A Self-Referenced Timing-Tool for High Repetition X-ray Free-Electron Laser Sources

Dissertation zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Physik
der Universität Hamburg

Vorgelegt von
Michael Diez
aus Hechingen

Hamburg,
2022

Gutachter der Dissertation:	Prof. Dr. Christian Bressler Prof. Dr. Michael Rübhausen
Zusammensetzung der Prüfungskommission:	Prof. Dr. Christian Bressler Dr. Andreas Galler Prof. Dr. Franz Kärtner Prof. Dr. Michael Potthoff Prof. Dr. Michael Rübhausen
Vorsitzende der Prüfungskommission	Prof. Dr. Michael Potthoff
Datum der Disputation:	23.03.2022
Vorsitzender Fach-Promotionsausschusses PHYSIK:	Prof. Dr. Wolfgang Hansen
Leiter des Fachbereichs PHYSIK:	Prof. Dr. Günter H. W. Sigl
Dekan der Fakultät MIN:	Prof. Dr. Heinrich Graener

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich keinen anderen als der im beigefügten Verzeichnis angegebenen Hilfsmittel bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.
Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einverstanden.

Hamburg, den _____ Unterschrift: _____

Abstract

This thesis explores a novel, highly versatile and self-referenced arrival-time detection scheme to measure the timing jitter between ultrashort X-ray pulses and optical laser pulses. These so-called timing-tools are used at free-electron laser facilities to enhance the temporal resolution of pump-probe experiments by measuring the temporal jitter of each optical / X-ray pulse pair. An interaction sample is used to imprint the X-ray arrival-time into a simultaneously propagating optical laser pulse. In contrast to other commonly used timing tools, the major advantage of the self-referenced detection scheme is its increased sensitivity, enabling the use of diamond as interaction sample for the first time. This is an important step towards reliable timing-tools at MHz repetition rate free-electron laser facilities, where the commonly used interaction samples can not withstand the deposited heat load by the intense X-ray pulses.

A detailed simulation in this thesis investigates the deposited heat load in commonly used timing-tool samples and in diamond at high repetition rate facilities, supporting the need for new diamond compatible timing-tools. The expected X-ray-induced refractive index change in diamond is simulated and used to calculate the self-referenced timing-tool signal by propagating an optical pulse through the entire setup, including dispersive effects up to the third order.

The novel self-referenced detection scheme was successfully tested at the European XFEL facility and the SPring-8 Angstrom Compact free electron LAser. At European XFEL, the MHz compatibility of the detection scheme, together with a diamond sample, was demonstrated. The timing jitter at European XFEL was measured with the radio frequency based synchronization (161 fs FWHM) and optical synchronization (90 fs FWHM) schemes. The self-referenced detection scheme was used to determine the X-ray-induced refractive index change in diamond (Δn=-5.7$\times 10^{-5}$).

The high versatility of the detection scheme was demonstrated by exploring the possibility of an in-situ timing-tool at the SPring-8 Angstrom Compact free electron LAser. A free-flowing flat-sheet liquid jet was used as interaction sample, replacing the sample volume for each measurement. The arrival-time measurements of the self-referenced timing-tool are correlated with the facilities own spatial encoding timing tool, and an excellent Pearson correlation of 0.98 is found in the analysis.

Kurzzusammenfassung

Diese Arbeit behandelt eine neue, vielfältig einsetzbare und selbstreferenzierte Methode zur Messung von Schwankungen der Ankunftszeiten von Röntgen- und Laserpulsen. Solche Aufbauten werden "Timing-Tools" genannt und existieren bereits an vielen Freie-Elektronen Lasern. Sie werden dazu genutzt die Zeitauflösung bei Pump-Probe Experimenten zu erhöhen, indem die genaue Ankunftszeit jedes Röntgenpuls / Laserpuls Paares gemessen wird. Dabei wird ein Probenmaterial benutzt, um die Ankunftszeit des Röntgenpulses einem gleichzeitig propagierendem Laserpuls aufzuprägen. Im Gegensatz zu den derzeitig benutzen Timing-Tools, bietet diese hier vorgestellte selbstreferenzierte Methode eine erhöhte Sensitivität, sodass es nun auch möglich ist Diamant als Probenmaterial zu benutzen. Im Hinblick auf die neuen Freie-Elektronen Laser mit hohen Wiederholraten ist dies ein wichtiger Schritt, da konventionell genutzte Probenmaterialien der von den Röntgenpulsen erzeugten Hitze nicht lange widerstehen könnten.
In einer detailreichen Simulation wird die, durch Röntgepulse mit einer hohen Frequenz, erzeugte Hitze in konventionell genutzten Materialien und Diamant analysiert. Diese Simulation unterstützt die Notwendigkeit eines Timing-Tools welches Diamant als Probenmaterial benutzen kann. Die durch die Röntgenpulse verursachte Änderung des Brechungsindex in Diamant wird berechnet und anhand dieser Werte wird das erwartete Signal des selbstreferenzierten Timing-Tools simuliert. Bei dieser Simulation wird ein Laserpuls durch das gesamte Setup propagiert und dispersive Effekte bis zur dritten Ordnung berücksichtig.
Das neue selbstreferenzierte Timing-Tool wurde erfolgreich bei European XFEL und SPring-8 Angstrom Compact free electron LAser getestet. Die Kompatibilität mit hohen Röntgenpuls Wiederholraten in Kombination mit einem Diamant Probenmaterial wurde bei European XFEL festgestellt. Die Schwankungen der Ankunftszeiten bei European XFEL wurde sowohl mit der Radiofrequenz basierten Synchronisation gemessen (161 fs Full Width at Half Maximum (FWHM)), als auch mit der optisch basierten Synchronisation der Anlage (90 fs FWHM). Mit der selbstreferenzierten Detektionsmethode konnte ebenfalls die Änderung des Brechungsindex in Diamant durch den Röntgenpuls gemessen werden (Δn=-5.7$\times 10^{-5}$).
Die hohe Vielfältigkeit der neuen Detektionsmethode wurden bei einem Experiment an SPring-8 Angstrom Compact free electron LAser getestet. Hier wurde ein flacher Wasserstrahl als Timing-Tool Probenmaterial benutzt, welcher das Probenvolumen für jede Messung vollständig austauscht. Die mit der selbstreferenzierten Detektionsmethode gemessenen Ankunftszeiten wurden mit den Ankunftszeiten des Standard Timing-Tool der Einrichtung verglichen und eine Pearson Korrelation von 0.98 gemessen.

Contents

List of Figures

Acronyms

BBO Beta Barium Borate.

BC Birefringent Crystals.

CaF_2 Calcium Fluoride.

CPI Common-Path-Interferometer.

CRLs Compound Refractive Lenses.

CVD Chemical Vapor Deposition.

DESY Deutsches Elektronen-Synchrotron.

EuXFEL European XFEL Facility.

FDE Finite Difference Element.

FLASH Free-Electron LASer in Hamburg.

FWHM Full Width at Half Maximum.

LCLS Linac Coherent Light Source.

OS Optical Synchronization.

PC Polarization Component.

RFS Radio Frequency Synchronization.

SACLA SPring-8 Angstrom Compact free electron LAser.

SF11 Schwer-Flint 11.

Si Silicon.

Si_3N_4 Silicon Nitride.

SLAC Stanford Linear-Accelerator Center.

SNR Signal-to-Noise-Ratio.

XFEL X-ray Free-Electron Laser.

YAG Yttrium Aluminum Garnet.

1. Introduction

Since the beginning of the universe, photons are the primary source of information about our surrounding. For the most part of human history, humankind was only able to passively observe our environment. The technological progress over many centuries led us to the invention of light sources and ultimately gave rise to the invention of coherent light sources [1, 2]. Nowadays, we can control the properties of light emitted by these coherent sources on a large scale, which enables us to investigate matter and even space-time itself in the form of gravitational waves [3]. During the last century, humankind discovered the entire range of the electromagnetic spectrum. Since the discovery of the X-ray region [4] we became able to see the invisible, and its impact since then on material and the life sciences can not be overstated. With the experimental realization of synchrotron radiation in 1947 [5], a powerful source for X-ray experiments with high brilliance was found. The first generation of X-ray synchrotron sources were operated parasitically at storage rings used for high-energy particle physics.
The first facility dedicated to generate synchrotron X-ray radiation opened 1980 in the United Kingdom and was the first facility of the so-called second generation of synchrotron sources [6]. These 2nd generation facilities generated X-ray beams using bending magnets, where electrons from the storage ring were forced on a circular trajectory, emitting a broad X-ray spectrum.
With the wide usage of wigglers and undulators, the first 3rd generation synchrotron facility arose in the early 1990s in France [7]. The 3rd generation synchrotron facilities were, and still are, specifically optimized to use undulators to generate X-ray beams that are orders of magnitudes brighter than the 2nd generation sources. Since then, many dedicated X-ray synchrotron facilities were built worldwide, granting access to a wide audience of researchers to carry out all kinds of experiments at highly specialized experimental stations. The increased access to synchrotron radiation led to a scientific revolution with many breakthrough experiments in several scientific areas, leading to significant contributions worthy of several Nobel Prizes.
Today, we have undulator-based synchrotron facilities in their 3rd generation, producing pulsed, high-quality X-ray beams with a peak brilliance on the order of 10^{21} [photons / s /mm^2 /0.1% bandwidth] for researchers from all over the world.
Despite their undoubtedly high impact, synchrotron sources are reaching their limits for experiments that require high X-ray pulse intensities and ultrashort X-ray pulses. Optical studies in the field of chemistry taught us that the fastest processes in chemical reactions occur on the femtosecond (1 fs = 10^{-15} s) timescale [8]. This timescale is not accessible

with the 50 picosecond (1 ps = 10^{-12} s) long synchrotron X-ray pulses. This was briefly overcome by the inclusion of the so-called time-slicing schemes [9]. Ultrashort fs X-ray pulses can be generated with this method, but in turn, the X-ray flux is greatly reduced. The greatly reduced photon number per pulse made it very challenging to gather a meaningful measurement with enough statistics within a typical one-week-long experiment.
The latest breakthrough in the field of accelerator science and technology enabled the construction of X-ray Free-Electron Laser (XFEL) sources. These sources produce intense ultrashort X-ray pulses with pulse durations in the order of tens of fs. The first operating XFEL in the soft X-ray energy regime (below 5 keV) was the Free-Electron LASer in Hamburg (FLASH) at Deutsches Elektronen-Synchrotron (DESY) in 2005 [10], followed by the first operation of the hard X-ray facility Linac Coherent Light Source (LCLS) at the Stanford Linear-Accelerator Center (SLAC) in 2009 [11]. The sources increased the peak brilliance by over 10 orders of magnitude up to 10^{33} [photons / s /mm^2 /0.1% bandwidth]. With the now available intense and ultrashort X-ray pulses, well-known scientific research fields from the optical wavelength region were adapted into the X-ray domain, e.g., femtochemistry or non-linear X-ray optics.
One of the most fascinating prospects of XFELs is the possibility to create so-called molecular movies for the first time. The atomic structure and electric configuration of atoms, molecules and proteins can be revealed using ultrafast X-ray imaging and spectroscopy methods with tens of fs time-resolution, unraveling all details of a chemical reaction.
Several experiments at XFEL facilities aim to study ultrafast dynamics in pump-probe experiments [12, 13, 14, 15, 16]. In these experiments, time-dependent events are initiated by an intense ultrashort pump laser pulse, mostly from an amplified fs optical laser system. The evolving state of the sample is subsequently probed with an ultrashort fs X-ray pulse. By repeating this for various different temporal offsets between laser-pump and X-ray-probe pulses, the underlying dynamics of the ultrafast process can be recorded. Since both, optical laser and X-ray pulses can be as short as tens of fs (or even shorter with more involved techniques [17, 18]), processes occurring on these timescales could be investigated using X-ray based techniques. Unlike all-optical pump-probe experiments, the pump and probe pulses at XFEL facilities are generated from two different sources. Therefore, both sources need to be synchronized to ensure a stable temporal relationship between the two pulses. All synchronization schemes suffer from unavoidable timing jitter sources, giving rise to uncertainty about the relative arrival-time of both pulses. Depending on the magnitude of the timing jitter between individual pulse pairs, the achievable time-resolution is significantly decreased. To still ensure that both pulses arrive at the sample at the desired temporal delay, sophisticated synchronization schemes have been developed and implemented at XFEL facilities [19, 20, 21]. However, even with those up-to-date synchronization techniques, the pump and probe pulses are still affected by a residual uncertainty of their relative arrival-time, which can accumulate up to hundreds of fs on longer timescales. To fully exploit the short pulse durations, the rela-

tive arrival-times of each pump-probe pulse pair is routinely measured and evaluated to post-process the experimental data at most XFEL facilities. Setups providing such measurements are so-called timing-tools and have been developed over the past 10 years.
The European XFEL Facility (EuXFEL) in Hamburg and Schenefeld is the first XFEL facility delivering intense ultrashort hard X-ray pulses with a 10 Hz burst pattern (pulse trains). Each burst consists of up to 2700 X-ray pulses with an internal repetition rate of 4.5 MHz [22, 23]. These high repetition rates and the combination with X-ray pulse energies of up to 4 mJ can build an enormous heat load on X-ray optics, invasive diagnostic instruments, and experimental samples, even exceeding their individual melting point and damage threshold [24, 25, 26]. This also affects the currently implemented timing-tools at most hard X-ray beamlines at XFEL facilities. Most of them use Silicon Nitride (Si_3N_4) or Yttrium Aluminum Garnet (YAG) as interaction sample to measure the relative X-ray arrival-time. Depending on the X-ray energy, these materials can absorb a substantial amount of the X-ray pulse energy, thermally heating up the sample or destroying them already with the first X-ray pulse. In contrast to the first XFEL facilities operating at repetition rates of 120 Hz or below, the sample materials can not fully thermalize between subsequent X-ray pulses. This leads to a heat pile-up, potentially destroying the sample within a single EuXFEL pulse train.
In this thesis, a MHz compatible timing-tool scheme is presented. The envisioned scheme measures the X-ray-induced refractive index change. This refractive index change is probed by measuring the phase shift of a time-sheared broadband optical pulse with interferometric precision using a Common-Path-Interferometer (CPI). The result is a self-referenced measurement of the X-ray arrival-time, yielding a superior sensitivity compared to conventionally used timing-tools. This enables the use of samples with a smaller X-ray absorption cross-section. With this scheme, it is possible to use diamond as interaction sample. Diamond is one of the few materials able to withstand the challenging conditions of the intense X-ray pulses impinging with a very high repetition rate.

This thesis is outlined into nine chapters. The working principle and challenges of operating an XFEL facility are briefly discussed in the following Chapter 2, using the EuXFEL as an example. An overview of the required fundamental concepts to understand the simulations and analysis in this thesis is given in Chapter 3. In the subsequent Chapter 4, the most commonly used timing-tool schemes with their advantages and disadvantages are explained and presented.
The basic experimental setup for the operation of the newly developed self-referenced timing-tool scheme is explained in Chapter 5.
The results of this thesis are presented and discussed in the following three chapters. Simulations regarding the accumulation of heat in the timing-tool samples at high repetition rate XFELs are discussed in the first part of Chapter 6. In the second part of the simulation chapter, the expected X-ray-induced refractive index change in diamond and the resulting signal and signal strength of the self-referenced timing-tool is discussed.

In Chapter 7 results of the relative X-ray arrival-time at the FXE beamline at EuXFEL are presented, using the self-referenced arrival-time tool with diamond as the interaction sample. The inter-train and intra-train X-ray arrival-time jitter of EuXFEL is analyzed and discussed. In addition, the X-ray-induced refractive index change is extracted from the experimental results, comparing it with the theoretically predicted values from the simulation chapter. In the following Chapter 8, the flexibility of the new self-referenced detection scheme is demonstrated, analyzing and discussing the results of a beamtime at SPring-8 Angstrom Compact free electron LAser (SACLA), where instead of diamond, a free-flowing liquid jet was used to measure the relative X-ray arrival-time in an in-situ compatible measurement mode.
Finally, a summary and conclusion, as well as possible future refinements of the presented experimental, work is given in Chapter 9.

2. X-ray Free-Electron Lasers

Since the past ten years, intense, few-fs hard X-ray pulses are increasingly available at XFEL sources all over the world [11, 27, 28, 29]. With the recent full opening of all six beamlines, EuXFEL is among the newest XFEL facilities. EuXFEL sets itself apart from other facilities by being the first operational facility delivering ultrashort intense hard X-ray pulses at MHz repetition rates, using a liquid helium cooled superconducting linear accelerator. EuXFEL is capable of providing the broadest range of X-ray energies, from soft X-rays to above 20 keV hard X-ray radiation [22, 23]. Being a state-of-the-art facility, EuXFEL will be used in this Chapter to explain the principle of an XFEL and its related subsystems. The main sources for a timing jitter between optical pulses and X-ray pulses are discussed. Finally, the synchronization scheme of EuXFEL is described.

2.1. Free-Electron Laser Principle

The layout of all XFEL facilities can be broken down into four basic sections (Fig. 2.1). The first section contains the injector, where an electron pulse (bunch) is generated and receives its first acceleration. The second section is the main accelerator, where the generated electron bunches are accelerated to ultra-relativistic energies with up to 17.5 GeV. The third subsection are the undulators, generating the X-ray pulses from the electron bunches. These X-ray pulses are then guided to the last section, where the experimental stations are located. An in-depth description of all aspects of the functionality of an XFEL facility would be beyond the scope of this work. Instead, a brief description of the most important aspects relevant for this thesis are presented. A full description can be found in [30] for EuXFEL in particular and in [31] and [32] for general properties of XFEL sources.

