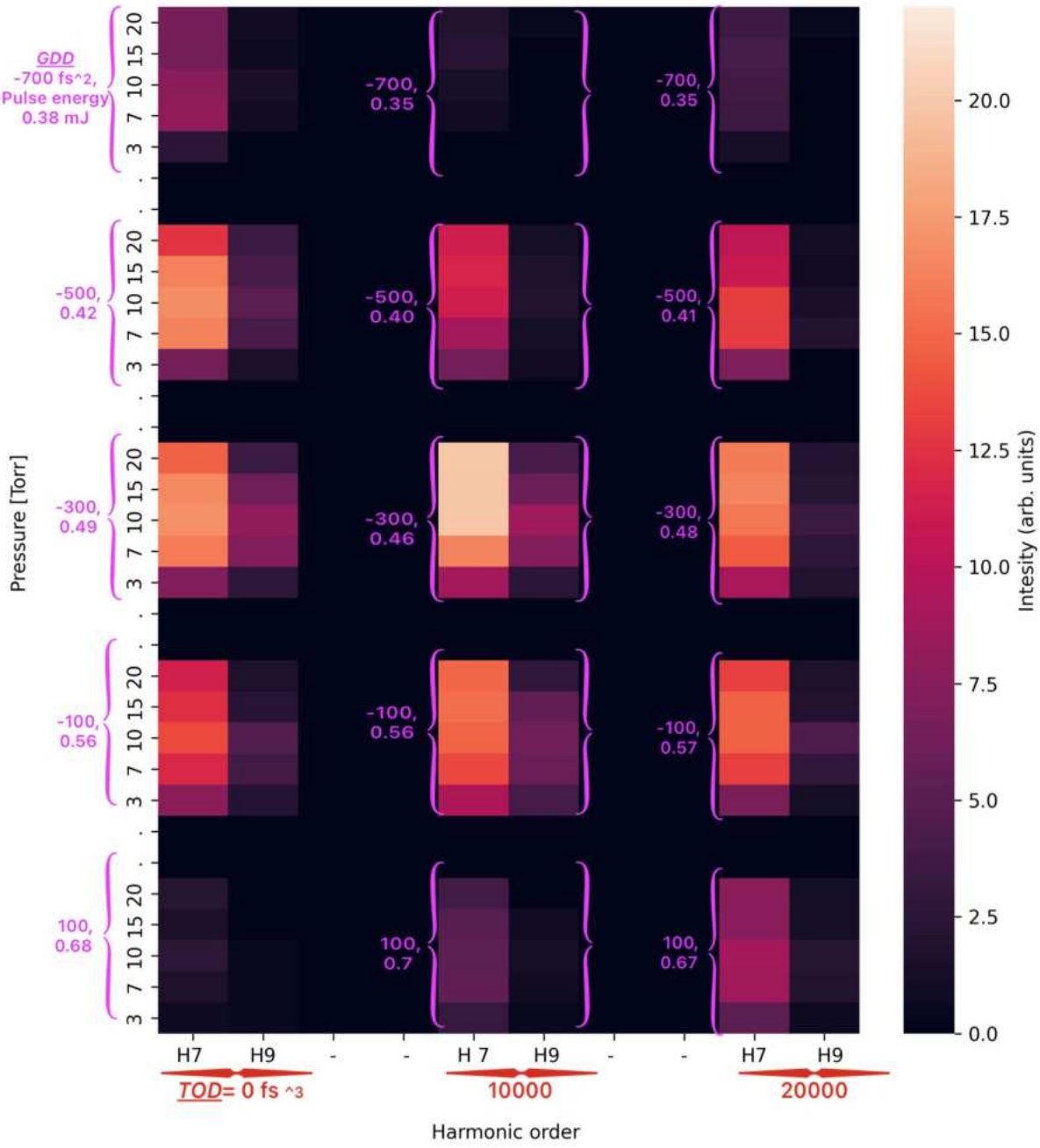


Figure 47: Heat maps of the HHG using 400 nm, 25 fs driving laser pulse. The phase matching maps show the influence of four different parameters on the intensities of two high harmonics, in the best cases, as shown in the figure. The parameters incorporated in the phase matching maps are: pulse energy of the driving laser, gas pressure values in the gas cell, various GDD, various values of TOD. The GDD and TOD are applied on the driving laser pulse.

-700 fs^2	0.38 mJ	0.35 mJ	0.35 mJ
-500 fs^2	0.42 mJ	0.40 mJ	0.41 mJ
-300 fs^2	0.49 mJ	0.46 mJ	0.48 mJ
-100 fs^2	0.56 mJ	0.56 mJ	0.57 mJ
GDD=100 fs^2	0.68 mJ	0.7 mJ	0.67 mJ
-	TOD = 0 fs^3	10000 fs^3	20000 fs^3

Table 2: This table shows the parameters that were used in the measurements of the HH using the 400 nm beam. With each driving pulse energy value that corresponds to a specific GDD and TOD, 5 different gas pressure values were investigated ranging from 3 to 20 Torr.

6.6 Discussion

The obtained spectrum and the background did not seem to be homogeneous, and the phosphor screen was affected by undesired inhomogeneous dark and bright regions, which made us extract the data using interpolation. The reason behind this inhomogeneity can be due to the diffraction of lower order harmonics that are not observable on the phosphor screen due to the grating limits; instead the lower order harmonics spread from the grating on the MCP generating electrons on different micro channels; consequently leading to inhomogeneous regions, as shown in Figure 43[b].

The number of harmonics resulting from the 400 nm wavelength, in comparison to the 800 nm, is significantly decreased. This can be due to the higher frequency of the driving field. In the case o 400 nm field, the field may not considered as static with respect to the atom. Consequently, the three-step model may not necessarily hold. With this high speed oscillation of the field, the potential of the atom may not adapt to the rapid changes. The calculation done in the first step in Figure 10 does not include the adaptation of the atom's potential to quick changes of the filed. Therefore, the effective phases that was indicated in the latter figure do not necessarily hold, or in the best cases there are two effective phases—for we observe only two harmonics–in the laser cycle from which the electrons are born and return back to re-collide with parent ion. Furthermore, the high speed oscillation of the field can make the motion of the electrons governed by the average of many cycles of the laser field. Consequently, this can make the electrons oscillate between the ground and the excited states without ionisation. Multiphoton ionisation and perturbation processes my play a major role here as well.

In the results of this experiment, as the phase matching maps show 47, one notices that achieving the phase matching conditions for the HH using the 400 nm, is done by applying lower pressure to conversion medium with respect to the recent HHG using NIR experiments. This indicates that there was a larger amount of plasma generated, which requires lower 'negativity' for the plasma phase mismatch term in order to balance the other terms, as mentioned in the section 3.

As shown in Figure 46, the decreased peak intensities of the HH resulting from the 400 nm to roughly half of the value with respect to the recent measurement of the HHG using the second 800 nm source that is shown in Figure 48[b]. This tells there are less photon emission in the conversion medium; consequently, less atoms contributing in the emission process.

Both the previous experiments using the NIR have demonstrated that using higher energies, a shift toward higher frequencies will occur, which can be due to the self phase modulation and the pulse broadening. The subfigure in Figure 46[b] shows that there is a slight shifting in the peaks of the 9^{th} harmonic with different measurements of different parameters. The 7^{th} harmonic also shows a pronounced shift. However, it is not strictly clear that the shifting here with the 400 nm is merely due to the variation of the pulse energies, given that there are two additional parameters: the GDD and the TOD. In the subfigure of Figure of 46[b],

the curve of the harmonics seems as if the peaks of the 9^{th} harmonics are affected in two degrees of freedom: the intensity of the peaks and their positions.

In the frame work of the pulse energies only, there can be a slighter shifting in the harmonic peaks positions of the 400 nm with respect to the HH from the NIR, for the steps between the various pulse energies are less.

Figure 48[b] shows a significant drop in the emitted photon energy using the 400 nm beam with respect to the other wavelengths of 800 nm. This agrees with the theoretical prediction of the three-step model: for the shorter wavelengths, the trajectories are relatively shorter; consequently, the electrons do not spend more time for more acceleration during the propagation. This makes the electron acquire less kinetic energy during the propagation and release less kinetic energy at the moment of re-collision. This yields a smaller cut-off photon energy with respect to the NIR HH yield. The results of the 400 nm shown in this experiment agree with the results of the work [15]. In both cases of HHG, the argon gas is used and two harmonics only are generated. In the comparison work, 26 fs pulse duration, 1 mJ driving pulse energy is used for the HHG. In the comparison work, the cut-off is $\approx$ 34 eV, and the cut-off energy of this experiment is $\approx$ 28.1 eV, as shown in Figures 48[b] and [c].

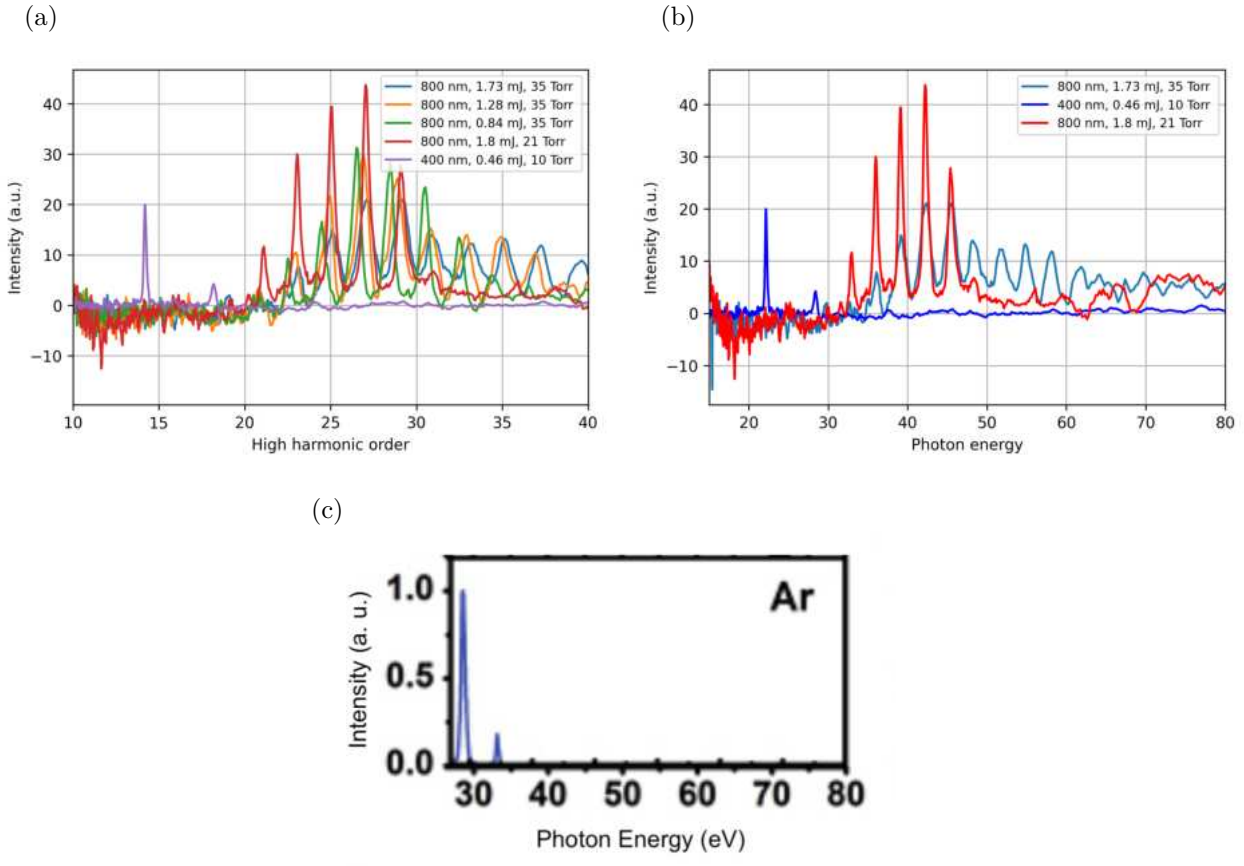


Figure 48: [a] Different HH spectra from the recent different measurements assigned to their orders. [b] HH Spectra as a function of photon energies from the first and the second experiment using NIR and roughly similar energy pulse values compared to the spectrum obtained from the 400 nm source. [c] HHG using driving laser pulse with 400 nm carrier wavelength. This figure is adapted from [15].

The HH generated from the 800 nm in the first experiment show significantly wider flexibility with respect to the gas pressure conditions at which HH are obtained with respect to the harmonics obtained from the 400 nm source. HH in gas pressures ranging roughly from 20 to 65 Torr can still be obtained. While in the case of the 400 nm, HH are observable only roughly in the range of 3 to 20 Torr. This can be observed in the phase matching maps in Figures 33 and 47. This can tell the three-step model and its predictions in the case of 800 nm with respect to the various long and short phases still valid under different gas pressure, while it is not the case in the harmonics generated from the 400nm.

Summery

This thesis demonstrated in two experiments the optimum phase matching conditions for both the 800 nm and the 400 nm sources. In the first experiment, for the purpose of investigating the phase matching, we used an available setup. In the second experiment, we used a new setup to compare HHG from 800 nm and 400 nm sources. Through curves, phase matching maps, and discussions, we demonstrated the behavior of different spectra using various values of the parameters: gas pressure, driving pulse energy in the first experiment, and in the second experiment, we integrated two additional parameters in the investigation, namely the GDD and the TOD. We tried to explain the potential reasons underlying the observed behaviors. The 400 nm light demonstrates less photon energy and lower required pressure conditions compared to the 800 nm source.