Using the example of EuXFEL, the general layout of an XFEL facility (with minor changes depending on each facility) is shown in Fig. 2.1. The first step is to generate a well-defined electron bunch via photoelectric emission by a pulsed UV laser impinging on a Cs_2Te cathode in the electron gun. Following the electron gun, a short linear accelerator (booster) is used to promptly accelerate the electron pulse to 120 MeV [24]. The electron bunch is further guided towards the main linear accelerator, passing through the 3rd harmonic module and the laser heater. When accelerating an electron bunch, not all electrons are accelerated to the exact same speed (or kinetic energy) due to an initial spread of electron energies and Coulomb forces between the electrons [24] [33]. This yields a spatial and temporal spread of the electron bunch. The 3rd harmonic module improves the

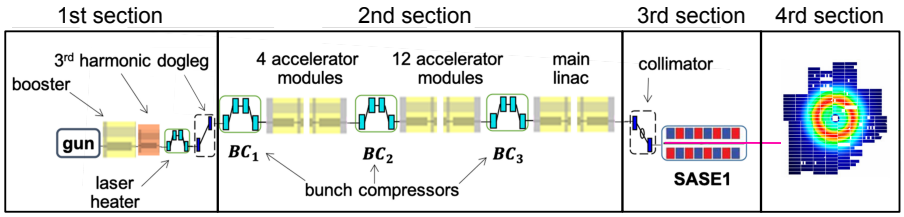

Figure 2.1.: Layout of the EuXFEL facility and the hard X-ray SASE1 undulator branch. Figure adapted from [30].

longitudinal phase-space properties of the electron bunches [34] while the laser heater prevents longitudinal microbunching instabilities in the electron beam [35]. The bunch-compressors are magnetic chicanes that compress the electron bunch based on the energy chirp within the electron pulse. Equivalent to an optical grating or prism compressor, the electron pulse is compressed by forcing electrons with different energies to travel over slightly different path lengths. Leading electrons with high energy travel a longer way through the chicane than trailing electrons with less energy. Hence, the trailing electrons of the electron bunch catch up to the leading electrons, and longitudinal bunch compression is achieved [36]. After a series of shorter linear accelerator modules and additional bunch compressor chicanes, the compressed electron bunch enters the main 2 km long linear accelerator. All linear accelerator modules at EuXFEL are superconducting nine-cell radio-frequency (RF) cavities, developed within the TESLA project [37]. Each cavity is He-cooled and operated at their designed resonance frequency of 1.3 GHz. The accelerating electric field points in beam direction on the cavity axis and is using RF excitation of the TEM_{101} cavity mode. The length of a single cavity cell is designed such that an electron traveling with relativistic velocities passes each cavity cell in half the period of the 1.3 GHz accelerating frequency. The zero crossings of the accelerating RF field is set to be at the entrance and exit of each cavity cell. By using multiple cavity cells, a standing wave pattern arises with alternating electric fields from one cell to the other. Thus, an electron experiences a forward accelerating electric field in each cavity cell. The cavities are powered by several klystrons. Together with the low-level RF control system, the phase and amplitude of the accelerating electric field can be controlled. At EuXFEL the electrons can be accelerated to a maximal energy of 17.5 GeV.

To generate the X-ray pulses from the accelerated and longitudinally compressed electron bunches, the electron bunches are propagated through an undulator (Fig. 2.2). The undulator itself is an arrangement of a sandwich of periodically placed permanent magnets with alternating polarity of the magnetic field B_u. The distance between two neighbouring equal poles is the undulator period λ_u and is typically in the order of a few cm. To generate hard X-ray pulses ($\geq$ 5 keV) at EuXFEL, an undulator period of 4 cm is used with an on-axis peak magnetic field of 1.2 T and a total magnetic length of 175 m, divided into 5 m long undulator modules [38]. The magnetic field can be tuned by widening or closing the undulator gap, separating the top and bottom half of the undulator. The al-

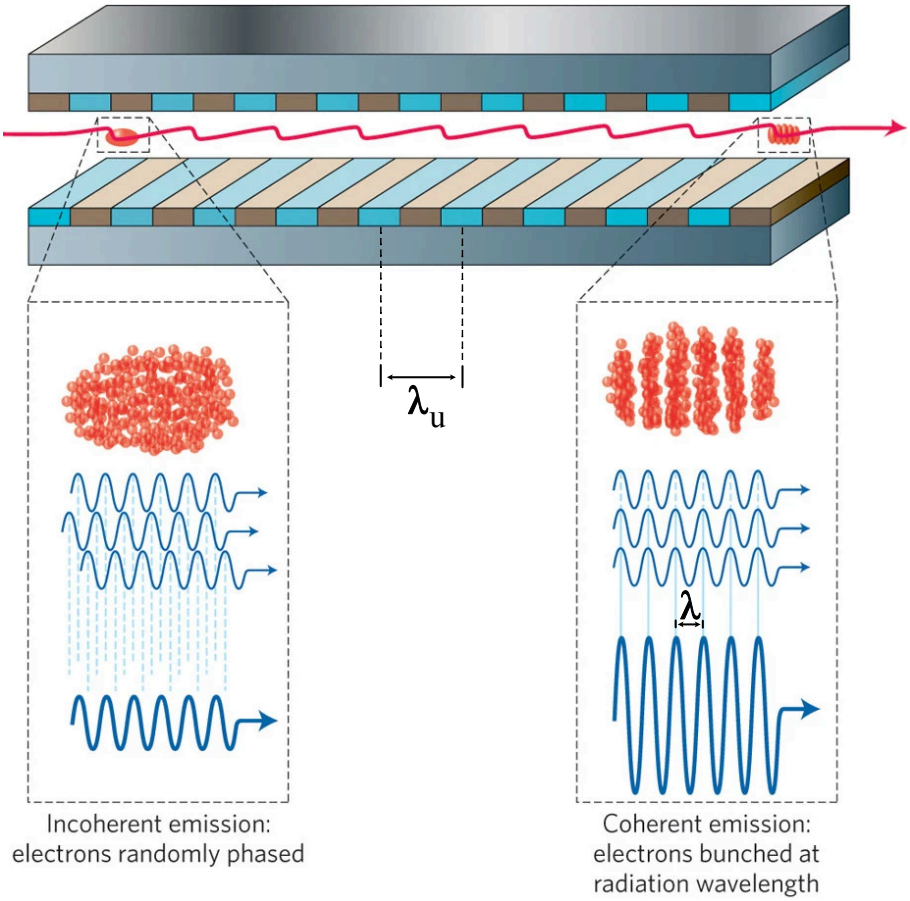

Figure 2.2.: Schematic of an undulator. The electron beam enters the undulator on the left and starts to oscillate transversely as it propagates through the undulator. In the first section, electrons are randomly ordered and emit incoherent radiation. Towards the end of the undulator the electrons formed microbunches within the electron bunch, and the electron bunch emits coherent radiation with wavelength λ. Figure adapted from [31].

ternating arrangement of the magnets and the resulting magnetic field in the undulator are forcing the electrons onto a sinusoidal path through the undulator. Due to the forced sinusoidal trajectory, the electrons are nearly permanently accelerated, emitting X-ray photons in a narrow cone along the undulator axis. The wavelength λ of the emitted photons is described by

$$\lambda = \frac{\lambda_u}{2\gamma_{rel}^2}\left(1 + \frac{K_u^2}{2}\right) \qquad \text{with} \qquad K_u = \frac{eB_u\lambda_u}{2\pi m_e c_0}\,, \tag{2.1}$$

where K_u is the undulator parameter and γ_{rel} the Lorentz factor $\gamma_{rel} = \frac{1}{1-\beta^2}$ with $\beta = v/c_0$ [32]. The wavelength of the emitted photons can be manipulated by changing the magnetic field B_u through variations of the undulator gap size or by changing the kinetic electron energy.

In the first section of the undulator, the generated radiation is incoherent due to the

stochastic process of spontaneous emission and the electrons in the electron bunch are randomly arranged (Fig. 2.2). The electron bunch and emitted light field co-propagate through the undulator. The electrons interact with the co-propagating electromagnetic field, losing or gaining energy to it. Electrons losing their energy to the light field travel a longer path on their sinusoidal trajectory, while electrons gaining energy, travel a shorter path. This entails a density modulation of the electron bunch, ultimately leading to thin slices of electrons, separated by the wavelength of the electromagnetic field (Fig. 2.2 bottom right). This process is called microbunching. These microbunches radiate in phase, and the emitted light adds up coherently, which causes a quadratic increase of the emitted intensity. In turn, this increases the effect of microbunching, forcing more electrons to emit coherently. The entire process from the first incoherently emitted radiation to the formation of microbunches, where the incoherent radiation is serving as seed for the followed coherent emission, is called Self-Amplified Spontaneous Emission (SASE) [39].
Followed by the undulator section, a strong permanent magnet separates the electron beam from the SASE generated coherent X-ray pulses. The X-ray pulses are then subsequently guided to the different six experimental endstations over a set of X-ray mirrors.

2.2. Sources of Timing Jitter

The operation of an XFEL facility requires precise synchronization of all subsystems. Even with cutting-edge technology, it is not possible to reliably deliver X-ray and optical pulses without any relative arrival-time jitter. In this Subsection, the main contributing sources of this timing jitter are discussed.

2.2.1. X-ray Source Related Timing Jitter

The arrival-time stability of the X-ray pulses at the experimental endstations are influenced by the timing stability of the electron bunches. The SASE process itself leads to additional statistical timing fluctuations due to the nature of the process but is mostly negligible for hard X-ray pulses since the statistical timing jitter is less than 1 fs [40].
The timing jitter of the electron bunches is directly related to the quality of the electron beam. Not only temporal jitter sources like the precise timing of the injector laser are affecting the temporal electron bunch stability, but also fluctuations in the electron beam energy. These are caused by amplitude fluctuations of the accelerating fields in the cavities or by fluctuations in the phase of the beam, i.e., the phase of the accelerating field in the cavities or the phase of the injector laser. Electron bunch energy fluctuations are directly translated into a temporal timing jitter by the bunch compressor chicanes because the total path length changes for different electron energies [20] [41]. These instabilities can be greatly reduced by advanced synchronization schemes to reduce phase and timing variations of the injector laser and the driving electric fields in the cavities, ultimately reducing the overall phase, amplitude, and timing jitter of the electron beam dramatically.

Such synchronization schemes are discussed in Section 2.3. The EuXFEL design goals require an amplitude deviation of the acceleration RF field below 0.01% which would translate in an arrival-time jitter of the electron bunch of smaller than 33 fs (RMS) and a phase stability of better than 0.01° which results in a maximum additional timing jitter of 21 fs (RMS) [24].

2.2.2. Optical Laser Source Related Timing Jitter

Although the optical laser is a much smaller machine, these systems are cutting-edge technology with inherent timing jitter fluctuations. The main concern for the timing stability of optical pulses at a large scale facility, is the propagation of the laser pulses from the laser source to the actual experiment. Usually, the laser source is located in a remote laser laboratory with very precise temperature and climate control systems. From there, the optical pulses are distributed to the facility experimental endstations [42] [43]. The beampath from the laser laboratory to the actual experiment can easily extend to several tens of meters without dedicated climate-controlled systems. This can lead to pointing instabilities as well as timing jitter due to temperature gradients, either in air, affecting the refractive index, or temperature related expansion of optics and optomechanics [44]. These beam propagation related timing instabilities are very hard to avoid, and without taking careful measures, the laser timing jitter introduced alone by the beam transportation can be as high as several tens of fs. With very careful beam propagation management, i.e., having large parts of the beam propagation under vacuum conditions, using relay imaging and only using state-of-the-art vibration dampening optomechanics, this number can be pushed to be well below 10 fs [45].
Timing jitter sources related directly to the laser source are more difficult to tackle and are always dependent on the actually used laser system. In general, laser oscillators are known to be nearly unharmed by their intrinsic temporal jitter, which usually is well below 1 fs [46]. They can also be synchronized to other laser oscillators with very low timing errors (see section 2.3). Major timing uncertainties can be introduced by amplifying these oscillator pulses with common schemes such as Chirped Pulse Amplification (CPA) [47] or Optical Parametric Chirped-Pulse Amplification (OPCPA) [48], which is used at the EuXFEL in-house developed pump-probe laser system [42]. Both amplification schemes can introduce substantial amount of timing jitter, up to hundreds of fs.
In a Ti:sapphire based CPA regenerative amplifier, short-term timing drifts of 10 fs RMS and around 200 fs on longer timescales were reported [49]. These long-term fluctuations are mostly attributed to thermal expansion effects, i.e., the stainless steel baseplate of 1 m length of a multipass amplifier expands approximately 3 µm at a temperature change of 0.2 °C, which can add up to a total path length difference of 40-60 µm, depending on the number of round trips in the amplifier cavity. Another source of introducing hundreds of fs of jitter, especially in OPCPA systems, are timing differences between the pump and seed pulses. Depending on the OPCPA scheme, especially grating based pulse compres-

sion is susceptible of introducing a timing jitter, which can surpass hundreds of fs [50]. The same optical locking schemes which are used to optically synchronize XFEL facilities can be used to greatly reduce these timing jitter effects in optical OPCPA based laser system to less than 50 fs RMS [44].
If all those effects are carefully treated and minimized by state-of-the-art technology (beam transport, laser internal synchronization schemes), each effect is only adding small amounts of timing jitter of the optical laser pulses, but summed up, they are large enough to affect the temporal resolution of X-ray based pump-probe experiments.

2.3. Synchronization of the European XFEL Facility

To deal with the various sources of timing jitter at large scale facilities, synchronization schemes are implemented with the goal to limit the timing jitter and timing drifts of all sources as much as possible. Nowadays, there are mainly two synchronization schemes used, one relying on the distribution of RF signals and the other relying on the distribution of optical laser pulses. A basic synchronization scheme of the EuXFEL, which uses a combination of both schemes, is shown in Fig. 2.3.

The synchronization at EuXFEL is based on a redundant system of GPS disciplined oscillators, the so-called RF-Main-Oscillator (RF-MO), with very high timing stability and a 10^{-12} frequency stability [52]. The phase stable reference signal is produced at 1.3 GHz and is distributed via coaxial cables to the injector and accelerator subsystems. In direct vicinity to the RF-MO, two main laser oscillators (MLO) are placed, which are directly synchronized to the RF-MO. These two MLOs are passively mode-locked semiconductor saturable absorber mirror based laser oscillators, delivering laser pulses with a central wavelength of 1553 nm and pulse duration of around 100 fs. The repetition rate is set to the 6th harmonic (216.667 MHz) of the RF-MO reference signal, such that the MLO can be directly synchronized to the reference signal [19]. The optical reference signal from the MLO is distributed across the entire facility by using polarization maintaining

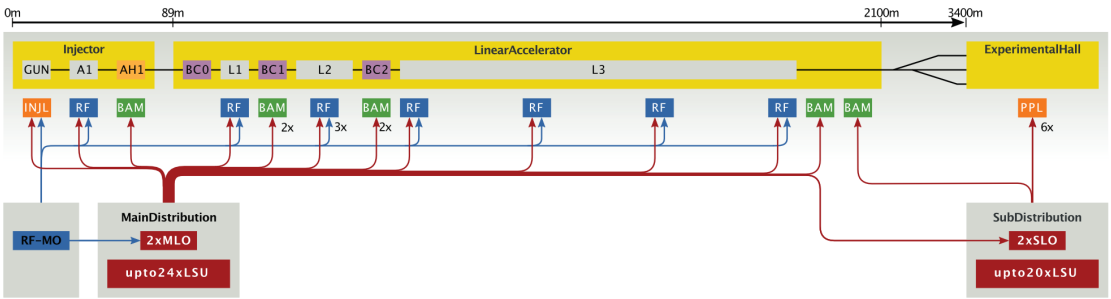

Figure 2.3.: Synchronization scheme of EuXFEL. The RF-MO signal distribution is shown in blue. The stabilized optical synchronization scheme is shown in red. Figure from [51]

fibers. The transportation of time reference signals over large distances via coaxial cable or optical fibers is very challenging. Especially coaxial cables suffer from huge attenuation when transporting high frequency signals and suffer from timing drifts caused by the thermal expansion of the cable, which can be as high as 30 fs/K/m at room temperature [53]. Optical fibers also suffer from thermal expansion, but optical fibers for the telecommunication wavelength are engineered for extremely low losses at this particular wavelength. Ultimately, the low losses in optical transmission and the relative low cost of optical fibers, enable a fs-precise synchronization of remote devices over large distances. The most important advantage of distributing optical timing reference signals, is the possibility to use the short optical pulses for optical cross-correlation measurements.
The distribution of the optical reference signal is done by so-called timing stabilized optical fiber links. For the optical reference distribution, the pulses of the MLO are split into, up to 24, individual link stabilization units (LSUs) by using polarizing beam cubes and waveplates [20] [51]. Each LSU is connected to a subsystem of the accelerator with an individual optical fiber. The optical fibers are terminated by a partly reflecting end-facet. Thus, a part of the transported optical pulse is reflected back to the LSU. In the LSU, the back-reflected optical pulse and a new pulse from the MLO are cross-correlated in a balanced optical cross-correlator (OXC) [54]. With a mechanical delay line and a fiber stretcher, the detected timing change can be actively corrected and stabilized, such that the arrival-time jitter of the optical reference pulse can be below 1 fs (RMS) [20].
At EuXFEL only two (for redundancy) stabilized optical links are sending a reference timing signal to the experimental hall. It is cheaper to operate a second laser oscillator in the experimental hall, synchronized to the stabilized optical signal, than pulling multiple km long optical fibers all the way to the experimental hall. The optical pulses of the subsidiary laser oscillator (SLO) in the experimental hall are split into several LSUs and are delivered to the dedicated experimental stations and laser laboratories.
Some components of the facility, especially the accelerating cavities, require the use of an RF timing signal. Cross-correlation techniques exist for RF based signals, but they are not nearly as precise as optical cross-correlation techniques. At EuXFEL, a laser-to-RF-phase detector is used to synchronize the RF timing signal to the simulatenously delivered optical time reference signal at the location of the subsystem. In comparison to the all-optical synchronization, an additional timing jitter of 2.1 fs (RMS) is measured when using the laser-to-RF-phase detection scheme [55].
For pump-probe experiments, the synchronization of the optical laser to the X-ray source is particularly important. EuXFEL is using a hybrid synchronization scheme for the optical pump-probe laser. The laser can be directly synchronized via a balanced two color OXC, yielding the lowest possible timing jitter between both devices [54, 20]. If the relative delay between the optical laser and X-ray pulses need to be shifted with respect to each other, the path length of the optical pulses need to be physically adjusted. To achieve large temporal shifts between the optical and X-ray pulses, the pump-probe laser

system can also be synchronized by creating an RF signal from the optical timing signal. The extraction of an RF signal from an optical pulse train generates additional timing jitter [56]. By synchronizing the pump-probe laser to this RF signal, it is easy to use a phaseshifter for larger timing adjustments. This mode of synchronization is especially important when setting up the machine after a shutdown or after major timing changes. Whenever this RF based synchronization of the optical laser is used throughout this thesis, it will be called Radio Frequency Synchronization (RFS). For experiments where no ultra-precise timing resolution between the optical and X-ray pulses is necessary, it is advantageous to use the RF synchronization because it is faster and more flexible to adjust. For experiments which require the highest possible time resolution, the optical laser must be synchronized directly to the Optical Synchronization (OS).