In conclusion, the 800 nm source provides higher photon energies in HHG. The parameter used in the 800 nm experiments gives more flexible degrees of freedom for achieving HHG compared to the 400 nm source. The theoretical predictions of the three-step model are clearer in the case of HHG using 800 nm, and mostly related to widely investigated parameters. A static field that plays a major role in the three-step model does not seem to do so in the case of HHG from 400 nm. Electrons do not seem to release significant energies at re-collision compared to the HHG from 800 nm.

Appendix A: Row Data and Extended Curve-Fitting

The following are the raw data that are directly collected of the MCP-phosphor detector's screen using 1.28 mJ driving pulse energy. This image shows roughly the phase matched case of high harmonic generation.

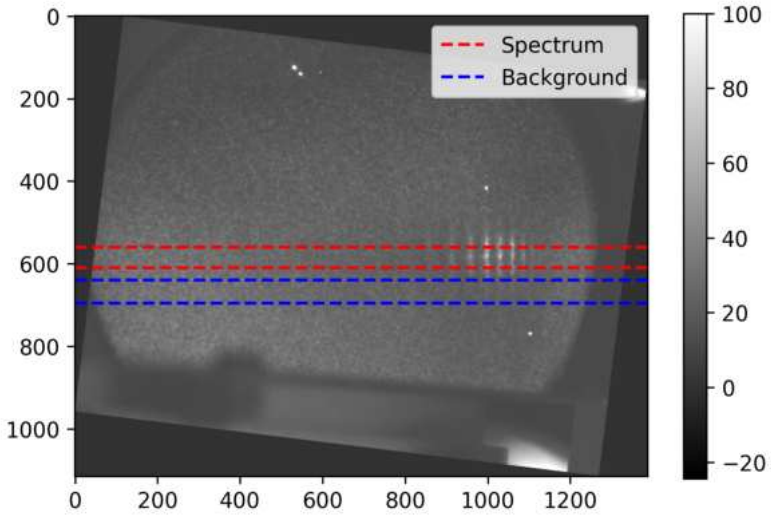


Figure 49: High harmonics directly captured via a CCD camera from the detector's screen. With these HH an extended calibration procedure is performed. This image is rotated as well by -7°. The HH are obtained using 0.84 mJ pulse energy at 35 Torr gas pressure. The area between red dashed lines is the area where the mean values of the spectrum are obtained. The area between the blue dashed lines is the area where the mean values of the background are obtained. These mean values are shown in Figure 52 in the blue and the Orange lines.

The following is the extended calibration for harmonics order for the driving pulse energy (1.28 mJ) at the gas pressure (35 Torr) at which the phase matching occurs.

In order to self run the calibration procedure, the peaks positions of the HH are $u_m = (914, 959, 998, 1031, 1061, 1086, 1108)$. The unknown values of Equation (23) (u_0 and A) that are obtained through the curve fitting are (701.54068202, -1254.99749996) consequently. These values are valid only for the first attempt with its own setup to determine the phase matching conditions.

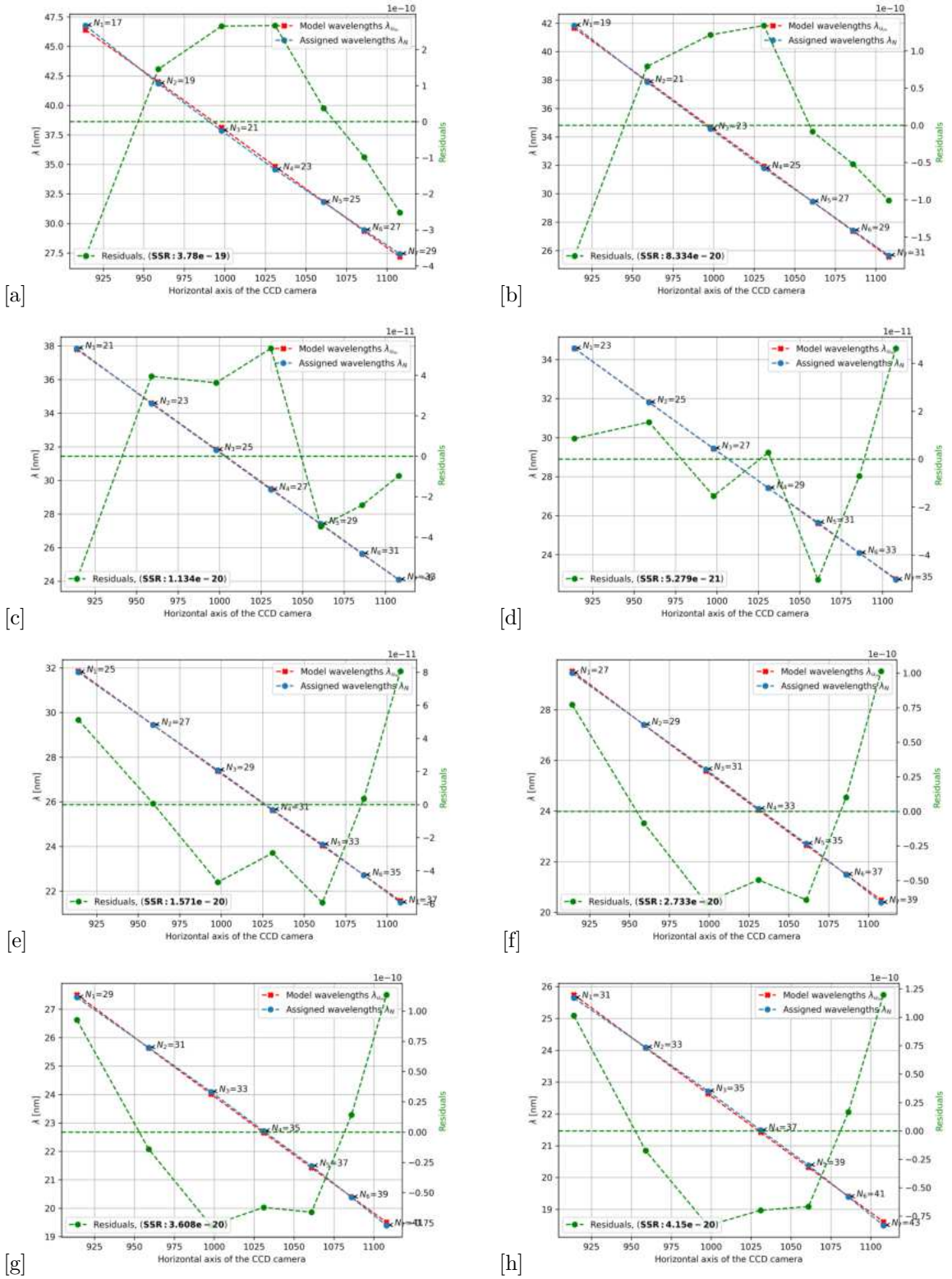


Figure 50: Extended curve fitting calculations for harmonics spectrum generated from (0.84 mJ, 35 Torr). The (blue and red dots) are the theoretical and experimental harmonics for different values of N to determine the orders. The green dots are the corresponding residual for each blue-red dots pair. The residuals are measured with respect to the dashed zero line. The SSR shows that the plot [d] has the best cure-fitting. There for the correct chain of harmonics starts at N=21. Note: there is no continuity of data between each two points.

The harmonics spectrum after running the calibration will be:

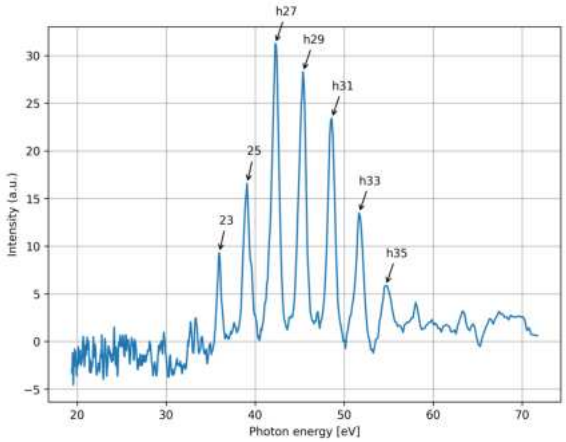


Figure 51: The high harmonic spectrum versus the photon energy values. The HH generated using the driving pulse energy 0.84 mJ under 35 Torr gas pressure.

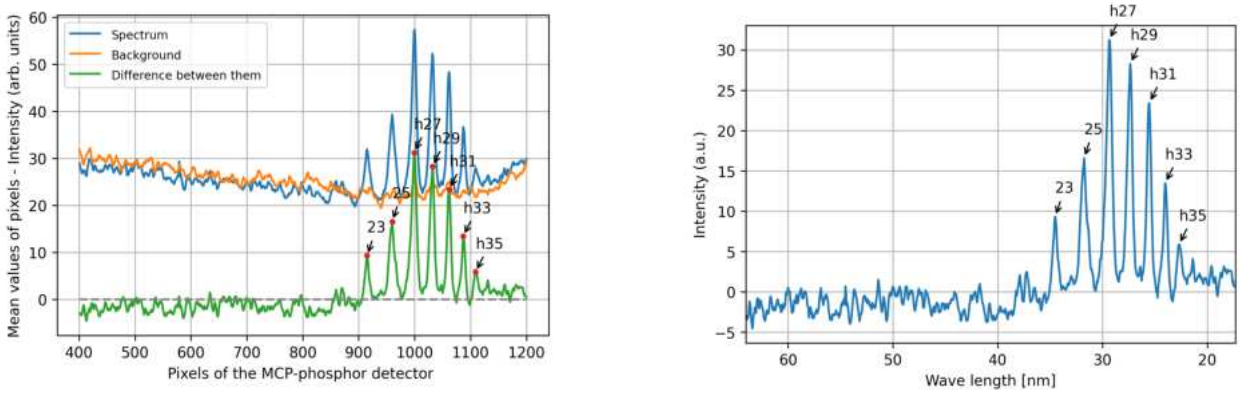


Figure 52: The harmonics spectrum with their wavelengths and with respect to the position of the detector's screen after determining their orders. This spectrum is generated using the driving pulse energy 0.84 mJ at the gas pressure 35 Torr.

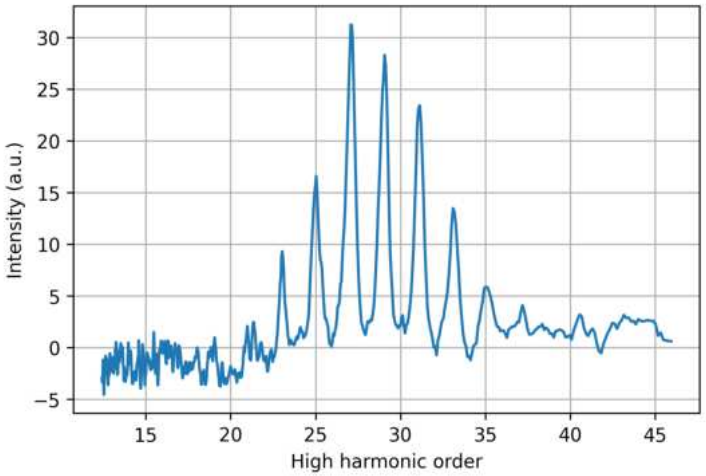


Figure 53: The spectrum of HH versus their orders. The values of the orders are obtained from Equation 23 after performing the curve fitting. The spectrum generated using the driving pulse energy 1.28 mJ under 35 Torr gas pressure. This figure serves as an extended calibration procedure for the one done using the driving pulse energy 1.73 mJ at 35 Torr in Figure 32 in order to show the consistency of the calibration. The calibration procedure is done in the subsection 5.3.

Appendix B: Data of the Measurements

The data that were obtained from the two experiments can be found in the following links. Theses data form the basis on which this thesis is carried out.

The data of the first experiment, HHG using an NIR driving laser field in an already available setup, can be found in the following link: HHG Using NIR^{1st}. The images are named according to the following format: 'Abd_220819_power[**value of the power**]_press[**value of the pressure**].bmp'. The filename alone provides all the needed information to know the parameters under which the measurement was conducted. The parameters in this experiment are the driving pulse energy and the gas pressure at the HHG cell.

For the second experiment, the first measurement, ten shots per measurement to calculate their average later. The HHG images are named according to the format: 'camXUVDET_[**yyyymmdd**]_[**hhmmss**]'. We made sure that each measurement—ten shots—was taken in the *same minute* to ease the classification later. Note, that some measurements has ± 10 shots per minute. The table shown with the measurements uses the time at which the measurements were taken. The data of this part of the second experiment can be found in the link: HHG Using NIR^{2nd}-Unaveraged Data

A The data where averaged using a code, see Appendix C, that averages the images according to the minute mentioned in their name. The code works by providing the input folder (where the unaveraged images are) and the out put folder. The shots taken in the same minutes will be compiled in one averaged image in the output folder named in the format: 'average_[**hhmm**]'. The averaged data of this measurement then can be found in the link: HHG Using NIR^{2nd}-Averaged Data. This folder also contains the table that lists the names of the images along with their corresponding measurement parameters.

The data of second measurement at the second experiment, HHG using 400 nm, were subject to the same mentioned image processing steps of the first measurement regarding the averaging and format name of the files. The data of the individual shots without averaging of this part can be found in the link: HHG Using 400 nm-Individual shot Data.

The averaged data of this part can be found in the link: HHG Using 400 nm-Averaged Data.

Appendix C: Calculation Code

The code will be in the link below. The code is used to plot the figures throughout this thesis and will be ordered according to the sequence of the figures. The title of each figure is written on the top of each cell.

Please follow the interactions of the code carefully. Note that some parts of the code are dependent on the details presented in Appendix B.

HHG code of 800 nm and 400 nm.

Appendix D: Calculation Code

The code in this Appendix is used to plot the figures throughout this thesis. It will be ordered according to the sequence of the figures. The title of each figure is written on the top of each cell. If you have any questions regarding the details of the code, please contact me through the email (abduljaleel[@]physik.hu-berlin.de).

Listing 1: The following is written in python, it is used to reproduce and simulate the three-step model and the needed calculations of the phase matching.

```
# ---------------- Time scale figure ----------------

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

fig, ax = plt.subplots(figsize=(15, 1.8), dpi=300)

start_time = 1e-18
end_time = 1e18

num_points = 1000
time_values = np.logspace(np.log10(start_time), np.log10(end_time),
                          num=num_points)

comment_times = [1e-18, 1e-14, 1e-9, 1e-3, 60, 2.4e6, 1.4e13, 1e17]
comment_labels = [
    'S-P␣states␣electron\nmotion␣in␣H␣atom', 'fs␣laser\npulse',
    'State-of-the-art␣CPU\nincrement', 'Camera␣flash\nduration',
    '1␣minute', 'Moon␣orbit\ncycle', 'First␣human\nrecorded␣existence',
    'Age␣of\nuniverse'
]

cmap_colors = plt.cm.viridis(np.linspace(0, 1, num_points))
custom_cmap = LinearSegmentedColormap.from_list("CustomCmap", cmap_colors)

for t, color in zip(time_values, cmap_colors):
    ax.plot([t, t], [0, 1], color=color, linewidth=5)

for t, label in zip(comment_times, comment_labels):
    ax.text(t, 1.15, label, ha='center', va='bottom', fontsize=12,
            fontweight='bold', color='black')

ax.set_xscale('log')
ax.set_xlim(start_time, end_time)
ax.set_ylim(0, 1.5)
ax.set_yticks([])
ax.set_xlabel('Time␣(seconds)', fontweight='bold')

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_linewidth(2)
ax.spines['left'].set_visible(False)
plt.tight_layout()
plt.show()

# ---------------- Qualitative Coulomb potential ----------------

import matplotlib.pyplot as plt
from scipy.constants import electron_volt as eV
import numpy as np
```

```
Z = 1
e = 1.602176634e-19
eps0 = 8.854187817e-12
f0 = 3.75e14
elc = 1.602e-19
f0 = 3.75e14
m_e = 9.109e-31
w0 = f0 * 2 * np.pi

phi_array = np.linspace(0, np.pi / 2, num=100, endpoint=False)
phi = 0
I = 2e18
C = 299792458
eps0 = 8.85e-12
r_ind = 1.0
E0 = np.sqrt((2 * I) / (C * eps0 * r_ind))
t = 2e-15

Modification_amount = 1

r = np.linspace(0.5e-12, 1e-11, 1000)

V = - (Z * e ** 2) / (4 * np.pi * eps0 * r)  # Potential equation

E_field = E0 *(np.cos((w0*t) + phi)*Modification_amount) #Modifying Laser Field
V_mod = V + r * E_field
V_mod2 = V + r * -E_field

# Plotting
plt.figure(figsize=(10, 6), dpi=300)

r_angst = r * 1e10
plt.plot(r_angst, V / eV, "--", label='Coulomb Potential', color='b')
plt.plot(-r_angst, V / eV, "--", color='b')

t_fs = t * 1e15

plt.plot(-r_angst, V_mod / eV,
         label=f'Modified Coulomb Potential at t={t_fs} fs', color='r')
plt.plot(r_angst, V_mod2 / eV, color='r')
plt.axhline(y=-1500, xmin=0, xmax=10, color='black', linestyle='-',
            label='Ionisation potential')

plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['bottom'].set_visible(True)
plt.gca().spines['left'].set_visible(True)

# To omit the ticks on the x and y axis
plt.xticks([], [])
plt.yticks([], [])

plt.xlabel('Distance from nucleus (  )')
plt.ylabel('Potential (eV)')
plt.title('Coulomb Potential Modified by Laser Field')
plt.legend()
plt.show()
```

```
# ----------------- The Gaussian and ADK plot ---------------------

import scipy.optimize as sop
import numpy as np
import matplotlib.pyplot as plt
import math

# Constants and Parameters
n = 1               # Principal quantum number
l = 1               # Orbital quantum number
m = 0               # Magnetic quantum number
r_ind = 1.          # Refractive index
z = 1               # Degree of ionisation, charge
C = 299792458       # Speed of light in m/s
l2 = 0.8            # Effective orbital quantum number (ADK)
n2 = 0.93           # Effective principal quantum number (ADK)
Ei = (z**2 / (2 * n2**2))  # Binding energy of argon (in a.u.)

eps0 = 8.85e-12     # Permittivity of vacuum
m_e = 9.109e-31     # Electron rest mass
elc = 1.602e-19     # Elementary charge

I = 2e18            # Laser intensity in W/m^2
phi1 = 0            # Phase
Eh_eV = 27.211      # Hartree energy in eV
a0 = 0.528e-10      # Bohr radius
Eh_SI = 27.211 * elc  # Hartree energy in Joules
E_in_au = Eh_SI / elc / a0  # Electric field in atomic units

# Time definitions
t = np.linspace(-50e-15, 50e-15, 1000, endpoint=False)
T = np.linspace(0, 4e-15, 1000)
T2 = np.linspace(0, 2e-14, 1000)

t_fwhm = 2.5e-14   # FWHM of the pulse period
f0 = 3.75e14       # Central frequency of driving laser
w0 = f0 * 2 * np.pi
sig = t_fwhm / (2 * np.sqrt(2 * np.log(2)))  # Width, standard deviation
I = 2e18           # Laser intensity
E0 = np.sqrt((2 * I) / (C * eps0 * r_ind))  # Peak electric field strength

# Gaussian Wave Packet Function
def Gauss_WP(t, sig, t_fwhm, w0, phi):
    term1 = np.exp(-t**2 / (2 * (sig**2)))
    term2 = np.cos((w0 * t) + phi1)
    return term1 * term2 * E0 / E_in_au

E = Gauss_WP(t, sig, t_fwhm, w0, phi1)  # Gaussian electric field in a.u.

# ADK Ionization Rate Function

# ADK Ionization Rate Function
def ionz_rate(E):
    term1 = np.sqrt(3 / (np.pi**3))
    term2 = (((2 * l) + 1) * math.factorial(l + abs(m))) / \
            (math.factorial(abs(m)) * math.factorial(l - abs(m)))
    term3 = (elc / (((n2**2) - (l2**2))**0.5))**(abs(m) + 0.5)
    term4 = ((n2 + l2) / (n2 - l2))**l2
    term5 = (z**2 / n2**3) * ((4 * np.e * z**3) / (E * (n2**3) * \
            (((n2**2) - (l2**2))**0.5)))**((2 * n2 - abs(m) - 1.5))
    w1 = term1 * term2 * term3 * term4 * term5 * \
         np.exp(-(2 * z**3) / (3 * E * n2**4))
```

```
    return w1

w1 = ionz_rate(E)  # ADK ionization rate

# Plotting
plt.figure(figsize=(10, 6), dpi=300)
t_fs = t * 1e15
E_envelope = np.abs(E0 * np.exp(-t**2 / (2 * (sig ** 2))) / E_in_au)

# ADK Ionization Probability Plot
ax1 = plt.gca()
ax1.plot(t_fs, w1, 'b-', linewidth=4, label='ADK ionization probability')
ax1.set_xlabel("Time (fs)")
ax1.set_ylabel("ADK ionization probability", color='b')
ax1.tick_params(axis='y', labelcolor='b')
ax1.legend(loc='upper left')

# Gaussian Wave Packet and Pulse Envelope Plot
ax2 = ax1.twinx()
c_freq = f0 / 1e12  # Convert to THz
ax2.plot(t_fs,E,'r-',label=f'G. wave P.,\nf0={c_freq}THz,\n$\lambda$ = 400 nm')
ax2.plot(t_fs, E_envelope, 'g--', label='Pulse envelope')  # Upper envelope
ax2.plot(t_fs, -E_envelope, 'g--')  # Lower envelope

# Annotating FWHM
half_max = max(E) / 2
fwhm_start = -12.5  # FWHM start
fwhm_end = 12.5     # FWHM end
ax2.axvline(fwhm_start, color='k', linestyle='--', label='FWHM')
ax2.axvline(fwhm_end, color='k', linestyle='--')
ax2.annotate('FWHM\n= 25 fs', xy=(0, half_max), xytext=(15, half_max),
             fontsize=12, color='black')

ax2.set_ylabel("Amplitude of Gaussian", color='r')
ax2.tick_params(axis='y', labelcolor='r')
ax2.legend(loc='upper right')

plt.show()


phi_array = np.linspace(0,np.pi/2,num=100, endpoint=False) # different phases


E0 = np.sqrt((2*I)/(C*eps0*r_ind))

def func1 (T2,phi,w):
    return np.cos((w*T2)+phi)-np.cos(phi)

def func2 (T2,phi,w):
    return w * -np.sin(phi)*T2

def Velocity(arrival_times,phi_array):
    return -((elc * E0) / (m_e * w0)) *\
        (np.sin((w0*arrival_times)+phi_array)-np.sin(phi_array))

def Pondro_E (elc,E0, m_e, w):
    return (elc**2 * E0**2)/(4*m_e*w**2)

Up = Pondro_E (elc,E0, m_e, w0) #in joules(c2(F/c)2)/(Kg*1/s2)= (m2 Kg/s2)


phi_array =np.linspace(0, np.pi/2, num=20,
```

```
                        endpoint=False)# More points for smoother curve

plt.figure(dpi=300)  # High resolution

T_fs = T * 1e15  # Convert time from seconds to femtoseconds

comment_phis = np.linspace(0, len(phi_array) - 1, num=10, dtype=int)

for pp, phi in enumerate(phi_array):
    plt.plot(T_fs, func1(T, phi, w0) - func2(T, phi, w0), label=str(phi))
    plt.plot(T_fs, 0 * T, color='k', linestyle='--')

    #  phi to degrees
    phi_degrees = (phi / np.pi) * 180

    # Add comments only for specific phi values
    if pp in comment_phis:
        x_comment = T_fs[-1] * 1.05
        y_comment = func1(T[-1], phi, w0) - func2(T[-1], phi, w0)

        # Check if phi equal to 18
        if phi_degrees == 18:
            comment = f'$\phi$: {phi_degrees:.2f}  ' #Format
            plt.text(x_comment, y_comment, comment, fontsize=10, color='red',
                     weight='bold', verticalalignment='center',
                     horizontalalignment='left')
        else:
            comment = f'$\phi$: {phi_degrees:.2f}  '
            plt.text(x_comment, y_comment, comment, fontsize=10,
                     verticalalignment='center', horizontalalignment='left')

plt.xlabel("Time (fs)")
plt.ylabel("Trajectories as a function of birth time")

plt.show()

phi_array = np.linspace(0*np.pi, 2*np.pi, num=100, endpoint=False)



 # counter for trajectories that cross zero once twice or more
cross_zero_count = 0
non_cross_zero_count = 0
multi_cross_count = 0

offset_factor = 0
ax1 = plt.gca()


for (pp, phi) in enumerate(phi_array):
    y_values = func1(T, phi, w0) - func2(T, phi, w0) + pp * offset_factor
    crosses_zero = [y > pp * offset_factor for y in y_values[1:]]

    cross_count = sum(1 for i in range(1, len(crosses_zero))\
                      if crosses_zero[i] != crosses_zero[i-1])

    phi_deg = np.degrees(phi)

    if cross_count >= 2:
        ax1.plot(t_fs, y_values, color='green')
        multi_cross_count += 1
    elif np.isclose(phi_deg, 18) or np.isclose(phi_deg, 198):
```

```
        ax1.text(0.05, pp * offset_factor, f"{phi_deg:.0f}  ", color='black')

        ax1.plot(t_fs, y_values, color='blue')
    elif cross_count == 1:
        ax1.plot(t_fs, y_values, color='red')
        cross_zero_count += 1
    else:
        ax1.plot(t_fs, y_values, color='black')
        non_cross_zero_count += 1

zero_line = np.linspace(0, (len(phi_array) - 1) * offset_factor, len(T))
ax1.plot(t_fs, zero_line, color='k', linestyle='--')



ax1.set_xlabel("Time (fs)")
ax1.set_ylabel("Trajectories as a function of birth time")

from matplotlib.lines import Line2D


# To arrange the legend
custom_lines = [Line2D([0], [0], color='black' , lw=2),
                Line2D([0], [0], color='red'   , lw=2),
                Line2D([0], [0], color='green' , lw=2),
                Line2D([0], [0], color='blue'  , lw=2)]

plt.legend(custom_lines, [f'Do not cross zero ({non_cross_zero_count})',
                          f'Cross zero once ({cross_zero_count})',
                          f'Multi-cross ({multi_cross_count})',
                          'Cut-off (2)'], loc='upper right')

phi_ticks = np.linspace(0, (len(phi_array) - 1) * offset_factor, 20)
phi_tick_labels = np.degrees(np.linspace(0, 2 * np.pi, num=20, endpoint=False))

ax1.set_yticks(phi_ticks)
ax1.set_yticklabels([f"{phi:.0f}" for phi in phi_tick_labels])
ax1.set_ylabel("Trajectories approx. $\phi^ $ vals. with offest")

plt.show()



# Adding the offeset to the curves

phi_array = np.linspace(0*np.pi, 2*np.pi, num=100, endpoint=False)

cross_zero_count = 0
non_cross_zero_count = 0
multi_cross_count = 0

offset_factor = 0.6
ax1 = plt.gca()


for (pp, phi) in enumerate(phi_array):
    y_values = func1(T, phi, w0) - func2(T, phi, w0) + pp * offset_factor
    crosses_zero = [y > pp * offset_factor for y in y_values[1:]]

    cross_count = sum(1 for i in range(1, len(crosses_zero))\
                      if crosses_zero[i] != crosses_zero[i-1])
```

```
    phi_deg = np.degrees(phi)

    if cross_count >= 2:
        ax1.plot(t_fs, y_values, color='green')
        multi_cross_count += 1
    elif np.isclose(phi_deg, 18) or np.isclose(phi_deg, 198):
        ax1.text(0.05, pp * offset_factor, f"{phi_deg:.0f}  ", color='black')

        ax1.plot(t_fs, y_values, color='blue')
    elif cross_count == 1:
        ax1.plot(t_fs, y_values, color='red')
        cross_zero_count += 1
    else:
        ax1.plot(t_fs, y_values, color='black')
        non_cross_zero_count += 1

zero_line = np.linspace(0, (len(phi_array) - 1) * offset_factor, len(T))
ax1.plot(t_fs, zero_line, color='k', linestyle='--')



ax1.set_xlabel("Time (fs)")
ax1.set_ylabel("Trajectories as a function of birth time")

from matplotlib.lines import Line2D


# To arrange the legend
custom_lines = [Line2D([0], [0], color='black' , lw=2),
                Line2D([0], [0], color='red'   , lw=2),
                Line2D([0], [0], color='green' , lw=2),
                Line2D([0], [0], color='blue'  , lw=2)]

plt.legend(custom_lines, [f'Do not cross zero ({non_cross_zero_count})',
                          f'Cross zero once ({cross_zero_count})',
                          f'Multi-cross ({multi_cross_count})',
                          'Cut-off (2)'], loc='upper right')




phi_ticks = np.linspace(0, (len(phi_array) - 1) * offset_factor, 20)
phi_tick_labels = np.degrees(np.linspace(0, 2 * np.pi, num=20, endpoint=False))

ax1.set_yticks(phi_ticks)