3. Fundamental Concepts

In this chapter, the required fundamental concepts to fully describe and analyze the presented self-referenced timing-tool approach are discussed. Fundamental diamond properties are introduced, followed by the interaction of X-ray photons with matter and a special focus on diamond. The last part of this chapter outlines fundamental principles of optics and ultrashort lasers pulses.

3.1. Diamond Properties

Diamond is used as a target sample in the newly developed detection scheme for X-ray arrival-time measurements presented in this thesis. In contrast to currently used materials, diamond is suited to withstand high repetition rates and high-intensity XFEL pulses. The thermal dynamics in diamond after the absorption of an intense X-ray pulse are simulated in Chapter 6.1, while the transient refractive index change after the absorption of an X-ray pulse is described and simulated in Chapter 6.2. These simulations require a brief introduction of the solid-state properties of diamond.
Diamond is one of the most unique materials occurring in nature and possesses extreme physical properties [57, 58, 59], setting it apart from all other solids. It is the hardest, least compressible and stiffest material occurring in nature. It is an incredible thermal conductor (more than 4 times better than copper), has a very low thermal expansion coefficient and is resistant to most chemical acids. In addition, it is transparent from the deep UV region into the infrared region. While natural diamond exhibits all those properties, they are rarely all present in the same stone due to impurities or crystal defects. The rarity of natural diamonds and the even higher rarity of diamonds with the desired combination of all desired physical properties led to the commercial production of synthetic diamonds by high-pressure and high-temperature processes or by Chemical Vapor Deposition (CVD) [60]. Impurities in CVD diamond are mainly nitrogen or boron atoms, contributing to a doping mechanism where a carbon atom is replaced in the atomic lattice. These undesired atoms are incorporated into the synthetic crystal due to almost unpreventable contamination of the CVD reaction chamber. Over the years, the understanding of the synthesis of CVD diamonds reached a level to produce electric grade CVD diamonds, which are nowadays used for applications where the purity and dopant level of materials is of high importance, such as the research on quantum devices [61]. In these CVD diamonds the concentration of nitrogen and boron atoms is smaller than 5 ppb [62].

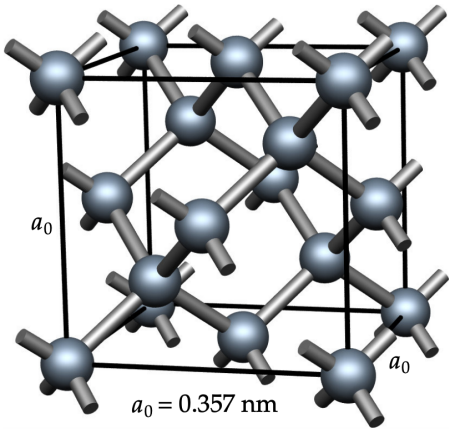

Figure 3.1.: Diamond unit cell with the cubic lattice parameter a_0. Image from [64].

Carbon in its ground state has a $1s^2 2s^2 2p^2$ electronic configuration. When forming a diamond crystal, the s orbital overlaps with the p orbitals to form covalent bonds in a tetrahedral geometry by sp^3 hybridization [63]. The resulting atomic structure of the crystal can be described by two inter-penetrating face-centered cubic (FCC) lattices (Fig. 3.1). The two lattices are shifted with respect to each other by a fraction of $1/4$ of the FCC lattice parameter a_0=0.357 nm. The bond length between two carbon atoms is 0.154 nm.
The crystal lattice and overlapping atomic orbitals give rise to the electronic band structure E($\boldsymbol{k}$) [65], describing the allowed electron (and hole) energies of the crystal. Diamond is a wide-bandgap insulator with an indirect bandgap of 5.47 eV and a direct bandgap of 7.3 eV at the Γ point. An important property to describe effects related to the band structure of a material is the effective mass m^* that is used in the so-called effective mass approximation [66]. It is used to describe the dispersion of an electron (or hole) near the extrema of the band structure. The effective mass is defined as a tensor

$$\left(\frac{1}{m^*}\right)_{i,j} = \frac{1}{\hbar^2}\frac{\partial^2 E(\boldsymbol{k})}{\partial k_i \partial k_j} \,, \tag{3.1}$$

where $(1/m^*)_{i,j}$ is the effective mass tensor, describing the curvature near the bottom of the conduction band or the top of the valence band of the band structure. In the most simple case of parabolic energy bands with spherical constant energy surfaces, the effective mass is a scalar and the band structure around its minima (or maxima) is described in analogy to a free moving electron by

$$E(\boldsymbol{k}) = E_0(\boldsymbol{k_0}) + \frac{\hbar^2 \boldsymbol{k}^2}{2m^*} \,, \tag{3.2}$$

with the addition of the energy offset $E_0(\boldsymbol{k_0})$ to account for the energy level offset of the band minima (or maxima) and the effective mass m^*, which is used to alter the curvature to fit with the band structure. For most semiconductors, the bands can be approximated as parabolic, but the energy surfaces are ellipsoids.

For this thesis, another important property is the birefringence of a material. Despite the isotropic crystal structure, CVD diamonds are often very weakly birefringent. This is caused by growth defects that cause strain on the crystal lattice. Nowadays, CVD diamonds with a birefringence as low as $\Delta n = 5 \times 10^{-7}$ can be produced [67]. This order of birefringence in diamond is small enough to not affect the expected signal of a diamond based timing-tool (see Section 6.2.3).

3.2. X-ray Interaction with matter

X-ray photons can interact with matter in different processes [68]. At lower X-ray energies, the dominant interaction process is the photoelectric effect with its cross-section σ_{pe}. The incoming X-ray photon is absorbed in this process by a bound electron. At higher photon energies scattering processes are becoming more dominant. These scattering processes are split into coherent scattering with a cross-section of σ_{cs} and incoherent scattering with a cross-section σ_{is}. Coherent scattering processes deflect the incident photon elastically off of atoms. Incoherent scattering transfers parts of the photon energy to an electron, potentially leading to its ionization. The pair-production process σ_{pp} starts to dominate at very high energies, where a positron-electron pair is generated. The incoming photon energy must exceed the rest energy $2m_e c^2$ of the positron-electron pair and the generation can only occur in proximity to a nucleus (or electron at even higher energies) due to momentum conservation. The total cross-section is the sum of all contributions $\sigma_{\mathrm{tot}} = \rho_{\mathrm{pe}} + \sigma_{\mathrm{cs}} + \sigma_{\mathrm{is}} + \sigma_{\mathrm{pp}}$. These cross-sections are shown for carbon in Fig. 3.2. The energy regimes of the dominating cross-sections can be identified: up to about 2×10^4 eV the photoelectric effect dominates, from 2×10^4 eV to 3×10^7 eV incoherent scattering is dominating and above 3×10^7 eV the pair production in the nuclear field is the dominating process. Photon energies in the range from $\sim$100 eV to $\sim$150 keV can be readily reached by some modern 3rd generation X-ray facilities [69]. The gray area indicates the X-ray photon energies covered by the EuXFEL hard X-ray beamlines [23]. In this energy region, the photoelectric threshold energies of all elements are accessed, and many elements (Z=8-47) have their K-shells in this energy range.

With the atomic number density n of the interacting matter, the total X-ray attenuation coefficient is described as $\mu_{tot} = n\sigma_{tot}$. The attenuation of an X-ray beam with intensity I_0 can be calculated with the Lambert-Beer-Law [71] by the equation

$$\begin{aligned} I(d) &= I_0\, e^{-\mu_{tot} d} \\ &= I_0\, e^{-n\sigma_{tot} d}\ , \end{aligned} \tag{3.3}$$

where d is the thickness of the material.

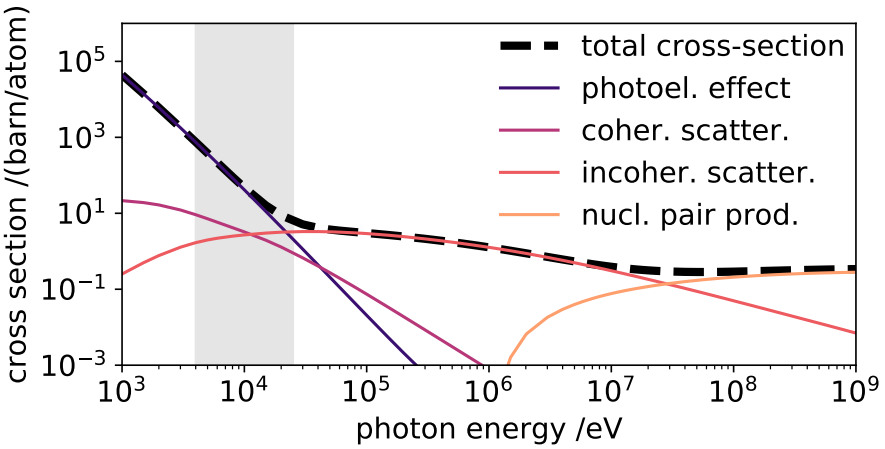

Figure 3.2.: X-ray interaction with matter cross-sections of carbon. The gray shaded part is the energy region covered by the hard X-ray beamlines of EuXFEL. Data from [70].

3.2.1. X-ray-induced Electron Cascades in Diamond

As discussed before, the photoelectric effect is the dominant interaction between light and matter at X-ray photon energies covered by XFEL facilities. Inelastic scattering processes of the X-ray photons will be neglected in the following discussion.
Photoabsorption of an X-ray photon excites an electron from the valence band or deep atomic shells to a high energetic continuum state. Hereby, the photoelectric cross-section for inner-shell excitations is the largest. The deep-shell holes subsequently decay via X-ray emission or Auger processes, during which another electron is ejected into the conduction band [72]. For elements Z<30, Auger decay dominates, while the radiative decay dominates for high Z elements [73]. The K-shell fluorescence yield is almost negligible with 0.35%. The electron-electron impact ionization is the main driving process for the secondary electron generation, but also Auger-like processes play a significant role [74]. Experiments show that the electron-electron pair creation energy to promote an electron from the valence band to the conduction band is above the bandgap energy [75, 76]. A reason for the increased pair creation energy is that the impact ionization process is restricted by energy and momentum conservation [77]. Therefore, they often occur as indirect transition, thus, the direct bandgap is often not a relevant parameter [78]. In addition, electrons are not only located in the highest state of the valence band and the lowest state of the conduction band [78]. This additionally increases the required pair creation energy. The pair creation energy in many semiconductors and insulators was experimentally measured, and it was found that the pair creation energy is roughly three times the bandgap of a material [76].
For an initially generated high-energy electron, many impact ionization events are required to reduce the energy of the initial electron to below the pair creation threshold. At the same time, electrons transfer their excess energy via electron-phonon coupling pro-

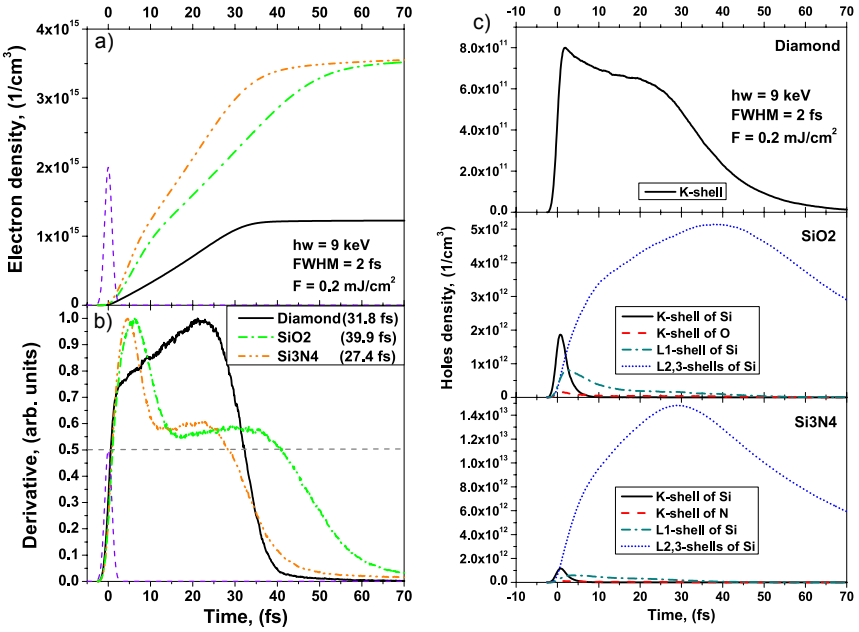

Figure 3.3.: XCASCADE simulation of dynamic electron densities in diamond, SiO_2 and Si_3N_4. Dynamics for 9 keV, 2fs FWHM, 0.2 mJ/cm^2 X-ray pulse is shown in a). The time derivative of the dynamic electron densities are depicted in b). On the right-hand side in c), the simulated core hole density in different shells is illustrated for diamond (top), SiO_2 (middle) and Si_3N_4 (bottom). Figure adapted from [81]

cesses to the lattice [77] [79].

To be able to describe the expected signal of the self-referenced timing-tool approach, it is important to have an understanding of the timescale on which the electron cascade is occurring and the amount of secondary electrons generated from a high energetic photoelectron. Simulations on this topic are done with the XCASCADE Monte Carlo toolkit that models X-ray-induced electron cascades [80]. XCASCADE models the kinetics of electrons with boundary conditions in a given volume of material. The material is modeled as a uniform homogenous arrangement of the materials atoms with its corresponding density. All relevant atomic energy levels and material-wide energetic properties, including valence and conduction bands, as well as the bandgap, are implemented. The relevant processes for the fs-scale cascading effect are incorporated, i.e., photoabsorption by core or valence shells, secondary electron impact ionization and Auger decay of core-shell holes. An excited electron in the conduction band is considered a free moving particle and the time until its next collision is calculated according to the Poisson distribution of its mean free path and velocity. Effects on longer timescales such as electron-phonon coupling are neglected.

For the simulation of the self-referenced measurement approach, two questions need to be answered. How many secondary electrons are created after absorption of one X-ray photon with a certain energy and on what timescale does the full electron cascading finish? For the materials used in this thesis, diamond and Si_3N_4, electron cascades after the

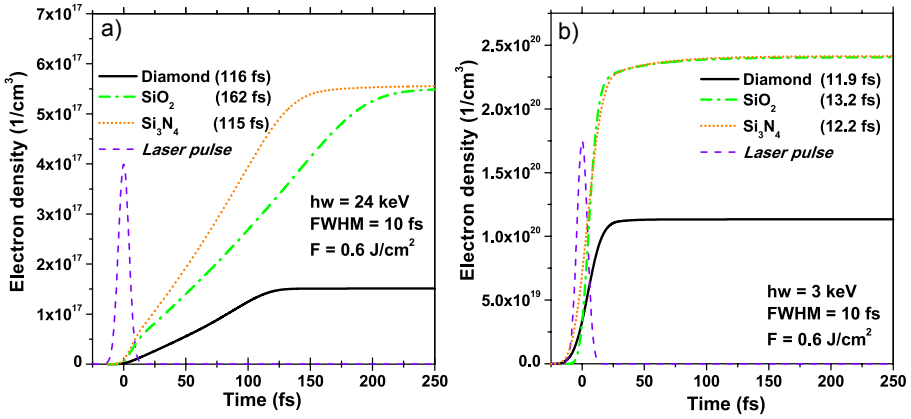

Figure 3.4.: XCASCADE simulation for electron density after an X-ray pulse with 10 fs pulse duration and 0.6 J/cm^2 fluence for two different X-ray photon energies in diamond, SiO_2 and Si_3N_4. a) shows the simulation for 24 keV, b) illustrated the increase in electron density after an X-ray pulse with 3 keV photon energy. Figure adapted from [81]

absorption of an X-ray photon are already simulated with XCASCADE and reported [81] (Fig. 3.3). These simulations illustrate the time-dependent evolution of the electron density of diamond, SiO_2 and Si_3N_4 after irradiation with an X-ray pulse with 2 fs FWHM pulse duration, photon energy of 9keV and fluence of 0.2 mJ/cm^2 (Fig. 3.3 a). During the initial 2 fs long X-ray pulse, a prompt increase of the electron density in all three materials can be observed, but this contribution to the total electron density after 70 fs remains very low (Fig. 3.3 b). The initial sharp increase is attributed to the direct photoabsorption of X-ray photons during the X-ray pulse. After the first initial increase, the dynamics evolve differently in each material. The simplest dynamic can be observed in diamond. After the initial sharp increase, the slope of the derivative of the electron density dynamic flattens due to the secondary electron cascading events and Auger-decay of K-shell holes of the carbon atoms (20-25 fs). After 30 fs, the increase in electron density slows down because most of the cascading electrons have already lost too much energy to remain above the pair creation energy threshold, and the electron cascade ends.