ax1.set_yticklabels([f"{phi:.0f}" for phi in phi_tick_labels])
ax1.set_ylabel("Trajectories approx. $\phi^ $ vals. with offest")

plt.show()


''''----------------Smooth cut off energy------------------------'''




plt.figure(dpi=300)

phi_array = np.linspace(0, np.pi/2,
                        num=1000,
```

```
                        endpoint=False)  # More points for smoother curve

T = np.linspace(0, 4e-15, 1000)  # More points for smoother curve

phi_array = np.linspace(0,np.pi/2,num=50, endpoint=False)

arrival_times = []

for (pp, phi) in enumerate(phi_array):
    value = func1(T,phi,w0)-func2(T,phi,w0)
    for (ii,val) in enumerate(value):
        if val > 0:

            break
    arrival_times.append(T[ii])

velocity_array = Velocity(np.array(arrival_times),phi_array)

kinetic_energy_array =  (1/2) * m_e *((velocity_array)**2)/Up


#  kinetic energy array with the smooth curve
kinetic_energy_array_smooth = (1/2) * m_e * ((Velocity(np.array(arrival_times),
                                                       phi_array))**2) / Up

plt.plot((phi_array/np.pi)*180,
         kinetic_energy_array_smooth)  #Plots K.E. vs phi
plt.xlabel("$\phi $")
plt.ylabel("E$_{k}$ / U$_p$ ratio")
plt.grid()



# Draw a vertical line at phi=17.999
vertical_line_angle = 17.999
plt.axvline(x=vertical_line_angle, color='red', linestyle='--',
            label=f'$\phi $ = {vertical_line_angle:.0f}  ')

horizontal_line_ratio = 3.17
plt.axhline(y=horizontal_line_ratio, color='blue', linestyle='--',
            label = f'$E_k$/$U_p$ = {horizontal_line_ratio:.2f}')

x_comment_before = (phi_array[0]/np.pi)*180
y_comment_before = horizontal_line_ratio * 0.9
comment_before = 'Long\ntrajectories\ncontribution'
plt.text(x_comment_before, y_comment_before, comment_before, fontsize=10,
         verticalalignment='center', horizontalalignment='left')

x_comment_after = (phi_array[-1]/np.pi)*60
y_comment_after = horizontal_line_ratio * 0.9
comment_after = 'Short\ntrajectories\ncontribution'
plt.text(x_comment_after, y_comment_after, comment_after, fontsize=10,
         verticalalignment='center', horizontalalignment='left')

plt.legend()
plt.show()

#-----------------------------------------------
#-----------------------------------------------


#%%%%
```

```
'------------------␣Qualitative␣few␣cycles␣gaussian␣pulse␣-----------------'


import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(-1, 1, 5000)


frequency = 9 / (max(t) - min(t))

# Gaussian envelope
gaussian_envelope = np.exp(-((t)**2) / (2 * (0.2)**2))

# Gaussian equation - term of oscillation
gaussian_pulse = np.cos(2 * np.pi * frequency * t) * gaussian_envelope

# Calculate the phase in radians for all points
phase_rad = 2 * np.pi * frequency * t

max_idx = np.argmax(gaussian_envelope)

# Convert 18 degrees to radians
target_phase_rad = np.deg2rad(18+180)
target_idx = np.argmin(np.abs(phase_rad - target_phase_rad))

target_phase_rad2 = np.deg2rad(18)
target_idx2 = np.argmin(np.abs(phase_rad - target_phase_rad2))




plt.figure(figsize=(10, 6),dpi=300)

plt.axis('off')
plt.gca().set_frame_on(False)

plt.plot(t, gaussian_pulse,
         linewidth=1)  # Increased linewidth for better visibility
plt.title("Gaussian␣Pulse␣with␣HHG␣Cutoff␣Phase␣at␣18  ")
plt.xlabel("Time")
plt.ylabel("Amplitude")
plt.axhline(0, color='gray', linestyle='--')  # zero line
plt.scatter(t[target_idx], gaussian_pulse[target_idx], color='red') # MarkPoint
plt.annotate('␣Cutoff␣phase␣=␣198  ␣', (t[target_idx],
                                      gaussian_pulse[target_idx]),
             textcoords="offset␣points", xytext=(19,-20), ha='center')

plt.annotate('Cutoff\n␣phase␣=␣18  ', (t[target_idx2],
                                     gaussian_pulse[target_idx2]),
             textcoords="offset␣points", xytext=(19,-20), ha='center')
plt.scatter(t[target_idx2], gaussian_pulse[target_idx2],
            color='red') # Mark point


plt.show()

#%%%

'-------Transition␣Dipole␣Moment␣vs␣Photon␣energy␣--------'
```

```
N_t      = 8500                     # Number of points in time.

Ip       = 15.76 / 27.2114          # Atom ionization potential for argon.


'''

I = 2e15 w/m^-2 = 0.516 E in au.
I = 2e16 W/m-^2 = 1.632
'''

E0       = 0.105    #For 800nm (The E0 is 0.105 a.u)

omega0   =   0.057 # In a.u.
A0           = E0 / omega0
T            = 2.0 * np.pi / omega0

#-------------------------------------------------
#-------------------------------------------------



def E(phir):                # Electric field in scaled coordinates.

        return np.cos(phir)

def A(phir):                # Vector potential in scaled coordinates.

        return - np.sin(phir)

def intA(phir):                 # Integrated vector potential in scaled coordinates.

        return np.cos(phir)


# phib : Phase of birth (ionization)
# phir : Phase of recombination
# psr : Momentum divided A0

def saddlePointEquations (x0, phi3r):

        eqs = np.zeros(2, dtype=float)

        psr, phi1r = x0

        Ab     = A(phi1r)
        #Ar     = A(phi3r)

        intAb = intA(phi1r)
        intAr = intA(phi3r)

        # Ionization equation
        eqs[0] = np.power(psr + Ab, 2.0)

        # Propagation equation

        eqs[1] = intAr - intAb + (phi3r - phi1r) * psr

        return eqs

def solveSystem (phi3r):
```

```
        N     = np.size(phi3r)
        phi1r = np.zeros(N)
        psr   = np.zeros(N)

# For all times in range, use least squares method to solve saddle-point system.
# Initial guess for first round is 0 and 0 for both psr and phi1r.

        for k in range(N):

                if (k == 0):
                        res = sop.least_squares (saddlePointEquations, \
                                                                x0=(0, 0), \
                                                                args=(phi3r[k],))
                else:
                        res = sop.least_squares (saddlePointEquations, \
                                                        x0=(psr[k - 1], phi1r[k - 1]), \
                                                        args=(phi3r[k],))

                psr[k]   = res.x[0]
                phi1r[k] = res.x[1]

        return phi1r, psr

def getSolutions (N_t, PHI_MIN, PHI_MAX):

# Recombination phase phi3r is defined in units of laser cycle.
# Spans 0 to 1.
# Only parts within range defined by PHI_MIN and PHI_MAX are calculated for.

        phi3r = np.linspace(0.0, 1.0, N_t)

# idxMIN/MAX are the min. & max. array indices within which is the recollision
# window defined by PHI_MIN/MAX

        idxMIN = np.argmin(np.absolute(phi3r - PHI_MIN))
        idxMAX = np.argmin(np.absolute(phi3r - PHI_MAX))

        # scale phi3r to units of 2 * pi
        phi3r *= 2.0 * np.pi

        # Define phase of ionization and canonical momentum vectors.
        phi1r = np.zeros(N_t)
        psr   = np.zeros(N_t)

        # Solve system
        phi1r[idxMIN : idxMAX], psr[idxMIN : idxMAX] = solveSystem\
        (phi3r[idxMIN : idxMAX])

        idx = np.array([idxMIN, idxMAX])

        # Return all this stuff.

        return phi1r, phi3r, psr, idx

#-------------------------------------------------
#-------------------------------------------------


# Recollision window of time (in units of laser cycle).
# Choose PHI_MIN = 0.30, PHI_MAX = 0.7 for short trajectories.
# Choose PHI_MIN = 0.60, PHI_MAX = 1.0 for long trajectories.
# Choose PHI_MIN = 0.30, PHI_MAX = 1.0 for both trajectories.
```

```
PHI_MIN = 0.3
PHI_MAX = 0.75

# Solve saddle-point equations in scaled coordinates.

# phib : Phase of birth (ionization).
# phir : Phase of recombination.
# psr : Momentum divided A0.

phib, phir, psr, idx = getSolutions (N_t, PHI_MIN, PHI_MAX)

k  = A0 * np.copy(psr)      # Canonical momentum
tb = np.copy(phib) / omega0 # Time of birth (ionization)
tr = np.copy(phir) / omega0 # Time of recombination
dt = tr[1] - tr[0]

filt = np.zeros_like(phir)
tr[0 : idx[0]] = np.nan
tr[idx[1] : N_t] = np.nan

KE = np.power(k + A0 * A(phir), 2.0) / 2.0


# With v = k + A(t)
v = (k + A0 * A(phir))[idx[0] : idx[1]]
arec  = v / np.power(Ip + np.power(v, 2.0), 3.0)


fig, ax = plt.subplots(1, 1, figsize=(4, 3),dpi=300)
ax.semilogy(27.2113 * KE[idx[0] : idx[1]], arec)
ax.set_ylabel(r'$|D(\Omega)|^2$')
ax.set_xlabel("Energy␣(eV)")


plt.show()

#-------------------------------------------------

#%%%

'-------Spectrum␣viewing--------'

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.ndimage import rotate
from scipy.signal import find_peaks, peak_widths


c=2.997e8

#  paste the pathe name of the image  (bd_220819_Power1.73_press35.bmp)
M = plt.imread('bd_220819_Power1.73_press35.bmp',format = None)

M = np.array(M)*1.0
rM = rotate(M, -7.0)
spec = np.mean(rM[560:610,400:1200],axis=0)    # spectrum
bgnd = np.mean(rM[640:695,400:1200],axis=0)    # background noise
diff = spec - bgnd                      # filtering out the  noise
```

```
u = np.arange(401,1201,1) # The desired span of the screen


peaks,_= find_peaks(diff, height = 3,width=5, distance=10)
print ("Peaks details are:",_['peak_heights'])


plt.figure(dpi=300)

#comments
peak_comments = ["x","h27","h29","h31","h33","h35","h37","..","h41","..","h45"]


for i, peak_index in enumerate(peaks):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(u[peak_index], diff[peak_index]),
                     xytext=(u[peak_index], diff[peak_index] + 2),
                      arrowprops=dict(facecolor='black', arrowstyle='->'),
                      fontsize=comment_size)


plt.plot(u,spec, label='Spectrum')
plt.plot(u,bgnd, label='Background')

plt.plot(u,diff,label='Difference between them')
print(u[peaks])
plt.plot(u[peaks], diff[peaks],".")
plt.title("Power=1.73 W, pressure 30 Torr")

plt.xlabel("Pixels of the MCP-phosphor detector ")
plt.ylabel("Mean values of pixels - Intensity (arb. units)")
plt.grid()

legend = plt.legend(loc='center left')

legend_font_size = 8
for text in legend.get_texts():
    text.set_fontsize(legend_font_size)


plt.plot(u,np.zeros_like(diff) , "--", color = "gray")



plt.show()


plt.figure(dpi=300)
plt.imshow(rM, cmap='gray')
plt.title('Original Image')
plt.colorbar()

plt.axhline(y=560, color='r', linestyle='--', label='Start of Spectrum')
plt.axhline(y=610, color='r', linestyle='--', label='End of Spectrum')
plt.axhline(y=640, color='b', linestyle='--', label='Start of Background')
plt.axhline(y=695, color='b', linestyle='--', label='End of Background')
plt.legend()

plt.show()
```

```
#%%%
'-------Curve fitting algorithm--------'




import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.ndimage import rotate
from scipy.signal import find_peaks


c=2.997e8


# Paste the path name of the image  (bd_220819_Power1.73_press35.bmp)
M = plt.imread('bd_220819_Power1.73_press35.bmp',format = None)



M = np.array(M)*1.0
rM = rotate(M, -7.0)
spec = np.mean(rM[560:610,400:1200],axis=0)    # spectrum
bgnd = np.mean(rM[640:695,400:1200],axis=0)    # background noise
diff = spec - bgnd                       # filtering out the  noise



u = np.arange(401,1201,1) # the desired span of the screen

plt.figure(dpi=300)
peaks,_= find_peaks(diff, height = 3,width=5, distance=10)
print ("Peaks details are:",_['peak_heights'])



# comments for each peak
peak_comments = ["21","23","25","h27","h29","h31","h33","h35","h37"]

for i, peak_index in enumerate(peaks):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(u[peak_index], diff[peak_index]),
                     xytext=(u[peak_index], diff[peak_index] + 2),
                      arrowprops=dict(facecolor='black', arrowstyle='->'),
                      fontsize=comment_size)


plt.plot(u,spec, label='Spectrum')
plt.plot(u,bgnd, label='Background')

plt.plot(u,diff,label='Difference between them')
print(u[peaks])
plt.plot(u[peaks], diff[peaks],".")
plt.title("Power=1.73 W, pressure 30 Torr")

plt.xlabel("Pixels of the MCP-phosphor detector ")
plt.ylabel("Mean values of pixels - Intensity (arb. units)")
```

```
plt.grid()


legend = plt.legend(loc='center␣left')


legend_font_size = 8


for text in legend.get_texts():
    text.set_fontsize(legend_font_size)


plt.plot(u,np.zeros_like(diff) , "--", color = "gray")


plt.show()


plt.figure()
plt.imshow(rM, cmap='gray')
plt.title('Original␣Image')
plt.colorbar()
plt.axhline(y=560, color='r', linestyle='--', label='Start␣of␣Spectrum')
plt.axhline(y=610, color='r', linestyle='--', label='End␣of␣Spectrum')
plt.axhline(y=640, color='b', linestyle='--', label='Start␣of␣Background')
plt.axhline(y=695, color='b', linestyle='--', label='End␣of␣Background')
plt.legend()

plt.show()


'-------Curve␣fitting␣procdure--------'



# positions ot the peaks:

 # --- > peak pixels of 0.84 mJ _ 35 Torr ->>> Also copy the pathe name of the
       # corresponding image in M 'Above' as well!
#peak_pixels = np.array([914,  959, 998, 1031, 1061, 1086, 1108])



 # ------ > peak pixels of 1.73 mJ _ 35 Torr
peak_pixels = np.array([ 873, 928, 971, 1008, 1041, 1066, 1094, 1116])




def func(u,u0,A): # calculates y(x)
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9

N = 21

wave_length = np.array([1/N,1/(N+2),1/(N+4),1/(N+6),
                        1/(N+8),1/(N+10),1/(N+12),1/(N+14)])*795e-9

u = np.arange(401,1201,1) # the entire span of the screen
```

```
#Curve fitting
popt, pcov = curve_fit(func,peak_pixels,wave_length,
                       p0=[ 541.229006   , -1645.32998136] ,maxfev=200000)


'''
'The value of A is quite big, for it makes the (tan) yields very
 small value in order to prevent a rapid change.'
'''
plt.plot(peak_pixels,(func(peak_pixels,*popt))*\
         10**(9),"X--",
         label="Curve-fitted␣experimental␣harmonics␣peaks␣$\lambda_N$",
         color='r' )



plt.plot(peak_pixels, (wave_length)*10**(9),"o--",
         label="The␣expected␣(theoretical)␣harmonics␣$\lambda_N$")

wave_length_labels = [f"$N_1$={N}", f"$N_2$={N+2}", f"$N_3$={N+4}",
                      f"$N_4$={N+6}", f"$N_5$={N+8}", f"$N_6$={N+10}",
                      f"$N_7$={N+12}",f"$N_8$={N+14}",
                      "...","..."]  # to Add comments

for i, (x, y) in enumerate(zip(peak_pixels, wave_length * 10 ** 9)):
    if i < len(wave_length_labels):
        comment = wave_length_labels[i]
        plt.annotate(f'{comment}', xy=(x, y), xytext=(x + 5, y),
                     arrowprops=dict(facecolor='black', arrowstyle='->'))


plt.xlabel("Pixels␣")
plt.ylabel("$\lambda$($u_m$)␣[nm]")
plt.title(f"Theoretical␣harmonic␣order␣$N_1$={N},␣P=␣1.73␣W,␣Pr=35␣Torr")
plt.grid()
plt.legend()
plt.show()



residuals= (func(peak_pixels,*popt))-wave_length

ssr = np.sum(residuals**2)

plt.plot(peak_pixels, residuals, 'o-',
         label=f"Residuals,␣($\\mathbf{{SSR:␣{ssr:.4}}}$)")
plt.axhline(0, color='red', linestyle='--')

plt.xlabel("x")
plt.ylabel("Residuals")
plt.title(f"Residuals␣results␣for␣the␣prediction␣($N_1$:␣{N})")
plt.grid(True)
plt.legend()
plt.show()

u_values = np.arange(401, 1201, 1)

wavelengthspan = func(u_values, *popt)  # Calculate wavelengths using func()
```

```
plt.plot(wavelengthspan*1e9,diff,label='Difference␣between␣them')

plt.xlabel("Wave␣length␣[nm]")
plt.ylabel("Intensity␣(a.u.)")
plt.grid()


peak_comments = ["21","23","25","h27","h29","h31","h33","h35","h37"]
peaks, _ = find_peaks(diff, height=3, width=5, distance=10)

#   comments
for i, peak_index in enumerate(peaks):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(wavelengthspan[peak_index] *\
                                       1e9, diff[peak_index]),
                     xytext=(wavelengthspan[peak_index] *\
                             1e9, diff[peak_index] + 2),
                     arrowprops=dict(facecolor='black', arrowstyle='->'),
                     fontsize=comment_size)

plt.xlim(wavelengthspan[0] * 1e9,
         wavelengthspan[-1] * 1e9) # to flip the wave length

plt.show()



wavelengthspan = func(u_values, *popt)  # Calculate wavelengths using func()

omega = (2*np.pi* c) / wavelengthspan
hbar=6.6e-34*2*np.pi
e = 1.6e-19# electron charge

Energy_eV=(omega*1.05457182e-34)/e
width = 8
height = 5


plt.figure(dpi=300)

plt.plot(Energy_eV,diff)

peaks, _ = find_peaks(diff, height=3, width=5, distance=10)


plt.xlabel("Photon␣energy␣[eV]")
plt.ylabel("Intensity␣(a.u.)")
plt.grid()

peak_comments = ["21","23","25","h27","h29","h31","h33","h35","h37"]

for i, peak_index in enumerate(peaks):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(Energy_eV[peak_index],
```

```
                                    diff[peak_index]),
                        xytext=(Energy_eV[peak_index], diff[peak_index] + 2),
                        arrowprops=dict(facecolor='black', arrowstyle='->'),
                        fontsize=comment_size)
plt.show()


# two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(13, 5),dpi=300)

ax1.plot(peak_pixels, (func(peak_pixels, *popt)) * 10**(9), "X--",
         label="Curve-fitted␣experimental␣harmonics␣peaks␣$\lambda_N$",
         color='r')

ax1.plot(peak_pixels, (wave_length) * 10**(9), "o--",
         label="The␣expected␣(theoretical)␣harmonics␣$\lambda_N$")

wave_length_labels = [f"$N_1$={N}", f"$N_2$={N+2}", f"$N_3$={N+4}",
                      f"$N_4$={N+6}", f"$N_5$={N+8}", f"$N_6$={N+10}",
                      f"$N_7$={N+12}",f"$N_8$={N+14}", "...","..."]

for i, (x, y) in enumerate(zip(peak_pixels, wave_length * 10 ** 9)):
    if i < len(wave_length_labels):
        comment = wave_length_labels[i]
        ax1.annotate(f'{comment}', xy=(x, y), xytext=(x + 5, y),
                     arrowprops=dict(facecolor='black', arrowstyle='->'))

ax1.set_xlabel("Horizontal␣axis␣of␣the␣detector")
ax1.set_ylabel("$\lambda$($u_m$)␣[nm]")

ax1.grid()
ax1.legend()



residuals = (func(peak_pixels, *popt)) - wave_length
ssr = np.sum(residuals**2)

ax2.plot(peak_pixels, residuals, 'o-',
         label=f"Residuals,␣($\\mathbf{{SSR:␣{ssr:.4}}}$)")

ax2.axhline(0, color='red', linestyle='--')

ax2.set_xlabel("Horizontal␣axis␣of␣the␣detector␣")
ax2.set_ylabel("Residuals")
ax2.grid()
ax2.legend()

plt.tight_layout()

plt.show()



fig, ax1 = plt.subplots(figsize=(7, 5),dpi=300)

ax1.plot(peak_pixels, (func(peak_pixels, *popt)) * 10**(9), "X--",
         label="Curve-fitted␣experimental␣harmonics␣peaks␣$\lambda_N$",
         color='r')
ax1.plot(peak_pixels, (wave_length) * 10**(9), "o--",
         label="The␣expected␣(theoretical)␣harmonics␣$\lambda_N$")
```

```
wave_length_labels = [f"$N_1$={N}", f"$N_2$={N+2}", f"$N_3$={N+4}",
                      f"$N_4$={N+6}", f"$N_5$={N+8}", f"$N_6$={N+10}",
                      f"$N_7$={N+12}",f"$N_8$={N+14}",
                      "...","..."]  # to add comments

for i, (x, y) in enumerate(zip(peak_pixels, wave_length * 10 ** 9)):
    if i < len(wave_length_labels):
        comment = wave_length_labels[i]
        ax1.annotate(f'{comment}', xy=(x, y), xytext=(x + 5, y),
                     arrowprops=dict(facecolor='black', arrowstyle='->'))

ax1.set_xlabel("Horizontal axis of the detector")
ax1.set_ylabel("$\lambda$($u_m$) [nm]")
ax1.grid()
ax1.legend()

ax2 = ax1.twinx()

residuals = (func(peak_pixels, *popt)) - wave_length
ssr = np.sum(residuals**2)

ax2.plot(peak_pixels, residuals, 'o--',
         label=f"Residuals, ($\\mathbf{{SSR: {ssr:.4}}}$)", color="g")

ax2.axhline(0, color='red', linestyle='--')

ax2.set_ylabel("Residuals")

ax2.legend(loc="lower left")

plt.tight_layout()

plt.show()


#%%%

'-----------------------1st 800 nm Phase matching Maps-----------------------'


import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from matplotlib.colors import ListedColormap


# Paste the path of the intensities named:
excel_file = 'Intensities_ 800nm(1)_0.84_1.28_1.73.xlsx'

xldata = pd.read_excel(excel_file,
                       sheet_name = "P173",
                       index_col = 0)

xl_clean = xldata.replace(np.NaN, 0)


print(xl_clean)

heatmap_data = xl_clean
plt.figure(dpi=300)
```

```
sns.heatmap(heatmap_data, cbar_kws={'label': 'Intesity (arb. units)'},
            annot=False, vmin=0, vmax=35)

plt.tight_layout()

plt.xlabel("Harmonic number ")
plt.ylabel("Pressure [Torr] ")
plt.title("Driving laser pulse energy 1.73 mJ")


plt.show()

#------------------------------------------------------------
#------------------------------------------------------------




# Please add the path name of the file: P_0.84_1.28_1.73.xlsx
excel_file = 'Path of the file: Peak sheet P_0.84_1.28_1.73.xlsx'

xldata = pd.read_excel(excel_file,
                       sheet_name = "P128",
                       index_col = 0)

xl_clean2 = xldata.replace(np.NaN, 0)


print(xl_clean)

heatmap_data = xl_clean2
plt.figure(dpi=300)

sns.heatmap(heatmap_data, cbar_kws={'label': 'Intesity (arb. units)'},
            annot=False, vmin=0, vmax=35)

plt.tight_layout()

plt.xlabel("Harmonic number ")
plt.ylabel("Pressure [Torr] ")
plt.title("Driving laser pulse energy 1.28 mJ")


plt.show()


#------------------------------------------------------------
#------------------------------------------------------------


# Please add the path name of the file: P_0.84_1.28_1.73.xlsx
excel_file = 'Path of the file: Peak sheet P_0.84_1.28_1.73.xlsx'

xldata = pd.read_excel(excel_file,
                       sheet_name = "P0,84_2",
                       index_col = 0)

xl_clean3 = xldata.replace(np.NaN, 0)


print(xl_clean)
```

```
heatmap_data = xl_clean3
plt.figure(dpi=300)

sns.heatmap(heatmap_data,  cbar_kws={'label': 'Intesity␣(arb.␣units)'},
            annot=False,vmin=0, vmax=35)


plt.tight_layout()


plt.xlabel("Harmonic␣number␣")
plt.ylabel("Pressure␣[Torr]␣")
plt.title("Driving␣laser␣pulse␣energy␣0.84␣mJ")


plt.show()


#------------------------------------------------------------
#------------------------------------------------------------


# Please add the path name of the file: P_0.84_1.28_1.73.xlsx
excel_file = 'Path␣of␣the␣file:␣Peak␣sheet␣P_0.84_1.28_1.73.xlsx'

xldata = pd.read_excel(excel_file,
                       sheet_name = "P0,4",
                       index_col = 0)

xl_clean2 = xldata.replace(np.NaN, 0)


print(xl_clean)

heatmap_data = xl_clean2
plt.figure(dpi=300)

sns.heatmap(heatmap_data, cbar_kws={'label': 'Intesity␣(arb.␣units)'},
            annot=False, vmin=0, vmax=35)

plt.tight_layout()

plt.xlabel("Harmonic␣number␣")
plt.ylabel("Pressure␣[Torr]␣")
plt.title("Driving␣laser␣pulse␣energy␣0.4␣mJ")


plt.show()

#%%


'-----------------------Image␣Averiging-----------------------'

'␣This␣code␣is␣used␣to␣average␣both␣the␣800␣nm␣shots␣and␣the␣400␣nm␣shots␣'


import os
import numpy as np
from PIL import Image

# Specify the folder path where the images are located
```

```
folder_path = "␣path␣of␣the␣foler␣where␣the␣single␣shots␣are!"

# Specify the path to the destination folder where the
# averaged images will be stored
destination_folder = "The␣path␣of␣the␣output␣folder"

os.chdir(folder_path)

all_files = os.listdir()
image_files = [filename for filename in all_files if\
               filename.endswith((".png", ".PNG"))]

image_groups = {}

for image_file in image_files:
    # identify the hour-minute name from the file name
    hour_minute = image_file.split("_")[2][:4]

    if hour_minute in image_groups:
        image_groups[hour_minute].append(image_file)
    else:
        image_groups[hour_minute] = [image_file]

os.makedirs(destination_folder, exist_ok=True)

for group_name, group_files in image_groups.items():
    image_path = group_files[0]
    image = Image.open(image_path)
    width, height = image.size
    num_images = len(group_files)

    average_image = np.zeros((height, width, 3), dtype=np.float32)

    for image_file in group_files:
        try:
            image = Image.open(image_file)
            image_array = np.array(image, dtype=np.float32)
            expected_shape = (height, width, 3)
            if image_array.shape != expected_shape:
                # Check if image has only one channel
                if image_array.ndim == 2:
                    image_array = np.stack((image_array,) * 3, axis=-1)
                else:
                    raise ValueError(f"Shape␣{image_array.shape}␣does␣not\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣match␣the␣expected␣dimensions\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣{expected_shape}.")

            average_image += image_array
        except (IOError, ValueError) as e:
            print(f"Skipping␣{image_file}.␣Error:␣{str(e)}")

    average_image /= num_images

    average_image = np.round(average_image).astype(np.uint8)

    result_image = Image.fromarray(average_image, mode="RGB")

    result_image_path = os.path.join(destination_folder,
                                     f"average_{group_name}.png")

    result_image.save(result_image_path)
```

```
#%%
# ----------------Curve fitting of the 800_2 ----------------

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.ndimage import rotate
from scipy.signal import find_peaks



c=2.997e8



# Paste the path name of the image  (average_1155.png)
M = plt.imread('average_1155.png',format = None)



M = np.array(M)*1.0
rM = rotate(M, 0.0)



spec_800_2 = np.mean(rM[290:410, 400:1200],axis=0)    # spectrum
bgnd_800_2 = np.mean(rM[420:460, 400:1200],axis=0)    # background noise


reduced_spec_800_2 = spec_800_2[:, 0]*255