For SiO_2 and Si_3N_4, the cascading dynamics and increase in electron density is more complex because more atomic shells are involved in the cascading. The initial sharp increase is dominated by the photoabsorption by the K-shells of the Si atoms. These K-shell core holes decay very fast after several fs. On longer timescales (<20 fs), L_1 shell holes undergo Koster-Cronig or Auger transitions [72]. The $L_{2,3}$ shell holes decay on longer timescales of tens of fs. The interplay of the decay of the various core holes lead to the more complicated electron density evolution for SiO_2 and Si_3N_4 (Fig. 3.3 c).

The temporal evolution of the electron densities for two different X-ray photon energies are different (Fig. 3.4). X-ray pulses with a pulse duration of 10 fs FWHM and a fluence of 0.6 J/cm^2 were used to simulate the induced electron densities. The simulated X-ray photon energies are 24 keV (Fig. 3.4 a) and 3 keV (Fig. 3.4 b). For the 24 keV X-ray pulse, the

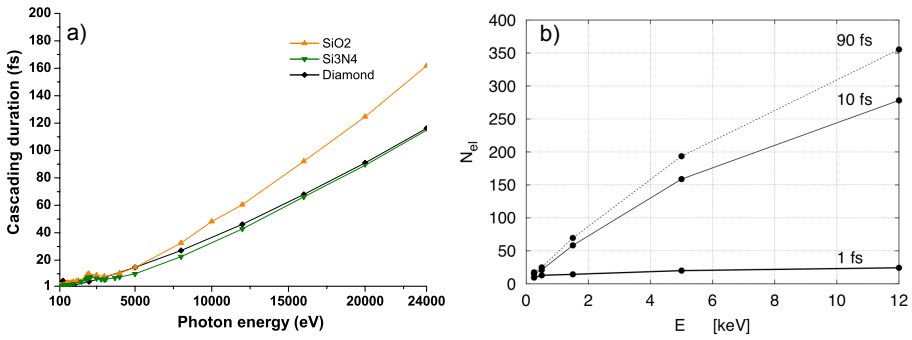

Figure 3.5.: Subfigure a) illustrates the FWHM electron cascade times for diamond, SiO_2 and Si_3N_4 from 100 eV to 24 keV. Subfigure b) shows how many secondary electrons are generated after absorption of one X-ray photon, after 1fs, 10fs and 90 fs, for different X-ray photon energies. Subfigure a) adapted from [81], subfigure b) from [74]

electron density keeps increasing after an initial sharp increase during the X-ray pulse. The increase in secondary electrons flattens for all three materials until reaching the final electron density. The cascading times are 162 fs for SiO_2, 116 fs for diamond and 115 fs for Si_3N_4. The initial photoelectron is most probably generated by photoabsorption in the K-shell, thus, generating a photoelectron with an initial energy of 23-24 keV. A substantial amount of collisions, generating secondary electrons, are needed to decrease the energy of the initial photoelectron below the ionization threshold. The duration of the electron cascade depends on the photoelectron initial kinetic energy and the amount of energy it loses by collision ionization. The electron density dynamics for an X-ray pulse with only 3 keV photons happens on faster timescales. The increase of electron density during the actual X-ray pulse is sharp again but flattens out fast after the X-ray pulse and reaches its final state. The cascading times are only 13 fs for SiO_2 and 12 fs for diamond and Si_3N_4. This can be attributed to substantially fewer needed collisions to decrease the initial photoelectron kinetic energy to below the pair creation energy threshold.
The differences in the final electron densities in each material after an X-ray pulse with 24 keV and 3 keV photon energy can be explained by the X-ray absorption cross-section of the materials.
The FWHM electron cascading times for the three materials are illustrated by Medvedev in Fig. 3.5 a) for different X-ray photon energies. As expected, cascading times are increasing with higher X-ray energies because more collision ionization events are needed to decrease the kinetic energy of the initial photoelectron below the pair creation energy threshold. In subfigure 3.5 b), Ziaja et al. calculated the number of secondary electrons generated by one photoelectron after the absorption of an X-ray photon for different X-ray photon energies after 1 fs, 10 fs and 90 fs.

3.3. Dielectric Properties of Materials

To simulate the X-ray-induced refractive index change (Chapter 7.5) in a dielectric material, the fundamentals of electrodynamics are introduced in this Section. The Drude model and Maxwell Garnett theory of dielectric mixtures are presented to calculate the X-ray-induced refractive index change.

3.3.1. Fundamental Electrodynamic Properties

The theory of electrodynamics is one of the most successful theories in physics. Many optical experiments can be purely described by using Maxwell's equations [82, 83]. Maxwell's four equations describe how electric charges and currents give rise to electric and magnetic fields and how these fields evolve over time. The four Maxwell equations are [84]

$$\boldsymbol{\nabla} \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{3.4a}$$

$$\boldsymbol{\nabla} \times \boldsymbol{H} = \frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{j} \tag{3.4b}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{D} = \rho \tag{3.4c}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0\ , \tag{3.4d}$$

where the $\boldsymbol{E}(\boldsymbol{r}, t)$ is the electric field strength and $\boldsymbol{H}(\boldsymbol{r}, t)$ the magnetic field strength. The quantities $\boldsymbol{D}$ and $\boldsymbol{B}$ are the electric and magnetic flux densities, respectively. $\boldsymbol{j}$ is the electric current density and ρ the volume charge density of any external charges and are the sources of the electromagnetic field. The response of a macroscopic material to an external or internal electric or magnetic field is given by the material equations:

$$\boldsymbol{D} = \epsilon_0 \boldsymbol{E} + \boldsymbol{P} \tag{3.5a}$$

$$\boldsymbol{B} = \mu_0 \left(\boldsymbol{H} + \boldsymbol{M}\right) \tag{3.5b}$$

$\boldsymbol{P}$ is the induced electric polarization density and $\boldsymbol{M}$ the induced magnetization density. The other constants are the vacuum permittivity ϵ_0 and vacuum permeability μ_0.
In free space ($\boldsymbol{P} = 0$, $\boldsymbol{M} = 0$, $\boldsymbol{j} = 0$), the four Maxwell equations are simplified to:

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{3.6a}$$

$$\nabla \times \boldsymbol{B} = \epsilon_0 \mu_0 \frac{\partial\, E}{\partial t} \tag{3.6b}$$

$$\nabla \cdot \boldsymbol{E} = 0 \tag{3.6c}$$

$$\nabla \cdot \boldsymbol{B} = 0\ , \tag{3.6d}$$

The propagation of an electromagnetic wave through free space is described by the wave equation, which can be derived by taking the curl of Eq. 3.6a and Eq. 3.6b:

$$\nabla \times (\nabla \times \boldsymbol{E}) = \nabla(\nabla \boldsymbol{E}) - \nabla^2 \boldsymbol{E} = -\frac{\partial}{\partial t}(\nabla \times \boldsymbol{B}) = -\epsilon_0 \mu_0 \frac{\partial^2 \boldsymbol{E}}{\partial t^2} \tag{3.7a}$$

$$\nabla \times (\nabla \times \boldsymbol{B}) = \nabla(\nabla \boldsymbol{B}) - \nabla^2 \boldsymbol{B} = \epsilon_0 \mu_0 \frac{\partial}{\partial t}(\nabla \times \boldsymbol{E}) = -\epsilon_0 \mu_0 \frac{\partial^2 \boldsymbol{B}}{\partial t^2} \tag{3.7b}$$

Using $\nabla \boldsymbol{E} = \nabla \boldsymbol{B} = 0$, the wave equations are obtained

$$\left(\nabla^2 - \frac{1}{c_0^2}\frac{\partial^2}{\partial t^2}\right)\boldsymbol{E} = 0 \tag{3.8a}$$

$$\left(\nabla^2 - \frac{1}{c_0^2}\frac{\partial^2}{\partial t^2}\right)\boldsymbol{B} = 0\ , \tag{3.8b}$$

with the vacuum speed light $c_0 = \frac{1}{\sqrt{\epsilon_0 \mu_0}}$.
In the context of optics, mostly a non-magnetic, linear, homogeneous and isotropic dielectric medium with no free charges or currents can be assumed. In such a medium, the material equations are

$$\boldsymbol{D} = \epsilon_0 \epsilon_r \boldsymbol{E} \tag{3.9}$$

$$\boldsymbol{B} = \mu_0 \mu_r \boldsymbol{H}\ , \tag{3.10}$$

with the relative permittivity ϵ_r and relative permeability μ_r. The permittivity and permeability are defined as

$$\epsilon = \epsilon_0(1 + \chi_e) = \epsilon_0 \epsilon_r \tag{3.11}$$

$$\mu = \mu_0(1 + \chi_m) = \mu_0 \mu_r\ , \tag{3.12}$$

with the electric and magnetic susceptibilities χ_e and χ_m, respectively. The relative permittivity ϵ_r and relative permeability μ_r are frequency dependent material properties. In dielectric materials, the permittivity ϵ and permeability μ is used in Eq. 3.8a and 3.8b, instead of only using the vacuum permittivity ϵ_0 and vacuum permeability μ_0. The speed of light in a dielectric medium is then defined as:

$$c = \frac{1}{\sqrt{\epsilon \mu}} \tag{3.13}$$

The ratio of the vacuum speed of light to that in the medium, c_0/c is defined as refractive index:

$$n = \frac{c_0}{c} = \sqrt{\frac{\epsilon}{\epsilon_0}\frac{\mu}{\mu_0}} = \sqrt{\epsilon_r \mu_r} \tag{3.14}$$

Most materials are non-magnetic at optical frequencies, such that $\mu_r = 1$ and the refractive index becomes simply $n = \sqrt{\epsilon_r}$. This relation will be used in this thesis (Section 6.2) to calculate the expected X-ray-induced refractive index change in diamond.
The wave equations for the electric (Eq. 3.8a) and magnetic (Eq. 3.8b) field in a linear medium are solved by real monochromatic plane-waves. Simple solutions for the wave equation can be obtained when restricting $\boldsymbol{E}$ to depend only on one coordinate, e.g., the z-coordinate, such that

$$\frac{\partial \boldsymbol{E}}{\partial x} = \frac{\partial \boldsymbol{E}}{\partial y} = 0 \,. \tag{3.15}$$

For this special case, the resulting wave has only the components E_x and E_y. The general solution for plane waves are:

$$E_x(z,t) = f_x(z-ct) + g_x(z+ct) \tag{3.16}$$
$$E_y(z,t) = f_y(z-ct) + g_y(z+ct) \tag{3.17}$$

The two functions f and g are arbitrary, but continuously differentiable and represent plane waves because z=const are areas of constant phase. For periodic plane waves with the wavelength λ, $f(z-ct)$ can be written as $f(z+\lambda-ct)$. For periodic waves, the ansatz

$$\boldsymbol{E} = \boldsymbol{E}_0 \cdot f(z-ct) = \boldsymbol{E}_0 \cdot \cos\left[k(z-ct)\right] \tag{3.18}$$

can be used. With the requirement for periodicity, the constant k can be identified as

$$k = \frac{2\pi}{\lambda} \tag{3.19}$$

and is called the wave number. Hence, Eq. 3.18 can be rewritten to

$$\boldsymbol{E} = \boldsymbol{E}_0 \cdot \cos(kz - \omega t) \,. \tag{3.20}$$

For plane waves propagating in an arbitrary direction a vector $\boldsymbol{k} = \{k_x, k_y, k_z\}$ is defined, pointing into the direction of the propagating wave. This vector is called wave vector with the amount

$$|\boldsymbol{k}| = k = \frac{2\pi}{\lambda} \,. \tag{3.21}$$

For practical reasons complex numbers are used often times to express propagating waves. The electric field $\boldsymbol{E}(\boldsymbol{r},t)$ then reads as

$$\boldsymbol{E} = \boldsymbol{E}_0 \cdot \mathrm{e}^{i(\boldsymbol{k}\boldsymbol{r}-\omega t)} + \text{c.c.} \tag{3.22}$$

with the amplitude $\boldsymbol{E}_0 = \{E_{0,x}, E_{0,y}, E_{0,z}\}$ and where c.c. stands for the complex conjugate.
The electric field, propagating along the z-axis can be described as the superposition of two electric field components $\boldsymbol{E}_{0,x}$ and $\boldsymbol{E}_{0,y}$. Both complex amplitudes can have different amplitudes and phases. The electric field resulting from these two components is defined with

$$\boldsymbol{E} = E_{0,x} \cdot \mathrm{e}^{i(kz-\omega t+\phi_x)} + E_{0,y} \cdot \mathrm{e}^{i(kz-\omega t+\phi_y)} \tag{3.23}$$

with ϕ_x and ϕ_y being the phase of each field component. If the phase difference between ϕ_x and ϕ_y equals zero or multiples of π, the linear combination of both components result in an electric field, oscillating in a constant plane. In this case the resulting electric field is called linear polarized and the polarization angle is defined by

$$\tan\alpha = \frac{|E_{0,x}|}{|E_{0,y}|} \ . \tag{3.24}$$

A phase offset of exactly $\frac{\pi}{2}$ between ϕ_x and ϕ_y results in a circular polarization of the electric field. For all other phase differences, the resulting polarization is elliptical with varying lengths of the major and minor axis.

3.3.2. The Drude Model

The optical properties of solids are closely related to the electronic properties of a material. The dielectric function ϵ_r is influenced by free carriers. These free carriers are often responsible for ultrafast electrical changes in materials after irradiance with ultrashort optical or X-ray pulses [66]. A simple model to estimate the contribution of free electrons is a model based on classical conductivity, called the Drude model [85]. This plasma model is based on a gas of free electrons. In this approach, details of the lattice potential and electron-electron interactions are not included. Aspects of the band structure are incorporated by using the effective electron mass m_e^* (see Chapter 3.1). The model replicates the properties of metals and metal-like materials such as degenerate semiconductors rather well.
An external optical electric field is driving electron oscillations within the material. The electron oscillations and their movement are dampened by collisions between the electrons with a mean free collision time τ. The collision frequency is $\gamma = \frac{1}{\tau}$. When applying the external driving optical electric field $\boldsymbol{E} = \boldsymbol{E_0}\mathrm{e}^{-i\omega t}$, the trajectory of a single electron is described by:

$$m_e^* \frac{\partial^2 \boldsymbol{r}}{\partial t^2} + m_e^* \gamma \frac{\partial \boldsymbol{r}}{\partial t} = -e\boldsymbol{E}_0 e^{-i\omega t} \ , \tag{3.25}$$

A particular solution to the differential equation above is $\boldsymbol{r}(t) = \boldsymbol{r}_0 \mathrm{e}^{-i\omega t}$ and Eq. 3.25 becomes:

$$\boldsymbol{r} = \frac{e}{m_e^*(\omega^2 + i\gamma\omega)}\boldsymbol{E} \tag{3.26}$$

The moving electrons create a macroscopic polarization density $\boldsymbol{P} = -Ne\boldsymbol{r}$, with the electron density N. With Eq. 3.26, the polarization density is given by:

$$\boldsymbol{P} = -\frac{Ne^2}{m_e^*(\omega^2 + i\gamma\omega)}\boldsymbol{E} \tag{3.27}$$

Inserting the relation for the polarization density $\boldsymbol{P}$ into (3.5a) yields

$$\boldsymbol{D} = \epsilon_0 \left(1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega}\right)\boldsymbol{E}\ , \tag{3.28}$$

with the plasma frequency $\omega_p^2 = \frac{Ne^2}{\epsilon_0 m_e^*}$. With Eq. 3.9, the complex dielectric function of a free electron gas is identified as:

$$\epsilon(\omega) = 1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} \tag{3.29}$$

$$= 1 - \left(\frac{\omega_p}{\omega}\right)^2 \frac{1}{1 + i/(\omega\tau)} \tag{3.30}$$

This dielectric function can be used to calculate the refractive index by using Eq. 3.14. To incorporate the contribution of, to the nuclei bound, electrons to the polarizability, often times the first term is changed to ϵ_∞. By using the mean free electron collision time τ, the dielectric function reads as:

$$\epsilon(\omega) = \epsilon_\infty - \left(\frac{\omega_p}{\omega}\right)^2 \frac{1}{1 + i/(\omega\tau)} \tag{3.31}$$

ϵ_∞ can be interpreted as the ground state dielectric function of the material, which is altered by the additionally added free electron density of the second term. The additional free electrons in the material are reducing the polarizability of the material, thus, the overall dielectric function decreases.

3.3.3. Maxwell Garnett Dielectric Mixtures

A more advanced model to describe the influence of an additionally created electron density in a dielectric medium is to use effective medium approximations. In 1904, Maxwell Garnett developed the Maxwell Garnett theory of dielectric mixtures to describe colorful effects in metal-doped glasses [86, 87]. The theory describes a medium consisting of various different dielectric components as a mixture of these dielectrics and unifies them to an effective medium with an effective dielectric constant.