reduced_bgnd_800_2 = bgnd_800_2[:, 0]*255

diff_800_2 = reduced_spec_800_2 - reduced_bgnd_800_2  # filtering out the noise


u = np.arange(400,1200,1) # the desired span of the screen


peaks800_2,_= find_peaks(diff_800_2, height =3,width=6, distance=18)

rounded_peak_heights = [round(val, 4) for val in _['peak_heights']]
print("Peaks␣details␣are:", rounded_peak_heights)


plt.figure(dpi=300)

# comments
peak_comments = ["h21","h23","h25","h27","h29"]

for i, peak_index in enumerate(peaks800_2):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(u[peak_index], diff_800_2[peak_index]),
                     xytext=(u[peak_index], diff_800_2[peak_index] + 4),
                      arrowprops=dict(facecolor='black', arrowstyle='-'),
                      fontsize=comment_size)
```

```
plt.plot(u,reduced_spec_800_2, label='Spectrum')
plt.plot(u,reduced_bgnd_800_2, label='Background')

plt.plot(u,diff_800_2,label='Difference between them')
plt.plot(u[peaks800_2], diff_800_2[peaks800_2],".")

plt.xlabel("Pixels of the MCP-phosphor detector ")
plt.ylabel("Mean values of pixels - Intensity (arb. units)")
plt.grid()

legend = plt.legend(loc='center left')

legend_font_size = 8

for text in legend.get_texts():
    text.set_fontsize(legend_font_size)


plt.plot(u,np.zeros_like(diff_800_2) , "--", color = "gray")

plt.show()


plt.figure(dpi=300)
plt.imshow(rM, cmap='gray')
plt.colorbar()
plt.axhline(y=280, color='r', linestyle='--', label='Spectrum')
plt.axhline(y=418, color='r', linestyle='--')
plt.axhline(y=420, color='b', linestyle='--')
plt.axhline(y=460, color='b', linestyle='--', label='Background')
legend = plt.legend(loc='lower left')

plt.legend()

plt.show()


### ------------------------------> The curve fit of the 2nd 800nm --> 1.8 mJ


peak_pixels_800_2 = np.array ([840, 884, 923, 956, 986])#800nm_2




def func(u,u0,A): # calculates y(x)
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9

N = 21 # The starting number of the Harmonic sequence

wave_length_800_2 = np.array([1/N,1/(N+2),1/(N+4),1/(N+6),1/(N+8)])*795e-9
u = np.arange(401,1201,1) # the desired span of the screen

popt_800_2, pcov = curve_fit(func,peak_pixels_800_2 ,wave_length_800_2,
                             p0=[ 670.01296871, -1253.31351353] ,maxfev=200000)
```

```
# Array of pixel positions for the desired span of the screen
u_values = np.arange(401, 1201, 1)

 # Calculate wavelengths using func()
wavelengthspan_800_2 = func(u_values, *popt_800_2)



plt.xlim(wavelengthspan_800_2[0] *\
         1e9, wavelengthspan_800_2[-1] * 1e9) # to flip the wave length

plt.plot(wavelengthspan_800_2*1e9,diff_800_2,label='Difference␣between␣them')

plt.xlim(50,20)
plt.xlabel("Wave␣length␣[nm]")
plt.ylabel("Intensity␣(a.u.)")
plt.grid()


peak_comments = ["h21","h23","h25","h27","h29"]
peaks, _ = find_peaks(diff_800_2, height=3, width=6, distance=18)

#   comments
for i, peak_index in enumerate(peaks):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(wavelengthspan_800_2[peak_index] * 1e9,
                                       diff_800_2[peak_index]),
                     xytext=(wavelengthspan_800_2[peak_index] * 1e9,
                             diff_800_2[peak_index] + 2),
                     arrowprops=dict(facecolor='black', arrowstyle='->'),
                     fontsize=comment_size)

plt.show()



# Calculate wavelengths using func()
wavelengthspan_800_2 = func(u_values, *popt_800_2)

omega_800_2 = (2*np.pi* c) / wavelengthspan_800_2
hbar=6.6e-34*2*np.pi
e = 1.6e-19

Energy_eV_800_2=(omega_800_2*1.05457182e-34)/e
width = 8  # Width in inches
height = 5  # Height in inches

figure_size = (8, 6)

plt.figure(figsize=figure_size,dpi=300)

plt.plot(Energy_eV_800_2,diff_800_2)

peaks, _ = find_peaks(diff_800_2, height=3, width=6, distance=18)



peak_comments = ["h21","h23","h25","h27","h29"]
```

```
for i, peak_index in enumerate(peaks):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(Energy_eV_800_2[peak_index],
                                       diff_800_2[peak_index]),
                     xytext=(Energy_eV_800_2[peak_index],
                             diff_800_2[peak_index] + 2),
                     arrowprops=dict(facecolor='black', arrowstyle='->'),
                     fontsize=comment_size)


plt.xlim(25,60)
plt.xlabel("Photon energy [eV]")
plt.ylabel("Intensity (a.u.)")
plt.grid()
plt.show()



figure_size = (8, 6)

plt.figure(figsize=figure_size,dpi=300)

# to flip the wave length
plt.xlim(wavelengthspan_800_2[0] * 1e9, wavelengthspan_800_2[-1] * 1e9)




# Calculate wavelengths using func()
wavelengthspan_800_2 = func(u_values, *popt_800_2)

omega_800_2 = (2*np.pi* c) / wavelengthspan_800_2
hbar=6.6e-34*2*np.pi
e = 1.6e-19

Energy_eV_800_2=(omega_800_2*1.05457182e-34)/e
width = 8  # Width in inches
height = 5  # Height in inches



plt.plot(Energy_eV_800_2,diff_800_2)

peaks,_= find_peaks(diff_800_2, height =3,width=6, distance=18)


plt.xlabel("Photon energy [eV]")
plt.ylabel("Intensity (a.u.)")
plt.grid()
plt.xlim(20,50)

peak_comments = ["h21","h23","h25","h27","h29"]

for i, peak_index in enumerate(peaks):
    if i < len(peak_comments):
        comment = peak_comments[i]
        comment_size = 10
        plt.annotate(f'{comment}', xy=(Energy_eV_800_2[peak_index],
                                       diff_800_2[peak_index]),
                     xytext=(Energy_eV_800_2[peak_index],
```

```
                        diff_800_2[peak_index] + 2),
                    arrowprops=dict(facecolor='black', arrowstyle='->'),
                    fontsize=comment_size)
plt.show()


u_values = np.arange(401, 1201, 1)

# Calculate wavelengths using func()
wavelengthspan_800_2 = func(u_values, *popt_800_2)



plt.figure(dpi=300)

plt.plot(Lam_0/wavelengthspan_800_2,diff_800_2,label='800_2 vs orders')



plt.xlim(15,35)

plt.legend()
plt.xlabel("Harmonics orders")
plt.ylabel("Intensity (a.u.)")
plt.grid()


#%%


'-------Heat maps 800_2-------'

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from matplotlib.colors import ListedColormap


# Paste the path of the intensities named:
excel_file = 'Intensities_800nm(2)_1.44_1.8_2.16.xlsx'


xldata = pd.read_excel(excel_file,
                       sheet_name = "1.8Watt",
                       index_col = 0)

xl_clean = xldata.replace(np.NaN, 0)


print(xl_clean)

heatmap_data = xl_clean
plt.figure(dpi=300)

sns.heatmap(heatmap_data, cbar_kws={'label': 'Intesity (arb. units)'},
            annot=False, vmin=0, vmax=45)

plt.tight_layout()

plt.xlabel("Harmonic number ")
plt.ylabel("Pressure [Torr] ")
```

```
plt.title("Driving␣laser␣pulse␣energy␣␣␣1.8␣mJ")



#------------------------------------------------------------
#------------------------------------------------------------

xldata = pd.read_excel(excel_file,
                       sheet_name = "1.4Watt",
                       index_col = 0)

xl_clean2 = xldata.replace(np.NaN, 0)


print(xl_clean)

heatmap_data = xl_clean2
plt.figure(dpi=300)

sns.heatmap(heatmap_data, cbar_kws={'label': 'Intesity␣(arb.␣units)'},
            annot=False, vmin=0, vmax=45)

plt.tight_layout()

plt.xlabel("Harmonic␣number␣")
plt.ylabel("Pressure␣[Torr]␣")
plt.title("Driving␣laser␣pulse␣energy␣␣␣1.44␣mJ")


#------------------------------------------------------------
#------------------------------------------------------------

xldata = pd.read_excel(excel_file,
                       sheet_name = "2.16Watt",
                       index_col = 0)

xl_clean3 = xldata.replace(np.NaN, 0)


print(xl_clean)

heatmap_data = xl_clean3
plt.figure(dpi=300)

sns.heatmap(heatmap_data,  cbar_kws={'label': 'Intesity␣(arb.␣units)'},
            annot=False,vmin=0, vmax=45)


plt.tight_layout()


plt.xlabel("Harmonic␣number␣")
plt.ylabel("Pressure␣[Torr]␣")
plt.title("Driving␣laser␣pulse␣energy␣2.16␣mJ")


plt.show()

#%%%


'---------------␣400␣nm␣interpolation␣-------------------'
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.ndimage import rotate
from scipy.signal import find_peaks
from scipy.interpolate import interp1d


c=2.997e8


# Paste the path name of the image  (average_1912.png)
M_400_1912 = plt.imread('Path name of the ... average_1912.png',format = None)


rM_400_1912 = rotate(M_400_1912, 0)

print(np.max(M_400_1912))


u = np.arange(1,1281,1) # the entire span of the screen


spec_400_1912 = np.mean(rM_400_1912[280:420, 0:1281], axis=0)# spectrum
bgnd_400_1912 = np.mean(rM_400_1912[420:460, 0:1281], axis=0)# background noise

reduced_spec_400_1912 = spec_400_1912[:,0]
reduced_bgnd_400_1912 = bgnd_400_1912[:,0]

# filtering out the noise
diff_400_1912 = reduced_spec_400_1912 - reduced_bgnd_400_1912

plt.figure(dpi=300)

plt.plot(u, reduced_spec_400_1912*255, label='Spectrum')
plt.plot(u, reduced_bgnd_400_1912*255, label='Background')
plt.plot(u, diff_400_1912*255, label='Difference')
plt.ylabel("Intensity a.u.")
plt.xlabel("Horisental axis of the CCD camera")

plt.legend()
plt.grid()
plt.show()

plt.figure(dpi=300)
plt.imshow(rM_400_1912, cmap='gray')
plt.colorbar()
plt.axhline(y=280, color='r', linestyle='--', label='Spectrum')
plt.axhline(y=418, color='r', linestyle='--')
plt.axhline(y=420, color='b', linestyle='--')
plt.axhline(y=460, color='b', linestyle='--', label='Background')
plt.legend()

plt.show()

#%%%


'--------------Interpolation of the HHG from the 400 nm curves---------------'
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.ndimage import rotate
from scipy.signal import find_peaks
from scipy.interpolate import interp1d
import os



# Paste the path name of the folder that contained the averaged 400 nm images
folder = 'Path name of the folder that contains the 400 nm averaged images'

all_files = os.listdir(folder)

# Filter for images
average_img_files = [f for f in all_files if f.startswith("average_") and\
                     f.endswith(".png")]

average_img_files = sorted(average_img_files, key=lambda\
                           x: int(x.split('_')[1].split('.png')[0]))

average_s = []

for img_file in average_img_files:
    img_path = os.path.join(folder, img_file)
    img = plt.imread(img_path, format=None)

    img_s = np.mean(img[280:420, :], axis=0)
    img_s =  img_s[:, 0]
    average_s.append(img_s - np.mean(img_s[1020:1080]))


average_b = average_s[38].copy()
average_b[558:591] = np.interp(np.arange(558,591),[558,591],
                               [average_s[38][558],average_s[38][591]])

average_b[742:767] = np.interp(np.arange(742,767),[742,767],
                               [average_s[38][742],average_s[38][767]])
b_mean = np.mean(average_b[460:520])



average_spec=[]
for s in average_s:
 spec = s - average_b*np.mean(s[460:520])/b_mean
 average_spec.append(spec)


plt.figure(dpi=300)
plt.ylabel("Inetnsity a.u.")
plt.xlabel("Horisental axis of the CCD camera")
plt.grid()
for s in average_s:
    plt.plot(s*255)



plt.figure(dpi=300,)        ## All the spectra against the

plt.xlabel("Horizontal axis of the CCD camera")
```

```
plt.ylabel("Inetnsity␣a.u.")


for s in average_spec:
    plt.plot(s*255)
plt.grid()


# Paste the path name of the image  (average_1155.png)
M_800_2 = plt.imread('Path␣name␣of␣the␣␣...␣average_1155.png',format = None)




M_800_2 = np.array(M_800_2)*1.0
rM_800_2 = rotate(M_800_2, 0.0)



spec_800_2 = np.mean(rM_800_2[290:410, 0:1280],axis=0)    # spectrum
bgnd_800_2 = np.mean(rM_800_2[420:460, 0:1280],axis=0)    # background noise

reduced_spec = spec_800_2[:, 0]
reduced_bgnd = bgnd_800_2[:, 0]

diff_800_2 = reduced_spec - reduced_bgnd



peak_pixels_800_2 = np.array ([840,884,923,956,986])#800nm_2


def func_800_2(u,u0,A): # calculates y(x)
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9

N = 21 # The starting number of the Harmonic sequence

wave_length_800_2 = np.array([1/N,1/(N+2),1/(N+4),1/(N+6),1/(N+8)])*795e-9


popt_800_2, pcov = curve_fit(func_800_2,peak_pixels_800_2,wave_length_800_2,
                             p0=[   670.01296871, -1253.31351353]
                             ,maxfev=200000)

u = np.arange(1,1281,1) # the desired span of the screen


# Calculate wavelengths using func()
wavelengthspan_800_2 = func_800_2(u, *popt_800_2)



plt.grid()

plt.figure(dpi=300)
plt.xlim(13, 25)


for s in average_spec:
```

```
    #s1=s1+s
    plt.plot(Lam_0/wavelengthspan_800_2, s * 255)


plt.xlabel("Harmonic order")
plt.ylabel("Intensity a.u. ")
plt.xlim(17.7,18.7)

plt.ylim(-3,10)

plt.grid()
plt.show()




plt.figure(dpi=300)
plt.xlim(13, 25)


for s in average_spec:
    plt.plot(Lam_0/wavelengthspan_800_2, s * 255)


plt.xlabel("Harmonic order")
plt.ylabel("Intensity a.u. ")

plt.grid()
plt.show()



file_name = 'average_1912.png'