In contrast to the Drude model described above, the effective medium approach incorporates a coupling of the free electrons with the dielectric host material. The free electrons interact instantaneously with the bound electrons of the host material, distorting their equilibrium distribution, creating a polarization of the crystal lattice. This self-induced electronic polarization cloud screens the charge of the free electron and moves along with it through the crystal. The electron moving through the crystal, the polarization cloud surrounding it, and the distortion to the bound electrons, together form a quasiparticle called electronic polaron [88, 89].
Here, we use these polarons in the framework of an effective medium approximation to describe the effective dielectric function $\epsilon_{eff}(\omega)$ of a dielectric mixture of an isotropic host material with dielectric function $\epsilon_h(\omega)$ and polarons with a dielectric function $\epsilon_p(\omega)$. The polarons can be assumed as spherical particles homogeneously suspended in the host material. In this case, the relationship between the dielectric function of the host and suspended particles is given by [90]

$$\frac{\epsilon_h - \epsilon_{eff}}{\epsilon_h + 2\epsilon_{eff}} = l \cdot \frac{\epsilon_p - \epsilon_{eff}}{\epsilon_p + 2\epsilon_{eff}} \,, \tag{3.32}$$

with $l = 4\pi N_p r_p^3/3$ being the volume fraction of the suspended particles. N_p is the polaron density and r_p the polaron radius. Solving Eq. 3.32 gives the effective dielectric function of the dielectric mixture:

$$\epsilon_{eff} = \epsilon_h \cdot \frac{2\epsilon_h + \epsilon_p - 2l[\epsilon_h - \epsilon_p]}{2\epsilon_h + \epsilon_p + l[\epsilon_h - \epsilon_p]} \tag{3.33}$$

This formula is used in Section 6.2 to estimate the X-ray-induced refractive index change in diamond.

3.4. Ultra-short Laser Pulses

For the simulation of the self-referenced arrival-time signal in Section 6.2, it is necessary to understand the fundamentals of ultra-short laser pulses. Here, important optical properties and the physics of ultrashort laser pulses and their linear propagation through dispersive media is briefly described. The mathematical description is based on [91, 92, 93, 94] and [95].

3.4.1. Wavepackets

Ultrashort laser pulses are wave packets consisting of a continuous superposition of monochromatic plane waves. This superposition can be achieved by mode-locking [96, 97] longitudinal modes of a laser resonator, creating a fixed phase relation between all modes.

The oscillating electric field of a laser pulse at a fixed point in space can be written as

$$E(t) = A(t) \cdot e^{i(\omega_0 t - \phi(t))} + c.c. \tag{3.34}$$

with the time-dependent amplitude (carrier envelope) $A(t)$, the central frequency ω_0 and time-dependent phase $\phi(t)$. For many applications it is convenient to mathematically treat the laser pulse not in the time domain, but in the spectral domain. Both domains are connected through the Fourier inversion theorem:

$$E(\omega) = \mathfrak{F}\{E(t)\} = \int_{-\infty}^{\infty} E(t)\, e^{-i\omega t}\, \mathrm{d}t \tag{3.35a}$$

$$E(t) = \mathfrak{F}\{E(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} E(\omega)\, e^{i\omega t} \mathrm{d}\omega \tag{3.35b}$$

Since only the real part of the electric field $E(t)$ is used, the complex spectrum $E(\omega)$ is Hermitian which means

$$E(\omega) = E^*(-\omega)\ . \tag{3.36}$$

Fourier transforming Eq. 3.34 give two symmetrical frequency distributions around zero, which means there are also negative frequencies contributing. Because of the hermitian properties of $E(\omega)$, only respecting positive frequencies is sufficient to fully describe $E(\omega)$, and therefore also $E(t)$:

$$E^+(t) = \frac{1}{2\pi} \int_0^{\infty} E^+(\omega)\, e^{i\omega t}\, \mathrm{d}\omega \tag{3.37}$$

with

$$E^+(\omega) = \begin{cases} E(\omega) & \text{for } \omega \geq 0 \\ 0 & \text{for } \omega < 0 \end{cases} \tag{3.38}$$

The total intensity $|E(t)|^2$ in the time domain and total spectral density $|E(\omega)|^2$ in the frequency domain can be measured with suitable detectors.
The shortest achievable pulse duration is correlated to the number of modelocked resonator modes. The higher the number of resonator modes a pulse contains, the shorter the pulse can be in the temporal domain. This can be shown for a Gaussian-shaped finite function $x(t)$. For a signal with its center in time and frequency at $t_0 = 0$ and $\omega_0 = 0$, the time and frequency variance is given by:

$$\sigma_t^2 = \frac{\int t^2 \left|x(t)\right|^2 \mathrm{d}t}{\int \left|E(t)\right|^2 \mathrm{d}t} \tag{3.39a}$$

$$\sigma_\omega^2 = \frac{\int \omega^2 \left|E(\omega)\right|^2 \mathrm{d}\omega}{\int \left|E(\omega)\right|^2 \mathrm{d}\omega} \tag{3.39b}$$

With the Schwarz Inequality, the above can be written as

$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4} \tag{3.40}$$

which is in analogy to the quantum mechanical uncertainty principle. This means that a certain bandwidth of the laser spectrum is required to be able to get an ultrashort laser pulse. More often Eq. 3.40 is written as

$$\Delta\tau \cdot \Delta\nu \geq K \tag{3.41}$$

and is referred to as the time-bandwidth-product (TBP). In this case $\Delta\tau$ is the pulse duration and $\Delta\nu$ the width of the spectrum, both measured at their FWHM values. The constant K depends on the shape of the carrier envelope of the laser pulse. The most used pulse shapes to model an ultrashort pulse are a Gaussian and Hyperbolic Secant pulse shape. For these two the K value is the following:

$$\text{Gaussian: } K = 0.441$$
$$\text{Sech}^2: \; K = 0.315$$

A laser pulse where the TBP is exactly the K value is called a bandwidth-limited pulse and can not be temporally shorter with the given bandwidth and pulse shape. In such a pulse the instantaneous frequency

$$\omega(t) = \omega_0 - \frac{\partial\phi(t)}{\partial t} \tag{3.42}$$

is constant over the total time evolution of the laser pulse.
The interplay between all the described properties in this Chapter are shown in Fig. 3.6. In all those cases, an ultrashort bandwidth-limited pulse with a central frequency of 375 THz ($\approx$ 800 nm) and FWHM bandwidth of 95 THz ($\approx$ 200 nm) is illustrated. Fig. 3.6 a) illustrates the spectral density $|E(\nu)|^2$ of the electric field in the frequency domain. The exact same electric field is shown in b) but in the temporal domain. The instantaneous frequency over the full pulse is constant, indicating a bandwidth-limited pulse.

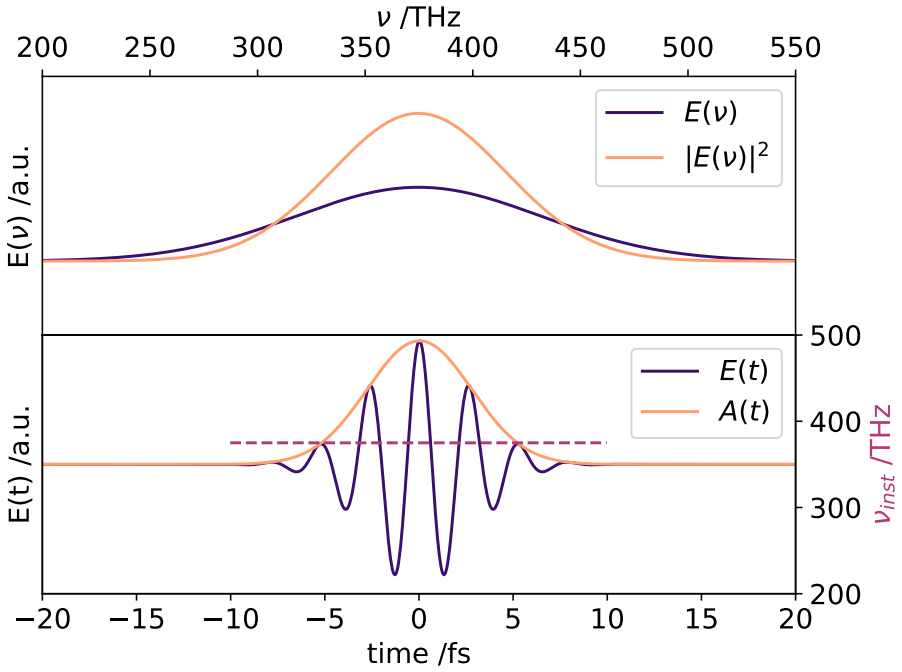

Figure 3.6.: Electric field of a bandwidth-limited laser pulse in spectral a) and time domain b).

Linear Pulse Propagation

Ultrashort laser pulses are heavily affected by propagation effects. These occur when the pulse is transmitted through any dispersive medium. The reason for these effects is the frequency dependent refractive index $n(\omega)$. Due to the variation of the refractive index for different frequencies, an ultrashort laser pulse travelling through any material is affected by dispersive effects. Empirically the refractive index of a material can be described via the Sellmaier Equation [98]

$$n^2(\lambda) = 1 + \sum_i \frac{B_i \lambda}{\lambda^2 - C_i} \,, \tag{3.43}$$

where λ is the wavelength and B_i and C_i are experimentally determined values called Sellmaier coefficients. Hence, the wavenumber $k = \frac{2\pi}{\lambda} = \frac{n\omega}{c}$ is also frequency dependent:

$$k(\omega) = \frac{n(\omega)\,\omega}{c} \,, \tag{3.44}$$

Here, the frequency-dependent refractive index is now also included. This introduces dispersive effects when a laser pulse travels through matter. The superposition of modelocked monochromatic waves with different frequencies will be changed since the individual frequencies will not travel at the same speed.
A very convenient way to describe these dispersive effects is by using linear system theory, which describes elements that influence the spectral amplitude and spectral phase in

a linear fashion. In general, this can be done in the time domain or spectral domain. The transmitted pulse $E_{out}(t)$ and $E_{out}(\omega)$ can be described with

$$E_{out}(t) = h(t) * E_{in}(t) \tag{3.45a}$$

$$E_{out}(\omega) = h(\omega)\, E_{in}(\omega) \tag{3.45b}$$

where $h(t)$ and $h(\omega)$ are the system impulse response and the system transfer functions. The linear propagation in the time domain is defined through a convolution of the incoming electric field and its impulse response function, whereas in the spectral domain it is defined through a simple multiplication. Therefore, it is more convenient to describe the propagation of a laser pulse in the spectral domain.
For all considerations a bandwidth-limited ultrashort pulse, as described in Eq. 3.35a with a Gaussian envelope for $E_{in}(\omega)$, is considered. The linear propagation in the spectral domain of a short pulse traveling along the z-axis is described as:

$$E_{out}(\omega, z) = E_{in}(\omega, 0)\, H(\omega, z)\, e^{-ik(\omega)z} \tag{3.46}$$

$H(\omega)$ is the real valued spectral amplitude response and $e^{-ik(\omega)z}$ is the phase transfer. After propagation, the spectral amplitude, as well as the spectral phase of the original pulse $E_{in}(\omega)$ is altered. The phase can be expressed in terms of the coefficients of a Taylor series expansion:

$$k(\omega) = k_0 + \frac{\partial k}{\partial \omega}(\omega - \omega_0) + \frac{1}{2}\frac{\partial^2 k}{\partial \omega^2}(\omega - \omega_0)^2 + \frac{1}{6}\frac{\partial^3 k}{\partial \omega^3}(\omega - \omega_0)^3 + ... \tag{3.47}$$

The first term represents the absolute phase of the pulse and describes the relative phase between the slowly varying amplitude and the rapid field oscillations. The second term is the linear phase term. Through the Fourier shift-theorem it is easily acceptable that a phase term linear in frequency shifts the entire pulse along the time axis in the time domain. Hence, this term describes the time a pulse needs to travel through the dispersive material and can be used to determine its group velocity. The term with the second derivative is usually referred to as the group velocity dispersion (GVD) and describes the temporal broadening of the pulse since the different monochromatic waves of the pulse travel with different group velocities through the dispersive material. For most materials, the refractive index increases with higher frequencies ($\frac{dn}{d\omega} > 0$) and leads to lower frequency traveling faster through the material than higher frequencies of the pulse. Such behaviour is called normal dispersion and will result in a linear positive chirped pulse, where the instantaneous frequency over time is increasing. For anomlous materials ($\frac{dn}{d\omega} < 0$), the instantaneous frequency is decreasing when a broadband laser pulse is transmitted through the material, and the pulse is called negatively chirped. All even order Taylor coefficients affect the pulse envelope in a symmetrical fashion, while the odd number coefficients introduce an asymmetric change of the pulse envelope.

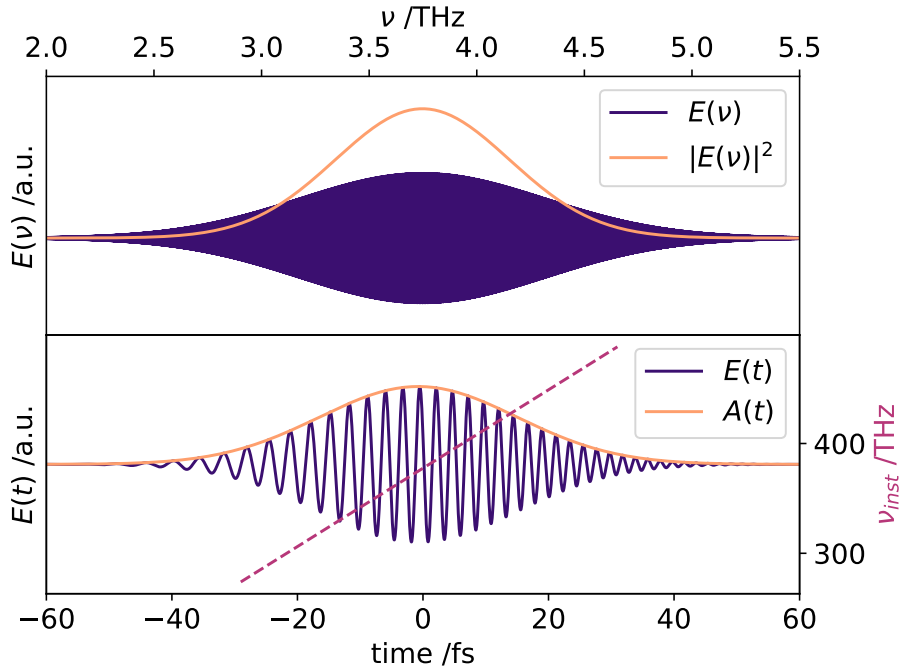

Figure 3.7.: Electric field of a linearly propagated bandwidth-limited laser pulse through dispersive material. Electric field of the transmitted pulse in the spectral domain a) and time domain b).

In Fig. 3.7, the bandwidth-limited pulse from Fig. 3.6, is linearly propagated through 1.5 mm of SiO_2 glass. Dispersive effects up to TOD are taken into account. Subfigure a) illustrates the measurable spectrum $|E(\omega)|^2$ and the spectral electric field $E(\omega)$. Note that the spectrum which can be measured is still the same as before the pulse was propagated. In subfigure b), the same pulse is shown in the time domain. Here, the added chirp of the laser pulse can be observed, ordering all frequencies in time from lower to higher frequencies. The instantaneous frequency is increasing linear over time and is indicated by the purple dashed line.

4. Review of X-ray Relative Arrival-Time Measurements

As described in Chapter 2, the operation of an XFEL facility is quite complex and requires a perfect interplay of many sub-systems. Despite all the cutting-edge technologies, there are still limitations in the performance of these machines. One critical limitation is the timing jitter between the XFEL X-ray source and synchronized optical laser sources. To fully exploit the ultrashort pulses of both sources, additional tools are required to enhance the time resolution of pump-probe experiments. These so-called timing-tools measure the exact relative arrival-time of each X-ray / optical pulse pair. After the experiment, the data can be used to post-process the experimental data and increase the time resolution. Different methods of measuring the relative arrival-times are discussed in this Chapter.

4.1. Electro-Optical Sampling

The first XFEL related timing diagnostic experiment was reported in the year 1999 and measured the longitudinal electric field of electron bunches on a sub-ps timescale, using the Pockels effect in electro-optic crystals [99]. Originally, the method was commonly used to measure transient electric fields and THz pulses from laser-driven sources and is known as Electro-Optic Sampling (EOS) [100] [101].
A second milestone arrival-time measurement of electron bunches of an FEL accelerator was reported by Cavalieri et al. in 2005 [102]. The measurements were done at SLAC and paved the way for all following XFEL timing diagnostics. To measure the electron bunch arrival-time, they placed an electro-optic crystal (ZnTe) in close proximity to the electron beam. Each time an electron bunch passes the sample, its refractive index is anisotropically distorted by the strong electromagnetic field of the ultrarelativistic electron bunch. The transient change of the crystal refractive index was then probed with a synchronized fs laser pulse, propagating through the electro-optic crystal at the exact same time as the electron bunch passes the crystal.
Today, most XFEL facilities are using various EOS schemes to measure the timing jitter of the electron bunches. Measuring the electron bunch arrival-time can give valuable insights into the operation and stability of the accelerator. The fact that the electron bunch arrival-time measurements are usually performed far away from the actual X-ray experiment makes them less effective for post-experiment timing jitter corrections. Therefore, following this breakthrough experiments, methods directly measuring the relative X-ray

arrival-time in close proximity to the actual experiment were developed.