if file_name in average_img_files:
    file_number = average_img_files.index(file_name)
else:
    print("File not found.")
    exit()

peaks, properties = find_peaks(average_spec[file_number] * 255, height=1,
                               width=1, distance=10)

print("Peak heights are-------------------->>>>", properties['peak_heights'])



plt.figure(dpi=300)

plt.plot(Lam_0 / wavelengthspan_800_2, average_spec[file_number] * 255,
         label="$\lambda_{pulse}$ = 400 nm\nE$_{pulse}$ = 0.46 mJ\nPressure\
                = 20 Torr\nGDD = -300\nTOD = 10000")  #f"{file_name}"

plt.legend()

plt.xlabel("Harmonic order")
plt.ylabel("Intensity a.u. ")
plt.xlim(11,45)

plt.grid()
plt.show()
```

```
plt.figure(dpi=300)
plt.grid()
plt.ylabel("Intensity␣a.u.")

plt.xlabel("Horizontal␣axis␣of␣the␣CCD␣camera")

plt.plot(average_b*255,label="Created␣background",zorder =2)
plt.plot(average_s[38]*255, label="Origional␣spectrum",zorder=1)
plt.plot((average_s[38]-average_b)*255,label="Difference")

plt.legend(loc="lower␣center")


plt.figure(dpi=300)

for s in average_s:
    plt.plot(s*255)


plt.ylabel("Inetnsity␣a.u.")
plt.xlabel("Horisental␣axis␣of␣the␣CCD␣camera")
plt.grid()

interpolated_backgrounds = []

for average in average_s:
    average_b = average.copy()
    average_b[558:591] = np.interp(np.arange(558, 591), [558, 591],
                                   [average[558], average[591]])

    average_b[742:767] = np.interp(np.arange(742, 767), [742, 767],
                                   [average[742], average[767]])
    b_mean = np.mean(average_b[460:520])

    interpolated_backgrounds.append(average_b)

plt.figure(dpi=300)
plt.ylabel("Intensity␣a.u.")
plt.xlabel("Horizontal␣axis␣of␣the␣CCD␣camera")
plt.grid()
for bg in interpolated_backgrounds:
    plt.plot(bg * 255)
plt.show()

'-------ALL␣THE␣SPECTRA,␣800_1␣,␣800_2,␣400␣nm-------'

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.ndimage import rotate
from scipy.signal import find_peaks, peak_widths
from scipy.interpolate import interp1d
import os


#### 1.73 ###'


# Paste the path name of the image  (Abd_220819_Power1.73_press35.bmp)
M_173 = plt.imread('Abd_220819_Power1.73_press35.bmp',format = None)
```

```
M_173 = np.array(M_173)*1.0
rM_173 = rotate(M_173, -7.0)
spec_173 = np.mean(rM_173[560:610,0:1280],axis=0)     # spectrum
bgnd_173 = np.mean(rM_173[650:695,0:1280],axis=0)     # background noise
diff_173 = spec_173 - bgnd_173                       # filtering out the  noise



u = np.arange(1,1281,1) # the desired span of the screen

def func_173(u,u0,A):
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9


N = 21 # The starting number of the Harmonic sequence

peak_pixels_173 = np.array([ 873 , 928,  971, 1008, 1041, 1066, 1094, 1116])

wave_length_173 = np.array([1/N,1/(N+2),1/(N+4),1/(N+6),1/(N+8),1/(N+10),
                            1/(N+12),1/(N+14)])*795e-9


popt_173, pcov = curve_fit(func_173,peak_pixels_173,wave_length_173,
                           p0=[  657.95177175, -1486.51659083] ,maxfev=200000)



u_values = np.arange(1, 1281, 1)

# Calculate wavelengths using func()
wavelengthspan_173 = func_173(u_values, *popt_173)



plt.figure(dpi=300)


##### 800 nm 1.28 mJ _ 35 Torr ######


c=2.997e8


# Paste the path name of the image  (Abd_220819_Power1.28_press35.bmp)
M_128 = plt.imread('Abd_220819_Power1.28_press35.bmp',format = None)


M_128 = np.array(M_128)*1.0
rM_128 = rotate(M_128, -7.0)


spec_128 = np.mean(rM_128[560:610,0:1280],axis=0)     # spectrum
bgnd_128 = np.mean(rM_128[640:695,0:1280],axis=0)     # background noise
diff_128 = spec_128 - bgnd_128
```

```
peak_pixels_128 = np.array([869, 925, 969, 1005, 1038, 1065, 1092])


def func_128(u,u0,A): # calculates y(x)
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9

N = 21 # The starting number of the Harmonic sequence

wave_length_128 = np.array([1/N,1/(N+2),1/(N+4),1/(N+6),1/(N+8),1/(N+10),
                            1/(N+12)])*795e-9


popt_128, pcov = curve_fit(func_128,peak_pixels_128,wave_length_128,
                           p0=[ -3.40548495e+03, -1.47202285e+10] ,
                           maxfev=50000000)

# Calculate wavelengths using func()
wavelengthspan_128 = func_128(u_values, *popt_128)


######### 0.84 ########


# Paste the path name of the image  (Abd_220819_Power0.84_press35.bmp)
M_084 = plt.imread('Abd_220819_Power0.84_press35.bmp',format = None)


M_084 = np.array(M_084)*1.0
rM_084 = rotate(M_084, -7.0)
spec_084 = np.mean(rM_084[560:610,0:1280],axis=0)    # spectrum
bgnd_084 = np.mean(rM_084[640:695,0:1280],axis=0)    # background noise
diff_084 = spec_084 - bgnd_084

# ------ > 0.84 _ 35 Torr
peak_pixels_084 = np.array([916,  960, 1000, 1032, 1061, 1087, 1107])


def func_084(u,u0,A): # calculates y(x)
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9

N = 23 #The starting number of the Harmonic sequence

wave_length_084 = np.array([1/N,1/(N+2),1/(N+4),1/(N+6),1/(N+8),1/(N+10),
                            1/(N+12)])*795e-9


popt_084, pcov = curve_fit(func_084,peak_pixels_084,wave_length_084,
                           p0=[  -3.40548495e+03, -1.47202285e+12] ,
                           maxfev=200000) #fit the difference in harmonics to 2


# Calculate wavelengths using func()
wavelengthspan_084 = func_084(u_values, *popt_084)
```

```
Lam_0/wavelengthspan_173,

plt.plot(Lam_0/wavelengthspan_173,diff_173,label='800 nm, 1.73 mJ, 35 Torr')

plt.plot(Lam_0/wavelengthspan_173,diff_128,label='800 nm, 1.28 mJ, 35 Torr')

#plt.plot(Lam_0/wavelengthspan_173,diff_084,label='800 nm, 0.84 mJ, 35 Torr')


#### --------------------------800 _ 2 _ 1155


# Paste the path name of the image  (average_1155.png)

M_800_2 = plt.imread('average_1155.png', format=None) # 800nm_2


M_800_2 = np.array(M_800_2)*1.0
rM_800_2 = rotate(M_800_2, 0.0)



spec_800_2 = np.mean(rM_800_2[290:410, 0:1280],axis=0)    # spectrum
bgnd_800_2 = np.mean(rM_800_2[420:460, 0:1280],axis=0)    # background noise

reduced_spec = spec_800_2[:, 0]
reduced_bgnd = bgnd_800_2[:, 0]

diff_800_2 = reduced_spec - reduced_bgnd

# spec_800_2 = np.mean(rM_800_2[280:420, 0:1280],axis=0)    # spectrum
# bgnd_800_2 = np.mean(rM_800_2[420:460, 0:1280],axis=0)    # background noise

# diff_800_2 = spec_800_2 - bgnd_800_2


peak_pixels_800_2 = np.array ([840,884,923,956,986])#800nm_2


def func_800_2(u,u0,A): # calculates y(x)
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9

N = 21 #The starting number of the Harmonic sequence

wave_length_800_2 = np.array([1/N,1/(N+2),1/(N+4),1/(N+6),1/(N+8)])*795e-9


popt_800_2, pcov = curve_fit(func_800_2,peak_pixels_800_2,wave_length_800_2,
                             p0=[670.01296871, -1253.31351353] ,maxfev=200000)


# Calculate wavelengths using func()
wavelengthspan_800_2 = func_800_2(u_values, *popt_800_2)




# Paste the path name of the folder  (310523_averaged images of 270423_correct)
```

```
folder = '310523_averaged␣images␣of␣270423_correct'

all_files = os.listdir(folder)

# Filter for images (assuming they are png files; adjust as needed)
average_img_files = [f for f in all_files if f.startswith("average_") and\
                     f.endswith(".png")]

average_img_files = sorted(average_img_files, key=lambda\
                           x: int(x.split('_')[1].split('.png')[0]))
average_s = []

for img_file in average_img_files:
    img_path = os.path.join(folder, img_file)
    img = plt.imread(img_path, format=None)

    img_s = np.mean(img[280:420, :], axis=0)
    img_s =  img_s[:, 0]
    average_s.append(img_s - np.mean(img_s[1020:1080]))



average_b = average_s[38].copy()
average_b[558:591] = np.interp(np.arange(558,591),[558,591],
                               [average_s[38][558],average_s[38][591]])

average_b[742:767] = np.interp(np.arange(742,767),[742,767],
                               [average_s[38][742],average_s[38][767]])
b_mean = np.mean(average_b[460:520])



average_spec=[]
for s in average_s:
 spec = s - average_b*np.mean(s[460:520])/b_mean #- average_b0
 average_spec.append(spec)


M_084 = np.array(M_084)*1.0
rM_084 = rotate(M_084, -7.0)
spec_084 = np.mean(rM_084[560:610,0:1280],axis=0)    # spectrum
bgnd_084 = np.mean(rM_084[640:695,0:1280],axis=0)    # background noise
diff_084 = spec_084 - bgnd_084

file_name = 'average_1913.png'

# Find the index of the file_name in average_img_files
if file_name in average_img_files:
    file_number = average_img_files.index(file_name)
else:
    print("File␣not␣found.")
    exit()

# Find peaks

# Print peak heights
#print("Peak heights are---------------------->>>>", properties['peak_heights'])



#Nfit(np.arange(0,1280)*1.1-100,*p)+p[2]
```

```
peak_pixels_400_1912 = np.array([583, 764]) # ------ >400_1912Torr


def func_400_1912(u,u0,A): # calculates y(x)
    Lambda = 100e-9*np.sin(np.arctan((u-u0)/A) + 30*np.pi/180)
    return Lambda

Lam_0 = 795e-9


N = 14 #The starting number of the Harmonic sequence


wave_length_400_1912 = (np.array([1/N,1/(N+4)])*795e-9)/2


popt_400_1912, pcov = curve_fit(func_400_1912,peak_pixels_400_1912,
                                wave_length_400_1912,
                                p0=[665.78698895, -1271.77440922] ,
                                maxfev=2000000000)



# Calculate wavelengths using func()
wavelengthspan_400_1912 = func_400_1912(u_values, *popt_400_1912)


file_name = 'average_1914.png'

if file_name in average_img_files:
    file_number = average_img_files.index(file_name)
else:
    print("File not found.")
    exit()


#plt.figure(dpi=300)


plt.plot(Lam_0/wavelengthspan_800_2,diff_800_2*255,
         label='800 nm, 1.8 mJ, 21 Torr')

plt.plot(Lam_0/wavelengthspan_800_2,average_spec[file_number] * 255,
         label='400 nm, 0.46 mJ, 10 Torr ' ) #label=f"{file_name}"



#plt.xlabel("CCD pixels")


plt.xlabel("High harmonic order")

#plt.xlabel("Pixels of the CCD")

plt.ylabel("Intensity (a.u.)")

plt.grid()
plt.legend(loc='upper right', fontsize=8)
plt.xlim(10, 40)
```

```
plt.show()



omega_173 = (2*np.pi* c) / wavelengthspan_173
hbar=6.6e-34*2*np.pi#     1.05457182e-34 #j.s
e = 1.6e-19# electron charge

Energy_eV_173=(omega_173*1.05457182e-34)/e


plt.figure(dpi=300)


plt.plot(Energy_eV_173,diff_173,label='800 nm, 1.73 mJ, 35 Torr')

#plt.plot(Energy_eV_173,diff_128,label='800 nm, 1.28 mJ, 35 Torr')

#plt.plot(Energy_eV_173,diff_084,label='800 nm, 0.84 mJ, 35 Torr')




#---------------------------

omega_800_2 = (2*np.pi* c) / wavelengthspan_800_2
hbar=6.6e-34*2*np.pi#     1.05457182e-34 #j.s
e = 1.6e-19# electron charge
Energy_eV_800_2=(omega_800_2*1.05457182e-34)/e




omega_400 = (2*np.pi* c) / (wavelengthspan_400_1912)
hbar=6.6e-34*2*np.pi#     1.05457182e-34 #j.s
e = 1.6e-19# electron charge
Energy_eV_400=(omega_400*1.05457182e-34)/e




plt.plot(Energy_eV_800_2,average_spec[file_number] * 255,'b' ,
         label='400 nm, 0.46 mJ, 10 Torr ' ) #label=f"{file_name}"

plt.plot(Energy_eV_800_2,diff_800_2* 255, 'r',label='800 nm, 1.8 mJ, 21 Torr' )


# Create the figure with the specified size




# Create the figure with the specified size
#figure_size = (8, 6)  # Adjust the width and height as needed

#figure_size = (5, 3)  # Adjust the width and height as needed

plt.xlabel("Photon energy")
plt.xlim(15,80)

#plt.ylim(-3,)
```

```
#plt.xlabel("Pixels of the CCD")

plt.ylabel("Intensity (a.u.)")

plt.grid()
plt.legend(loc='upper right', fontsize=8)

plt.show()


plt.figure(dpi=300)
#figure_size = (5, 1)  # Adjust the width and height as needed

plt.plot(Energy_eV_800_2,average_spec[file_number] * 255,'b' ,
         label='400 nm, 0.46 mJ, 10 Torr ' ) #label=f"{file_name}"


plt.xlabel("Photon energy")
#plt.xlim(25,80)

#plt.ylim(-3,)
#plt.xlabel("Pixels of the CCD")

plt.ylabel("Intensity (a.u.)")

plt.grid()
plt.legend(loc='upper right', fontsize=8)

plt.show()


#%%

'-------Heat maps 400nm-------'



import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Please paste the path name of the excel file :
excel_file = 'Path name of the ... Intensities_400nm_h9_h11.xlsx'

xldata = pd.read_excel(excel_file,
                       sheet_name = "TOD_tot",
                       index_col = 0)

xl_clean = xldata.replace(np.NaN, 0)


print(xl_clean)

heatmap_data = xl_clean
xl_clean = xldata.fillna(0)

plt.figure(figsize=(7, 8),dpi=300)

ax=sns.heatmap(heatmap_data, cbar_kws={'label': 'Intesity (arb. units)'},
               annot=False, vmin=0, vmax=22)
```

```
plt.tight_layout()


xldata = pd.read_excel(excel_file, sheet_name="TOD_tot", index_col=0)
xl_clean = xldata.fillna(0)


plt.xlabel("Harmonic␣order␣", labelpad=30)
plt.ylabel("Pressure␣[Torr]␣", labelpad=50)



plt.show()
```