4.2. Optical Streaking

Inspired by the principle of ordinary streak cameras [103], the THz streaking method was pioneered at FLASH [104]. Like the original streak camera in 1971, the THz streak camera is also used to measure light pulses, but this time it is used to investigate the pulse duration and time structure of individual soft X-ray pulses. Getting detailed information about the relative X-ray arrival-time are a side product of the X-ray pulse characterization with this method [105].
For the streaking experiments, a few-cycle THz pulse is generated with a synchronized optical laser pulse. These THz pulses are mostly generated using the tilted pulse-front pumping method in a $LiNbO_3$ crystal, generating THz pulses with variable central frequencies, ranging from 0.1 THz to 1 THz [106, 107]. In the reported experiments, THz pulses with a central frequency of 0.5 THz [108] and 0.6 THz [105, 109] were used, yielding a period of 2 ps and 1.67 ps, respectively. The THz pulse and X-ray pulse are spatially and temporally overlapped in a noble gas (He or Xe). The X-ray pulse ionizes the noble gas molecules, generating photoelectrons whose kinetic energy spectrum are analyzed with an electron-time-of-flight (eTOF) spectrometer. Depending on the relative arrival-time of the ultrashort X-ray pulse and the much longer few-cycle THz pulse, the generated photoelectrons interact with the THz external electric field, and their kinetic energies are shifted (Fig. 4.1).
To reliably measure the relative arrival-time, the X-ray pulses are required to temporally overlap with the THz pulse in the region around the zero-crossing, where the slope of the electric field is nearly linear. In this region, the energetic shift of the measured photoelectron kinetic energy spectrum correlates linearly with the relative arrival-time of the X-ray pulse. This limits the measurement window to about 0.5 ps for the THz streaking method at these THz frequencies.
Using the noble gases as interaction sample has the huge advantage of being nearly non-invasive. The photoionization cross-section decreases with increasing photon energies, thus, fewer photoelectrons are generated, and the measured kinetic energy spectrum decreases. Therefore, THz streaking schemes are most often used at soft X-ray beamlines. With a more complicated setup, using large Xe gas clusters generated by a supersonic pulse valve, enough photoelectrons can be generated even at hard X-ray photon energies [108]. Another disadvantage of the THz streaking is the rather complicated setup which requires the generation of few-cycle THz pulses and a vacuum interaction chamber with the noble gas.
The second noteworthy streaking method is the angular streaking [110, 111] approach. As with the THz streaking, X-ray-generated photoelectrons from noble gases are streaked with a laser pulse. This time, the laser pulse is a circular polarized IR pulse. The wave-

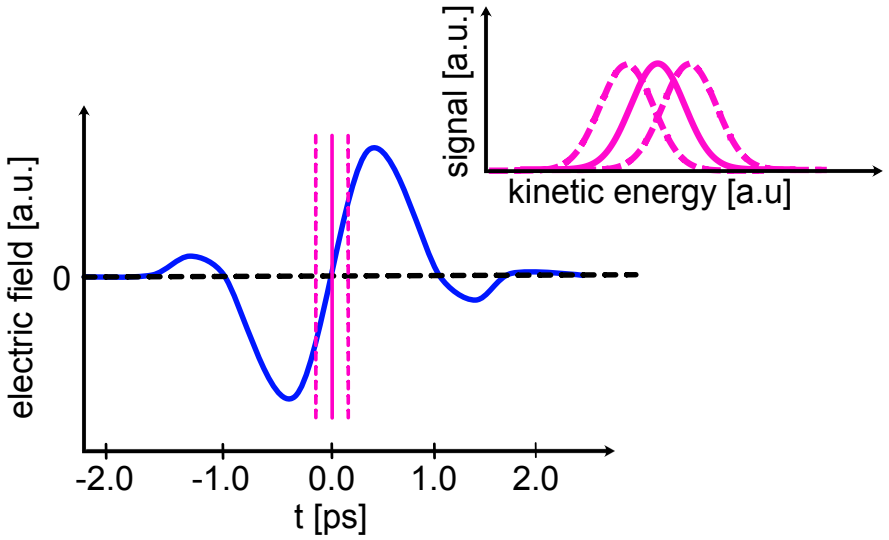

Figure 4.1.: THz streaking principle. A single-cycle THz pulse (blue) is temporally overlapped with an ultrashort intense X-ray pulse (pink). The delay between the THz and X-ray pulses is set to overlap with the linear slope region of the THz pulse. The kinetic energy spectrum of the X-ray-generated photoelectrons is measured with an eTOF spectrometer (top right). Depending on the relative arrival-time of the X-ray pulse (dashed pink lines), the photoelectrons are streaked, and their kinetic energies are down- or up-shifted.

length of the IR laser is adjusted such that one period of the electric field is a little longer than the ultrashort expected X-ray pulse. In the reported experiment, the IR wavelength was set to 10.6 μm, yielding a period of 35 fs. In this period, the electric field continuously rotates over a full cycle. Hence, information about the photoelectrons exact time of generation is not only imprinted in the kinetic energy but also in the ejecting angle. The photoelectrons are analyzed with a circular arrangement of 16 eTOF spectrometers, which can resolve the kinetic energy of the photoelectrons and their angular momentum. By using IR wavelengths with a period longer than the X-ray pulse, the duration and time structure of the X-ray pulse can be analyzed. The relative arrival-time of the X-ray pulses can be obtained by analyzing the observed streaking strength, given an initial characterization of the streaking pulse envelope [110].

4.3. X-ray / Optical Cross-Correlation

The most frequently used timing-tool schemes at hard X-ray XFEL sources are based on an X-ray / optical cross-correlation scheme. In this scheme, the X-ray pulses imprint its arrival-time by changing the optical properties (see Section 3.2.1) of a material. The change of these optical properties can be spectrally or spatially probed with a laser pulse. The first-ever measured X-ray pulse diagnostic using an X-ray / optical cross-correlation was measured in 2008 at FLASH and sampled the X-ray-induced reflectivity change of GaAs [112]. Since then, the X-ray / optical cross-correlation scheme evolved to be the

standard timing-tool scheme at most hard X-ray sources.

4.3.1. Spectral Encoding

In the spectral encoding scheme, the X-ray arrival-time is imprinted in the spectrum of a broadband laser pulse. The LCLS and SwissFEL standard timing-tool is based on the spectral encoding scheme, using a supercontinuum pulse which is chirped to a pulse duration in the order of 1 ps [113, 114, 115, 29]. Due to the chirp of the optical pulses, all spectral components are aligned in time, i.e., the wavelengths in the optical pulse are temporally ordered, usually higher wavelengths arrive early, and are followed by the lower wavelengths. The ps-long chirped optical pulse and the short X-ray pulse are temporally and spatially overlapped in an interaction sample (Fig. 4.2). Upon the arrival of the X-ray pulse, the optical properties of the sample are changed, altering the amplitude (absorption and reflection of the sample) of the transmitted chirped optical pulse. The X-ray arrival-time is imprinted in the optical spectrum as an amplitude modulation at the spectral region, which was currently transmitted through the sample as the X-ray pulse arrived. Depending on the arrival-time, the spectral position of this amplitude modulation changes. The time axis is calibrated by mapping the chirp of the optical pulse onto a wavelength-to-time calibration function. This can be achieved by splitting the laser beam into two branches before the supercontinuum generation and then perform an ordinary optical pump-probe experiment, where the delay between both pulses is systematically changed and the spectral position of the occurring coherent artifact is analyzed [116].
Various interaction samples for a spectral encoding timing-tool at LCLS were tested by Bionta et al. [115]. Si_3N_4 and Ce:YAG are the most suited materials and are nowadays the standard interaction samples [117], depending on the X-ray photon- and pulse-energy. Using these samples is a rather invasive method and large portions of the X-ray pulses are absorbed in the sample. Already in 2014, Bionta et al. commented that diamond would be the desired interaction sample, but the spectral encoding signal in diamond is often times to weak to be used as a reliable timing-tool sample. Only when the X-ray pulse energy is close to the damage threshold of diamond, a measurable timing signal can be observed [115].

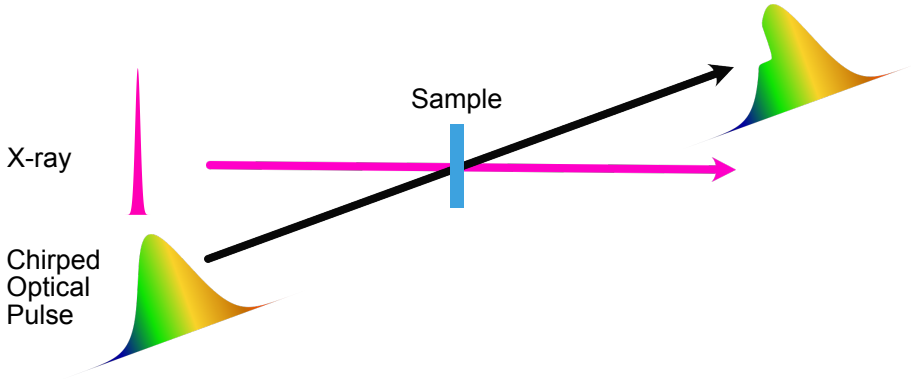

Figure 4.2.: Spectral encoding principle. A chirped, ps-long optical pulse is temporally overlapped with an ultrashort X-ray pulse in an interaction sample. By generating an electron cascade in the sample, the X-ray pulse imprints its arrival-time in the transmitted chirped optical pulse spectrum by modulating the spectral amplitude of the transmitted optical pulse. Depending on the relative arrival-time, the amplitude modulation occurs at different wavelength regions.

4.3.2. Spatial Encoding

The spatial encoding scheme is the main timing-tool at the SACLA XFEL [118]. The spatial encoding scheme uses a space-to-time mapping to measure the relative X-ray arrival-time (Fig. 4.3). The X-ray pulse impinge on the sample at normal incidence, while an ultrashort NIR optical pulse arrives at the sample with a defined angle. Hence, different spatial parts of the optical beam profile arrive at different times. The beam profile of the optical pulse is imaged with an appropriate camera. The X-ray pulse changes the refractive index of the sample. Therefore, the amplitude of the transmitted optical pulse is suddenly changing. Due to the incidence angle of the optical beam, spatial parts arriving before the X-ray pulse are transmitted without any X-ray-induced modulation, while later parts of the optical pulse are amplitude modulated by the striking X-ray pulse. The relative X-ray arrival-time is then encoded in the spatial intensity profile of the optical beam. The spatial-to-time axis calibration is purely geometrical and is defined by $\Delta t = \frac{D \cos \alpha}{c}$, with the footprint of the optical beam D on the sample and the incidence angle α. Typical samples for the spatial encoding scheme are Si_3N_4 or GaAs [119, 120, 121].
A disadvantage of the spectral encoding scheme is the requirement to have a larger X-ray beamsize on the sample to fully cover the optical footprint. In turn, this requires higher fluences compared to the spectral encoding scheme to achieve the same X-ray-induced refractive index change. To counter this, SACLA uses a single KB-mirror to create an extremely narrow $(3 \times 780)\,\mu m^2$ X-ray line focus to match the optical beam profile. With this, the required X-ray fluence is comparable to the spectral encoding scheme but at the cost of a more complicated setup.

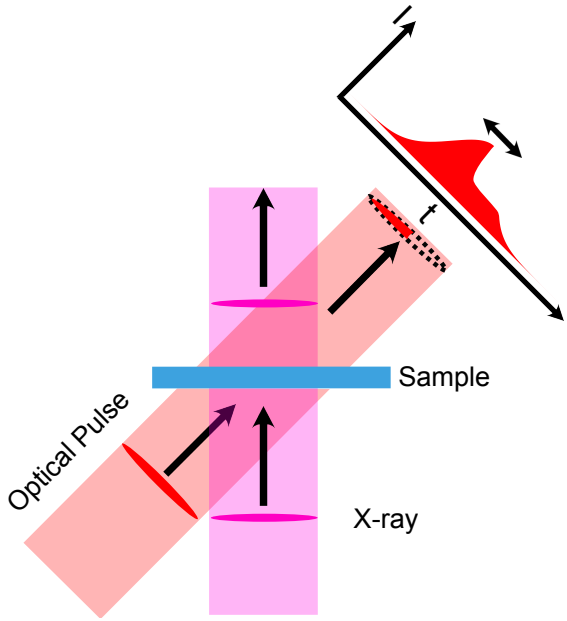

Figure 4.3.: Spatial encoding principle. The optical and X-ray pulse are overlapped in an interaction. The X-ray arrival-time is imprinted in the spatial beam profile of the transmitted optical beam.

4.4. Summary

Method	Complexity	Invasive	Soft X-ray	Hard X-ray	MHz compatible
EOS					
Streaking.					
Spec. En.					
Spat. En.					
This Thesis					

Table 4.1.: Overview of regular used timing-tool schemes and their discussed disadvantages. The self-referenced timing-tool scheme presented in this thesis is shown in the last row as comparison.

Although many advances in the temporal characterization of X-ray pulses were achieved in the last decade, yet a perfect approach to measure the relative arrival-time jitter needs to be found. Especially with regards to the new generation of XFEL facilities (EuXFEL and LCLS-II), covering a wide range of possible X-ray photon energies at high repetition rates, a robust, easy to operate, and widely applicable timing-tool scheme needs to be developed. An overview of all discussed timing-tool schemes and their particular drawbacks, preventing them from being an ideal solution, are collected in Tab. 4.1. Technically, EOS seems to be the perfect solution, but its downside is that it needs to be located in the electron tunnel, far away from all experimental stations. Either an entirely different laser than the experimental pump-probe laser is used, or the laser beam needs to be transported from the experimental stations over a very long beam path to the electron tunnel. In both scenarios, the measured electron bunch timing jitter might not be comparable to

the timing jitter at the experimental station.
The streaking approaches to measure the relative arrival-time suffer from complicated setups, requiring the generation and maintenance of THz pulses as well as a vacuum interaction chamber.
The working horses spectral and spatial encoding would be a good solution if they would reliably work with samples not absorbing a noticeable amount of the X-ray pulse. Although the timing-tools can be placed behind the actual experiment, making the attenuation of the X-ray pulse a non-problem, the sample itself would suffer from heat-pile-up at MHz facilities, ultimately destroying the sample (see Section 6.1).
The self-referenced timing-tool scheme presented in this thesis is almost fulfilling all requirements. It is easy to operate, works with soft and hard X-ray photon energies, and is compatible with diamond as interaction sample, making it robust enough to withstand high intensities at MHz repetition rates. The only minor downside is that a physical sample is placed in the X-ray beam, but with diamond we use a sample with a much smaller X-ray absorption cross-section than conventional spectral and spatial encoding samples, absorbing only a small percentage of the X-ray pulse energy at hard X-ray photon energies.
The results and experimental parameters of all cited publications in this Chapter are compiled in Tab. 4.2. This table will be used throughout this thesis to compare and benchmark our results against historical measurements.

[1] Amporphous Si_3N_4, Single Crystal Quartz, CVD Diamond, Crystalline Ce:YAG, Crystalline Gadolinum Gallium Garnet, Single Crystal Silicon

Year	Facility	Method	X-ray [keV]	X-ray Peak Fluence [mJ/cm^2]	Sample	Timing Jitter [fs FWHM]	Reference
2005	FLASH	EOS	N/A	N/A	ZnTe	456	[102]
2011	LCLS	Spec. En.	1	$\geq$215	Si_3N_4	N/A	[113]
2012	LCLS	Spat. En.	1.7	$\geq$220	Si_3N_4	N/A	[119]
2012	FLASH	THz Streak.	1.7	N/A	He gas	280	[105]
2013	LCLS	Spec. and Spat. En.	7.1	$\geq$155	Si_3N_4	423	[114]
2013	FLASH	Spat. En.	0.03 0.225	$\geq$620 $\geq$225	SiO_2	184 21	[122]
2014	LCLS	Spec. En.	1 - 10	N/A	Various[1]	N/A	[115]
2014	SACLA	THz Streak.	5 - 9	$\geq$465	Xe gas	213 - 404	[108]
2015	FLASH	THz Streak.	0.234	N/A	Xe gas	66 (OS)	[20]
2014	SACLA	Spat. En.	12	$\geq$440	GaAs	259	[120]
2016	SACLA	Spat. En.	10	$\geq$220	GaAs	603	[121]
2017	SACLA	THz Streak.	9	N/A	XE gas	580 - 700	[109]
2019	EuXFEL	Spec. En.	10.5	$\geq$1500	Gadolinum Iron Garnet	724 (RFS)	[123]
2020	EuXFEL	Spec. En.	N/A	N/A	Ce:YAG	655 (RFS) 56 (OS)	[124]

Table 4.2.: Diamond properties used to calculate the refractive index change with the Drude model.

5. Experimental Setup

The fundamental idea and principle of the self-referenced arrival-time measurement is described in this Chapter. The concept is based on an X-ray and optical cross-correlation scheme, using a frequency-to-time mapping to imprint the relative X-ray arrival-time in a chirped optical pulse (see Section 4.3.1). The intention of this setup is to increase the sensitivity for relative arrival-time measurements. This paves the way to use samples with smaller X-ray absorption cross-sections such as diamond. The setup is based on a CPI, which creates a self-referenced background-free measurement. An arrival-time signal will only be measured in the presence of an X-ray pulse, otherwise, a zero-signal will be obtained. This sets our approach apart from other X-ray / optical cross-correlation schemes. All other schemes deliver a minor signal on top of a large background signal, limiting their sensitivity, and oftentimes require a reference measurement. Our setup uses interferometric sensitivity to measure the phase- (and amplitude) change of a transmitted optical pulse after passing through a sample whose refractive index is changed by the partial absorption of a simultaneously transmitted X-ray pulse.
The complete setup to measure the self-referenced arrival-time signal is illustrated in Fig. 5.1. Slight variations of the setup during the measurement campaigns at EuXFEL (Section 7.1) and SACLA (Section 8.1) are described in the respective chapters. An ultrashort broadband laser pulse is used for the frequency-to-time mapping. This optical pulse enters the setup and is guided through a motorized optical delay line. The optical delay line is used to fine-tune the temporal delay between the optical and X-ray pulses. After the motorized delay line, a Schwer-Flint 11 (SF11) glass can be optionally inserted into the laser beam path (a). By adding the SF11, the chirp of the transmitted laser pulse can be increased, enhancing the laser pulse duration. As described in Section 3.4.1, chirping an optical pulse aligns all frequencies of a broadband pulse from low frequencies to higher frequencies in the temporal domain. The amount of glass needed to chirp the pulse depends on the desired pulse duration as well as the bandwidth and original pulse duration of the optical pulse. The temporal length of the chirped pulse defines the time window in which the relative arrival-time can be measured. The expected timing jitter for XFEL sources is $\leq$1 ps (Tab. 4.2), such that a measurement window of 1 - 1.5 ps is sufficient. Two polarizers and two a-cut Birefringent Crystals (BC) form the CPI (Fig. 5.1 b) and d). In all reported measurements we use a-cut αBeta Barium Borate (BBO) crystals as BC. The interaction sample, where the X-ray and optical pulse pairs are temporally and spatially overlapped, is placed in the center of the CPI (Fig. 5.1 c). The sample is used to imprint the X-ray arrival-time by an X-ray-induced refractive index change into the simultane-

ously transmitted optical pulse. Two lenses are used to focus the optical pulses onto the sample and to collimate the optical beam exiting the sample. The optical pulses are then guided to a spectrograph to evaluate the self-referenced X-ray arrival-time signal.
The self-referenced detection scheme relies on the inherent stability and robustness of the CPI. The first idea for a CPI occurred in 1958, described by L. Mertz [125] and was applied a few years later to realize a Polarization Fourier Spectrometer for astronomical measurements [126]. The advantage of this interferometer scheme is that the generated pulse-replica co-propagate through the setup, making it extremely robust against environmental disturbances, e.g., mechanical vibrations or air flows. Any occurring disturbance affects both pulse-replica equally, thus, effectively cancelling out the distortion. The CPI setup build for this thesis is illustrated in Fig. 5.2. The main parts are two identical a-cut BC between two crossed polarizers. The transmission axis of the first polarizer is set to 45°, such that behind the first polarizer, the transmitted light has a defined linear polarization of 45° (1). The 45° polarized pulse can equally be described as a linear combination of a horizontal (blue) and vertical (orange) Polarization Component (PC) with equal amplitudes.
The first BC is aligned with its optical axis to be horizontal. Due to the birefringence of the crystal, the horizontal PC experiences the extraordinary refractive index n_e, while the vertical PC experiences the ordinary refractive index n_o of the crystal. After passing the first BC, the two PCs are then time-delayed with respect to each other (2). The time-delay between the two PCs is defined by the ordinary n_o and extraordinary n_e refractive index and the crystal thickness.
The two PCs are transmitted through the (ideally) optically isotropic sample, experiencing the same refractive index. The second BC is rotated orthogonally to the first BC, such that the optical axis is now perpendicular to the optical axis of the first BC. The PC arriving first, which experienced the extraordinary refractive index n_e in the first BC, now experiences the ordinary refractive index n_o of the second BC, and vice versa for the trailing PC. In such a configuration, both PCs are temporally overlapped again behind the second BC. After the second BC, both PCs are effectively propagated through two crystals with identical thickness and accumulated the same amount of additional phase by experiencing the same effective refractive index. Both PCs are then exactly synchronized. The linear combination of both PCs recreates the original 45° linear polarized optical pulse (3).
This 45° polarized optical pulse is then completely blocked by the second polarizer, whose transmission axis is oriented to be -45°. In this case, no optical pulse can be observed behind the second polarizer (4).
If the amplitude or phase of one of the two PCs is artificially manipulated, the situation is different: In this case, the ultrashort X-ray pulse is temporally overlapped with the trailing vertical PC as shown in (5). The entire leading horizontal PC and half of the trailing vertical PC is transmitted through the sample before the X-ray pulse (green) strikes

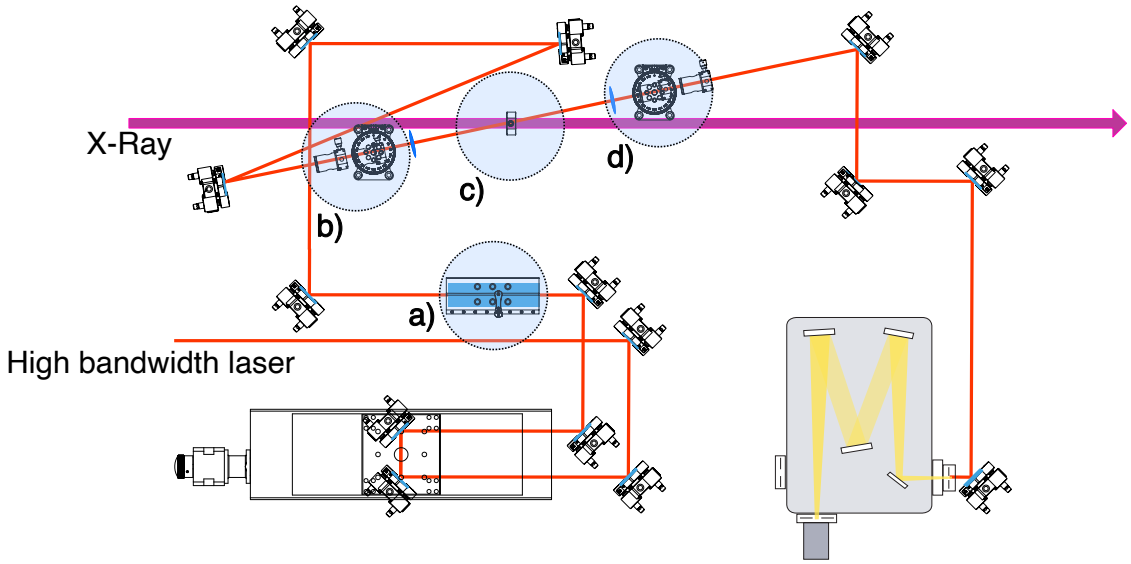

Figure 5.1.: Principal setup of a self-referenced arrival-time setup. The high bandwidth laser is guided over a mechanical delay stage and is transmitted through a dispersive SF11 block to additionally chirp the optical pulse. The first part of the CPI (b) creates two time-delayed PCs and is in front of the sample (c). Behind the sample, the second part of the CPI synchronizes the time-delayed PCs and creates the self-referenced arrival-time signal in the presence of an X-ray pulse. This signal is guided to a spectrograph where it is spectrally evaluated.

the sample. The X-ray pulse generates an electron cascade, ultimately modulating the electronic band structure, ultimately changing the complex refractive index of the sample (Section 3.2.1). Due to this change of the refractive index, parts of the vertical PC, arriving after the X-ray pulse, pass through the sample with the modulated refractive index. These are then subject to a phase-shift of the transmitted optical light (6). Due to the chirp of the original optical pulse, the temporally leading red spectral parts of both PCs transmit through the sample without the modulated refractive index. The trailing spectrally blue parts of the leading horizontal PC also pass through the sample with the static refractive index, while the identical blue parts of the trailing horizontal PC passes through the sample with its X-ray-modulated refractive index and are phase-shifted by a small amount.

Behind the second BC, the original 45° polarization can not be recreated by a linear combination of both synchronized PCs. The red parts of both PCs are still perfectly synchronized and recreate the original 45° polarization, and are therefore blocked by the second polarizer. However, the blue parts are not synchronized anymore due to the X-ray-induced phase-shift of the blue spectral parts in the trailing horizontal PC. The linear combination of these parts now yield an elliptically polarized optical pulse (7).

The perfectly synchronized first part of the optical pulse with the red spectral parts is still 45° polarized and still not transmitted by the second polarizer. The elliptical polarized

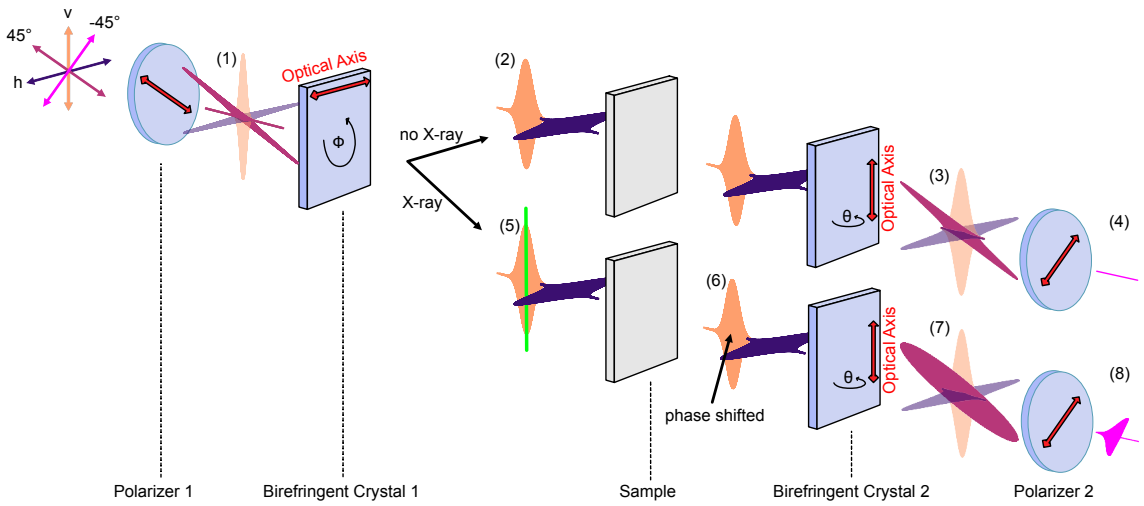

Figure 5.2.: Common-Path-Interferometer principle. The first polarizer defines the polarization of the optical to 45°, which can equally be described by two perpendicular polarized PCs (1). In the absence of any X-ray pulse no phase-shift is introduced and the two time-delayed PCs (2), generated by the first BC, can be perfectly synchronized behind the second BC. This recreates an 45° polarized pulse (3), blocked by the second polarizer (4). In the presence of an X-ray pulse (5), a phase-shift is introduced (6), preventing the perfect synchronization of both PCs behind the second BC and creating an elliptical polarized pulse (7), partly transmitted through the second polarizer (8).

blue parts of the optical pulse can partially transmit through the second polarizer, generating a self-referenced background-free arrival-time signal (8). Depending on the relative arrival-time between the optical and X-ray pulse, the spectral position of the transition between the original 45° polarization to the elliptical polarization changes for each optical / X-ray pulse pair and is clearly visible in the observed spectrum by a cut-off edge.
Due to the sensitivity of the setup to the two PCs, it needs to be placed between the last mirror in front of the sample and the first mirror after the sample, thus, all elements of the CPI need to be ordered on a straight beam path without any mirrors. This is important because the vertical and horizontal PCs (and therefore all linear combinations of both) are not reflected on mirrors with identical efficiencies when both are not under normal incidence [91]. Changing the amplitudes of each of the two PCs would negatively affect the possibility to recreate the original 45° polarized optical pulse behind the second BC.
The alignment of the setup starts by inserting the two crossed polarizers, where the transmission axis of the first polarizer is set to 45° and the transmission axis of the second polarizer is set to -45°. In this crossed-polarizer configuration, no light is detected behind the second polarizer. Between the two crossed polarizers, the two BCs are placed. As described before, the optical axis of the first BC has to be horizontal, while the optical axis of the second BC is set to be vertical. Each crystal is mounted on two rotational stages to have fine control of both rotational planes, granting a precise alignment of the optical axis (Φ angle in Fig. 5.2) and the rotation in the horizontal plane (θ angle) of both crys-

tals. The relative amplitudes of the two PCs can be adjusted with the Φ rotation angle. By rotating the angle θ around the optical axis, the effective thickness of the second BC can be changed, thus, fine-tuning the temporal synchronization of both PCs with extreme temporal accuracy of tens of as [127]. This fine adjustment is necessary since both BCs never have an identical thickness due to minor production errors.

6. Numerical Results

Numerical simulations are a crucial component to fully understand the fundamental mechanisms in newly developed experimental setups. Numerical simulations are used in this Chapter to demonstrate various aspects of the operation of the new self-referenced timing-tool.
An estimate of the X-ray-induced heat load for different timing-tool interaction samples is calculated in the first subsection. The calculations suggest that diamond is an exceptionally well-suited candidate to be used as a timing-tool sample, emphasizing the need for a timing-tool detection scheme, which can be used with diamond as interaction sample. This alone already justifies the development effort of the self-referenced detection scheme, enabling diamond as interaction sample.
The second subsection describes theoretical aspects of this new timing-tool detection scheme. The electron density in the conduction band after the partial absorption of an X-ray pulse is calculated. With this electron density, two different models are used to calculate the X-ray-induced refractive index change in diamond. The simple Drude theory and the more advanced Maxwell Garnett model for dielectric mixtures are used for these calculations. Finally, a full description of the emergence of the timing-tool signal is given by propagating an initial laser pulse through the optical setup and treating all dispersive effects up to the third order. A heavy focus is put on the X-ray-induced phase-shift of the transmitted optical pulse behind the diamond interaction sample as well as on the sensitive alignment of the two BC of the CPI.

6.1. X-ray-Induced Heat Evolution

A major concern at MHz repetition rate XFEL facilities is the X-ray-induced heat load on all components interacting with X-ray pulses. This includes optical elements used for beam transportation but also all non-invasive X-ray diagnostic elements. All reflective (e.g. X-ray mirrors, monochromators) or transmissive (e.g. vacuum windows, Compound Refractive Lenses (CRLs)) optics are susceptible to X-ray-induced damage. The permanent radiant exposure of materials with X-ray photons causes irreversible damage through radiolysis [68].
At XFEL facilities with intense X-ray pulses, even a single fs-short pulse can cause damage to materials [68, 128, 129]. Recent experiments at FLASH with bulk materials have shown that the X-ray-induced single-pulse damage threshold is closely related to the melting point of the irradiated material [130, 131]. The partial absorption of the intense

X-ray pulses deposits a substantial amount of energy in the material, ultimately leading to a temperature increase. At MHz repetition rates, the temperature increase by subsequent X-ray pulses can pile-up to temperatures far above the melting point of many materials and destroy them in the long run.
A simulation of the heat evolution after an EuXFEL pulse train is presented in this chapter. The common materials for X-ray beam diagnostics used for the simulation studies are Silicon (Si), Si_3N_4 and YAG. These materials are compared to a diamond interaction sample. In the following, diamond always refers to an artificial CVD diamond.
To calculate the thermal evolution of each material after the exposure to an X-ray pulse and the subsequent heating of its material volume, the so-called heat equation needs to be solved. One first calculates the instantaneous X-ray-induced heat in the given sample. Under the simple assumption that the energy of all absorbed X-ray photons is turned into heat, the temperature increase ΔT of the material volume can be calculated with [132]

$$\Delta T = \frac{\Delta Q}{m c_p}\,, \tag{6.1}$$

where ΔQ is the absorbed X-ray energy, m the mass of the absorbing material volume and c_p the material-specific heat capacity. The absorbed X-ray energy is calculated with the Beer-Lambert law (Eq. 3.3 in Section 3.2). The material-specific X-ray absorption lengths are shown in Fig. 6.1 a) with the photon energy-dependent absorption lengths taken from the X-ray database [133]. The attenuation of an X-ray pulse through a 50 µm thick sample of each material is shown in Fig. 6.1 b). The gray areas in both subfigures indicate the covered energy range at the EuXFEL hard X-ray beamlines.
The usual X-ray pulse duration of well below 100 fs is short compared to phonon lifetimes, such that in a first approximation, no thermal conduction takes place during the pulse duration [134]. Thus, the X-ray pulse absorption can be treated as an instantaneous event. Photoelectrons generated by the X-ray pulse are rather localized, and their mobility is negligible compared to the X-ray beam diameter of tens of µm.
After the initial increase in temperature, the locally heated volume will start to cool down via thermal conduction of the heat to regions outside the illuminated volume. In this approach, the heated sample volume is assumed to be much smaller than the total sample volume, such that a nearly infinite thermal bath around the heated volume can be assumed. The heat diffusion in solids is described by the heat equation [135, 136]

$$\frac{\partial T(\mathbf{x},t)}{\partial t} = D\nabla^2 T(\mathbf{x},t)\,, \tag{6.2}$$

with the temperature distribution $T(\mathbf{x},t)$ and the thermal diffusivity D. The thermal diffusivity is defined as $D = \frac{k}{\rho c_p}$ with the thermal conductivity k, heat capacity c_p and density $\rho = m/V$. Usually, the samples used for X-ray diagnostics are thin (up to 100 µm), thus, a reasonable assumption is that the X-ray-induced heat distribution along the X-ray path through the sample is uniform. Under these conditions, the heat equation can be

reduced to two dimensions. The X-ray-induced heat flows radially away from the hot center of the material. In two dimensions, the heat equation from Eq. 6.2 is reduced to:

$$\frac{\partial T(\mathbf{x},t)}{\partial t} = D\left(\frac{\partial^2 T(\mathbf{x},t)}{\partial x^2} + \frac{\partial^2 T(\mathbf{x},t)}{\partial y^2}\right) \tag{6.3}$$

To numerically solve the two-dimensional heat equation, a Finite Difference Element (FDE) approach is applied, using the so-called Forward Euler Method [137]. In this approach, the spatial domain is discretized into a mesh with discrete mesh elements of length Δx and width Δy. Each mesh element is set to a temperature, according to the temperature distribution in the material after the partial X-ray absorption. The heat equation is then solved for each pair of neighboring mesh elements. To numerically solve the heat equation, it needs to be discretized to an algebraic function where the derivatives are replaced by FDE approximations. The discretized algebraic function can be written as [138]

$$T_{ij}^{(n+1)} = T_{ij}^{(n)} + D\Delta t\left[\frac{T_{i+1j}^{(n)} - 2T_{ij}^{(n)} + T_{i-1j}^{(n)}}{\Delta x^2} + \frac{T_{i+1j}^{(n)} - 2T_{ij}^{(n)} + T_{i-1j}^{(n)}}{\Delta y^2}\right], \tag{6.4}$$

where i, j are the indices of the mesh element position [i,j] in the two-dimensional mesh. To ensure the convergence of the algorithm, it must be ensured that

$$\frac{D\Delta t}{\Delta x^2 + \Delta y^2} \leq \frac{1}{2} \tag{6.5}$$

holds true.