References

(1) Rose, T. S.; Rosker, M. J.; Zewail, A. H. Femtosecond real-time observation of wave packet oscillations (resonance) in dissociation reactions. *The Journal of Chemical Physics* **1988**, *88*, 6672–6673.

(2) Yamane, K.; Zhang, Z.; Oka, K.; Morita, R.; Yamashita, M.; Suguro, A. Optical pulse compression to 3.4 fs in the monocycle region by feedback phase compensation. *Optics Letters* **2003**, *28*, 2258–2260.

(3) Winterfeldt, C.; Spielmann, C.; Gerber, G. Colloquium: Optimal control of high-harmonic generation. *Reviews of Modern Physics* **2008**, *80*, 117.

(4) Krausz, F.; Ivanov, M. Attosecond physics. *Reviews of modern physics* **2009**, *81*, 163.

(5) Paul, P.-M.; Toma, E. S.; Breger, P.; Mullot, G.; Augé, F.; Balcou, P.; Muller, H. G.; Agostini, P. Observation of a train of attosecond pulses from high harmonic generation. *Science* **2001**, *292*, 1689–1692.

(6) Griffiths, D. J.; Colleger, R. Introduction to Electrodynamics Prentice Hall Upper Saddle River. *New Jersey* **1999**, *7458*.

(7) Saleh, B. E.; Teich, M. C., *Fundamentals of photonics*; john Wiley & sons: 2019.

(8) Agostini, P.; DiMauro, L. F. The physics of attosecond light pulses. *Reports on progress in physics* **2004**, *67*, 813.

(9) Gürs, K.; Müller, R. Breitband-modulation durch steuerung der emission eines optischen masers (Auskoppelmodulation). *Physics Letters* **1963**, *5*, 179–181.

(10) Steck, D. A. Classical and modern optics. *Oregon University* **2006**.

(11) Diels, J.-C.; Rudolph, W., *Ultrashort laser pulse phenomena*; Elsevier: 2006.

(12) Fushitani, M. Applications of pump-probe spectroscopy. *Annual Reports Section" C"(Physical Chemistry)* **2008**, *104*, 272–297.

(13) Mauritsson, J.; Remetter, T.; Swoboda, M.; Klünder, K.; L'Huillier, A.; Schafer, K.; Ghafur, O.; Kelkensberg, F.; Siu, W.; Johnsson, P., et al. Attosecond electron spectroscopy using a novel interferometric pump-probe technique. *Physical review letters* **2010**, *105*, 053001.

(14) Cheriaux, G.; Chambaret, J.-P. Ultra-short high-intensity laser pulse generation and amplification. *Measurement Science and Technology* **2001**, *12*, 1769.

(15) Gkortsas, V.-M.; Bhardwaj, S.; Lai, C.-J.; Hong, K.-H.; Falcao-Filho, E. L.; Kärtner, F. X. Interplay of mulitphoton and tunneling ionization in short-wavelength-driven high-order harmonic generation. *Physical Review A* **2011**, *84*, 013427.

(16) Kubin, M. Wavelength-tunable xuv-pulses for femtosecond xuv-ir nonlinear effects, Ph.D. Thesis, Master's thesis, Free University Berlin, 2013.

(17) Hareli, L.; Shoulga, G.; Bahabad, A. Phase matching and quasi-phase matching of high-order harmonic generation—a tutorial. *Journal of Physics B: Atomic, Molecular and Optical Physics* **2020**, *53*, 233001.

(18) Keldysh, L. et al. Ionization in the field of a strong electromagnetic wave. *Sov. Phys. JETP* **1965**, *20*, 1307–1314.

(19) Kuchiev, M. Y.; Ostrovsky, V. Quantum theory of high-harmonic generation via above-threshold ionization and stimulated recombination. *Journal of Physics B: Atomic, Molecular and Optical Physics* **1999**, *32*, L189.

(20) Wildenauer, J. Generation of the ninth, eleventh, and fifteenth harmonics of iodine laser radiation. *Journal of applied physics* **1987**, *62*, 41–48.

(21) Augst, S.; Meyerhofer, D. D.; Strickland, D.; Chin, S.-L. Laser ionization of noble gases by Coulomb-barrier suppression. *JOSA B* **1991**, *8*, 858–867.

(22) Lytle, A. L. Phase matching and coherence of high-order harmonic generation in hollow waveguides, Ph.D. Thesis, University of Colorado at Boulder, 2008.

(23) Popmintchev, T. Tunable ultrafast coherent light in the soft and hard X-ray regions of the spectrum: Phase matching of extreme high-order harmonic generation, Ph.D. Thesis, PhD. Thesis, University of Colorado, 2009.

(24) Kramers, H. A. Wellenmechanik und halbzahlige Quantisierung. *Zeitschrift für Physik* **1926**, *39*, 828–840.

(25) Griffiths, D. J.; Schroeter, D. F., *Introduction to quantum mechanics*; Cambridge university press: 2018.

(26) Ammosov, M. V.; Delone, N. B.; Krainov, V. P. Tunnel ionization of complex atoms and of atomic ions in an alternating electromagnetic field. *Soviet Journal of Experimental and Theoretical Physics* **1986**, *64*, 1191.

(27) De Bohan, A.; Piraux, B.; Ponce, L.; Taıeb, R.; Véniard, V.; Maquet, A. Direct and indirect pathways in strong field atomic ionization dynamics. *Physical review letters* **2002**, *89*, 113002.

(28) Ivanov, I.; Kim, K. T. Simple man model in the Heisenberg picture. *Communications Physics* **2020**, *3*, 1–11.

(29) Corkum, P. B. Plasma perspective on strong field multiphoton ionization. *Physical review letters* **1993**, *71*, 1994.

(30) Arfken, G. B.; Weber, H. J. Mathematical methods for physicists, 1999.

(31) Varshalovich, A.; Moskalev, A. N.; Khersonskii, V. K., *Quantum theory of angular momentum*; World Scientific.

(32) Brown, G. G. Attosecond In Situ Measurement and Recombination, Ph.D. Thesis, Université d'Ottawa/University of Ottawa, 2022.

(33) Yamashita, O. Geometrical phase shift of the extrinsic orbital angular momentum density of light propagating in a helically wound optical fiber. *Optics Communications* **2012**, *285*, 3061–3065.

(34) Rundquist, A.; Durfee III, C. G.; Chang, Z.; Herne, C.; Backus, S.; Murnane, M. M.; Kapteyn, H. C. Phase-matched generation of coherent soft X-rays. *Science* **1998**, *280*, 1412–1415.

(35) Hecht, E., *Optics*; Pearson Education India: 2012.

(36) Gouy, L. G., *Sur une propriété nouvelle des ondes lumineuses*; Gauthier-Villars: 1890.

(37) Powers, P. E.; Haus, J. W., *Fundamentals of nonlinear optics*; CRC press: 2017.

(38) Gaarde, M. B.; Salin, F.; Constant, E.; Balcou, P.; Schafer, K.; Kulander, K.; L'Huillier, A. Spatiotemporal separation of high harmonic radiation into two quantum path components. *Physical Review A* **1999**, *59*, 1367.

(39) Balcou, P.; Dederichs, A. S.; Gaarde, M. B.; L'Huillier, A. Quantum-path analysis and phase matching of high-order harmonic generation and high-order frequency mixing processes in strong laser fields. *Journal of Physics B: Atomic, Molecular and Optical Physics* **1999**, *32*, 2973.

(40) Balcou, P.; Salieres, P.; L'Huillier, A.; Lewenstein, M. Generalized phase-matching conditions for high harmonics: The role of field-gradient forces. *Physical Review A* **1997**, *55*, 3204.

(41) Center for X-ray Optics, Lawrence Berkeley National Laboratory Center for X-ray Optics (CXRO) Accessed on Date of Access, `https://cxro.lbl.gov/`.

(42) Kornilov, O.; Wilcox, R.; Gessner, O. Nanograting-based compact vacuum ultraviolet spectrometer and beam profiler for in situ characterization of high-order harmonic generation light sources. *Review of Scientific Instruments* **2010**, *81*, 063109.

(43) André, N.; Fedorov, A.; Chassela, O.; Grigoriev, A.; Le Comte, E.; Rouzaud, J.; Bassas, M. Detection efficiency of microchannel plates to penetrating radiation in space. *CEAS Space Journal* **2019**, *11*, 607–616.

(44) Constant, E.; Garzella, D.; Breger, P.; Mével, E.; Dorrer, C.; Le Blanc, C.; Salin, F.; Agostini, P. Optimizing high harmonic generation in absorbing gases: Model and experiment. *Physical Review Letters* **1999**, *82*, 1668.