Several material-specific properties need to be known for the calculation: the density ρ, which also depends on the thermal expansion coefficient α, the isobaric heat capacity c_p, and the thermal conductivity k. The density of a material is given by its mass/volume ratio m/V. The temperature-dependent change of a material volume is described by $\Delta V = 3\alpha_l \Delta T V$, where α_l is the linear thermal expansion coefficient and V the material volume before the temperature change. Experimental data for the temperature-dependent linear thermal expansion coefficients used in the simulation for diamond [139], Si [140], Si_3N_4 [141] and YAG [142] are shown in Fig. 6.2 a). The experimentally obtained data of the temperature-dependent specific heat capacities of diamond [143], Si [144], Si_3N_4 [144] and YAG [145] are indicated by the dots in Fig. 6.2 b) and are fitted with the fitting functions from the respective publications, using either the Shomate equation [146] or the Debye model [147, 143]. The experimental thermal conductivities are shown in Fig. 6.2 c) for diamond [148], Si [149], Si_3N_4 [150] and YAG [151] as dots, while a function $\propto 1/T$ is used to fit the experimental data.

When comparing the data in Fig. 6.1 and 6.2, it is obvious that diamond is the best material to withstand an intense X-ray pulse. The main reason is its longer X-ray absorption length for all X-ray energies above 3 keV. In combination with its high isobaric heat capac-

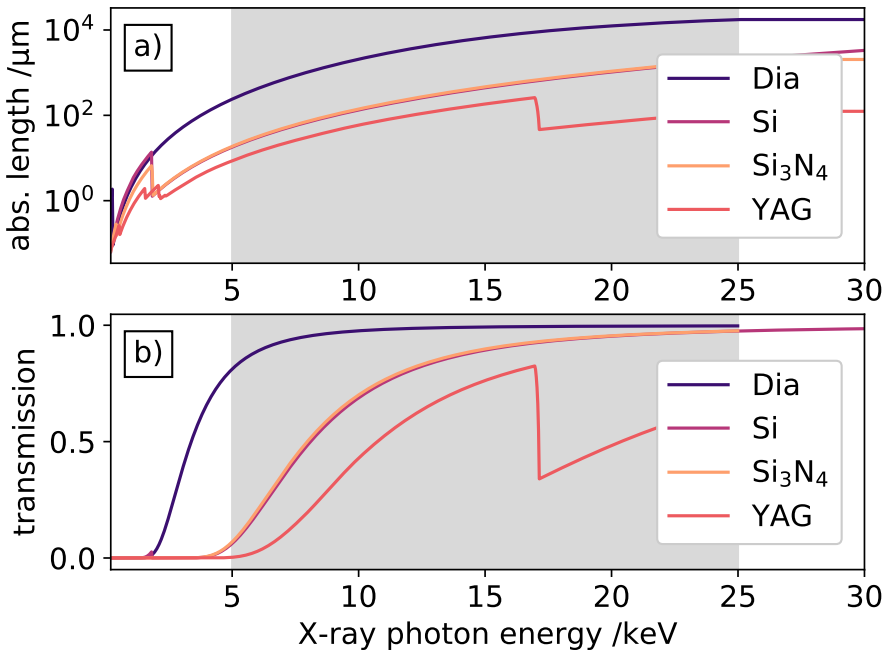

Figure 6.1.: X-ray attenuation lengths (absorption) for diamond (blue), Si (purple), Si_3N_4 (orange) and YAG (red) for X-ray photon energies from 200 to 25000 eV are shown in a). The transmission through 50 µm thick samples of diamond, Si, Si_3N_4 and YAG is shown in b). The gray shaded area indicates the X-ray photon energy covered by the EuXFEL hard X-ray beamlines. Data taken from [133].

ity and by far the largest thermal conductivity, diamond is expected to have the lowest X-ray-induced temperature increase and the fastest cooling after absorption of an intense X-ray pulse.

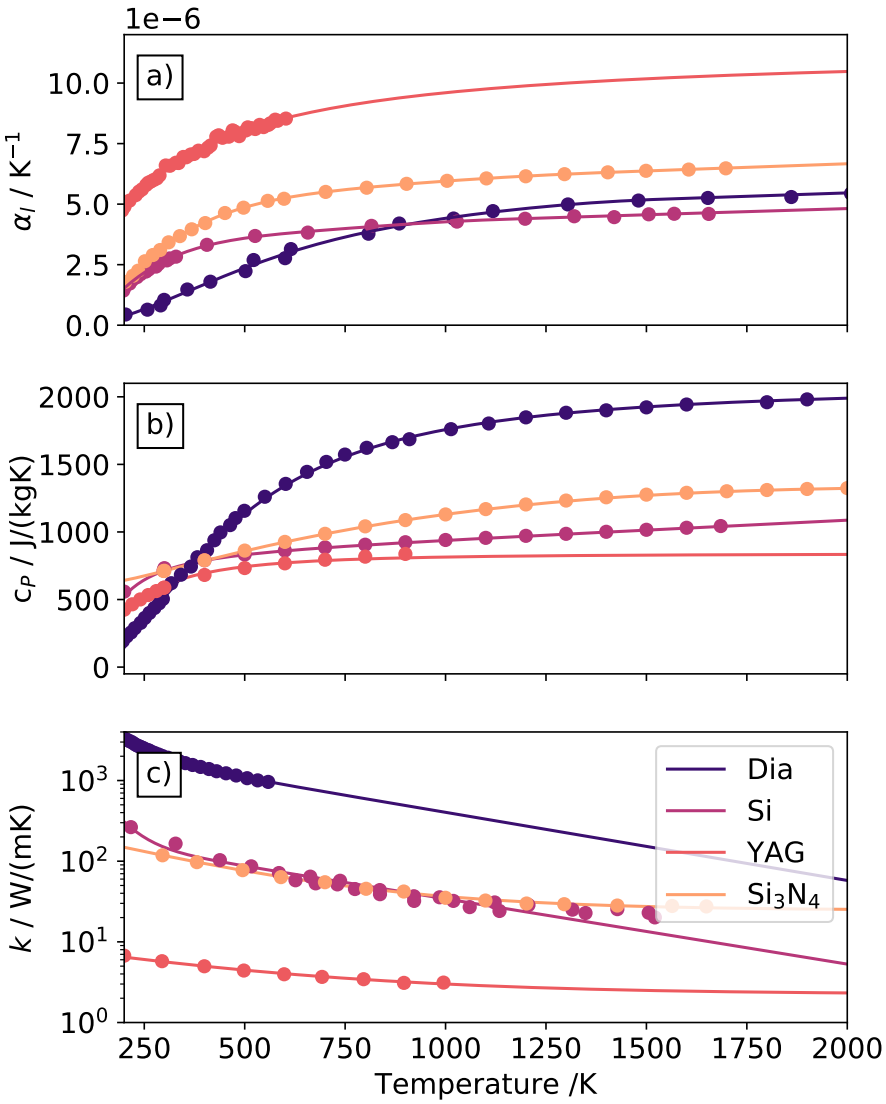

Figure 6.2.: Thermal properties of the four different materials used in the X-ray-induced heat simulation. Dots are experimental data points, while the solid lines are fitting functions with values from the respective sources. The temperature-dependent linear expansion coefficients of the materials are shown in a) for diamond [139], Si [140], Si_3N_4 [141] and YAG [142]. Experimental digitized isobaric heat capacities of the materials are shown in b) for diamond [143], Si [144], Si_3N_4 [144] and YAG [145]. The thermal conductivities of the materials are shown in c), again for diamond [148], Si [149], Si_3N_4 [150] and YAG [151].

A two-dimensional FDE simulation is shown in Fig. 6.3. The temporal evolution of the initial heat distribution in a 50 µm thick Si crystal after the partial absorption of an X-ray pulse (E_{ph} = 10 keV, E_{pulse} = 20 µJ) with Gaussian beam profile (⌀ = 100 µm FWHM) is shown in a) after 0 µs, b) after 5 µs, c) after 10 µs and d) after 60 µs. The initial temperature of the Si crystal is set to room temperature at 293 K. According to Eq. 3.3, 31% of the incident X-ray pulse energy is absorbed in the material and directly converted to heat (Eq. 6.1).
Due to the Gaussian beam profile, the temperature increase in the center of the beam is highest with $\Delta T \approx 6$ K. The initial sharp heat distribution at t=0 µs diffuses radially outwards, yielding a decrease of the maximum temperature in the center.
In the following, only the temporal evolution in the center of the Gaussian beam is shown since potential thermal-induced damage will occur there first. A comparison of the four different materials, each with a thickness of 50 µm and irradiated with the identical X-ray pulse (10 keV, 20 µJ, 100 µm FWHM) as before, is shown in Fig. 6.4. Due to its larger X-ray absorption cross-section, YAG absorbs with 51% the highest fraction of energy, followed by Si and Si_3N_4 with 31% and 30%, respectively. Because of the high X-ray absorption cross-section and the lowest heat capacity, YAG heats up most by a single X-ray pulse with a temperature increase of ΔT=7.3 K. Despite nearly identical X-ray absorption cross-sections and heat capacities at room temperature, the X-ray-induced temperature in Si and Si_3N_4 is different due to the different densities of the materials. Their increase in temperature is ΔT=6.6 K for Si and ΔT=4.7 K for Si_3N_4. The temperature increase in diamond is, according to this simulation, only ΔT=0.6 K, due to the much smaller X-ray absorption cross-section, absorbing only ~2% of the X-ray pulse energy.
After the initial temperature increase, the heat diffusion and subsequent cooling process of the material is simulated over a time span of 100 µs. The time steps of the simulation are set to Δt=10 ns, while the grid granularity is set to $dx = dy$ = 10 µm to meet the stability requirement of Eq. 6.5. Being a ceramic compound, the thermal conductivity of YAG is by far the lowest of the four materials, and nearly no thermal diffusion is observed. Like the specific heat at room temperature, the thermal conductivity of Si and Si_3N_4 at room temperature is almost identical, thus, the cooling of both materials occurs on similar timescales and are very slow compared to diamond. The thermal conductivity of diamond is nearly 10 times larger than those for Si and Si_3N_4 and values above 2100 W/mK in a very pure CVD diamond are reported [148, 152], thus, a huge difference in thermal diffusion rates is expected.

At the highest possible repetition rate at EuXFEL, a pulse train can be filled with up to 2700 X-ray pulses at an intra-train repetition rate of 4.5 MHz, corresponding to a time separation of 222 ns between subsequent X-ray pulses [23]. None of the materials can fully cool down to room temperature by thermal diffusion after 222 ns, even under the very moderate X-ray beam conditions with a rather large beam diameter of 100 µm FWHM and pulse energy of only 20 µJ (Fig. 6.4). Diamond is very close to a full recovery af-

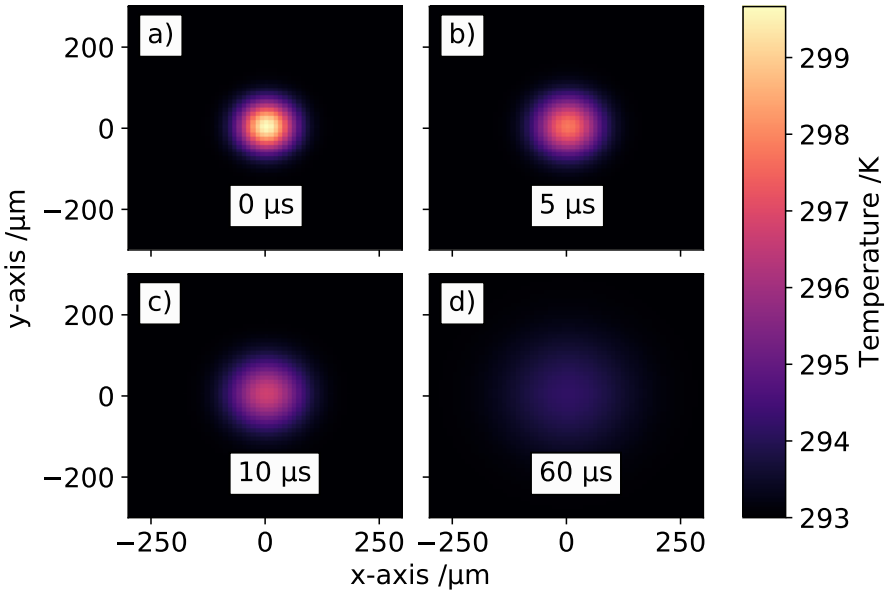

Figure 6.3.: Heat diffusion in 50µm thick Si after the initial absorption of an X-ray pulse at 10 keV with a pulse energy of 20 µJ at different times of t=0 µs in a), t=5 µs in b), t=10 µs in c) and t=60 µs in d).

ter 222 ns, while YAG, Si and Si_3N_4 do not equilibrate back to room temperature after 100 µs. Next, the heat pile-up within a pulse train in these materials is investigated with a simulation of 20 X-ray pulses with a temporal separation of 222 ns (Fig. 6.5). The X-ray pulse energy and beamsize remains unchanged. The materials diffuses heat away from the center region between the X-ray pulses. Diamond decreases its temperature from initially 293.6 K to 293.3 K in these first 222 ns. Si heats up to 299.6 K and decreases its temperature to 299.4 K. Si_3N_4 is heated up to 297.7 K and cools down to 297.6 K. During the 222 ns time window, almost no cooling can be observed in YAG.

After 20 X-ray pulses, YAG, Si and Si_3N_4 are already heated up to 467 K, 409 K and 378 K, while diamond remains at 296 K. Due to their temperature dependence, the thermal properties c_p, k and α are severely affected. Since the heat capacity c_p increases with temperature, more absorbed energy is needed to heat up the same material volume. On the other hand, the thermal conductivity k decreases with temperature, slowing down the thermal diffusion process. Since only the volume, radiated by the central parts of the Gaussian X-ray pulses, are noticeably heated up, the temperature difference between this hot volume and the cooler surrounding volume increases, leading to a steeper temperature gradient, and thus a more efficient heat diffusion to the surrounding material. This effect can be seen on Si and Si_3N_4 when comparing the slope of the cooling process between the first two X-ray pulses and the 19th and 20th X-ray pulse. Between the later pulse pair, the temperature decrease between the two X-ray pulses is increased compared to the first pulse pair.

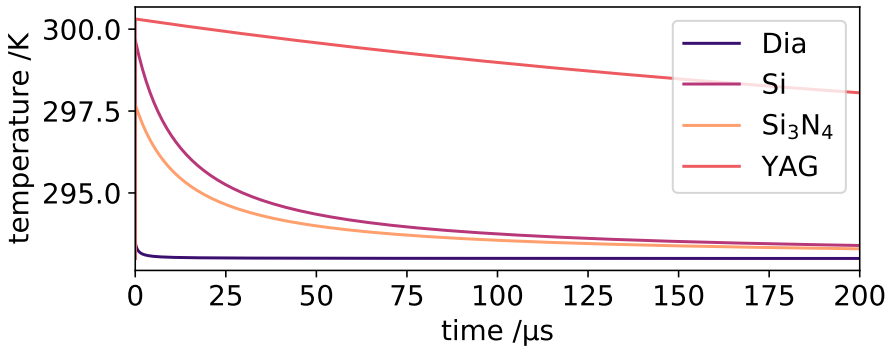

Figure 6.4.: X-ray-induced temperature increase and subsequent diffusion, after the partial absorption of a 20 µJ, 100 µm FWHM X-ray pulse with photon energies of 10 keV in diamond, Si, Si_3N_4 and YAG in the center of the Gaussian beam profile.

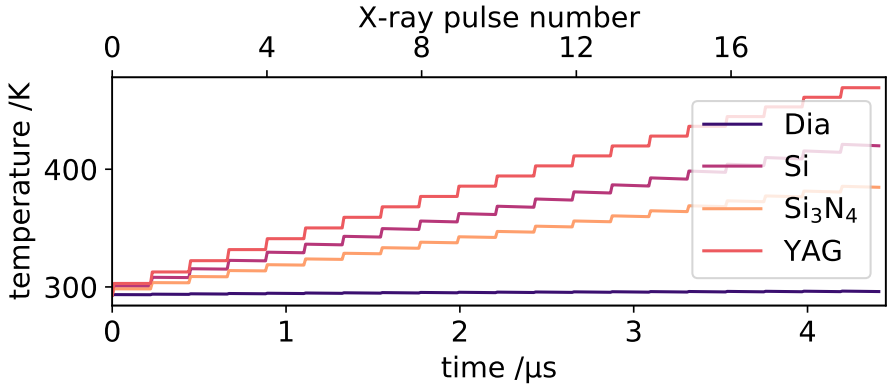

Figure 6.5.: X-ray-induced heat after 20 X-ray pulses with a repetition rate of 4.5 MHz. The X-ray pulse properties are set to 10 keV photon energy, 20 µJ pulse energy with a 100 µm FWHM Gaussian beam profile.

To explore the accumulation of heat over a longer time, a pulse train with 1000 X-ray pulses is simulated, using the same X-ray conditions as above (Fig. 6.6). In a), the simulated temperature-increase for diamond, YAG, Si and Si_3N_4 after 1000 X-ray pulses is shown. The FDE simulation stops when the melting point of a material is reached. Si reaches its melting point of 1687 K after 464 X-ray pulses, Si_3N_4 reaches its melting point of 2173 K after 985 X-ray pulses and YAG reaches its melting point of 2213 K already after 372 pulses. Analyzing the entire dynamic of repeated X-ray-induced heating and subsequent cooling processes, two dominating trends can be identified. At first, the surrounding material is close to room temperature, while the material volume irradiated by the central parts of the Gaussian X-ray pulses can already experience a temperature increase of several hundreds K. This leads to a large temperature gradient between the hot X-ray heated material and its surrounding material, which enhances the heat diffusion process to be very efficient and slowing down the overall build-up of heat in the material. This is the case for Si between the first and roughly 200th X-ray pulse and for

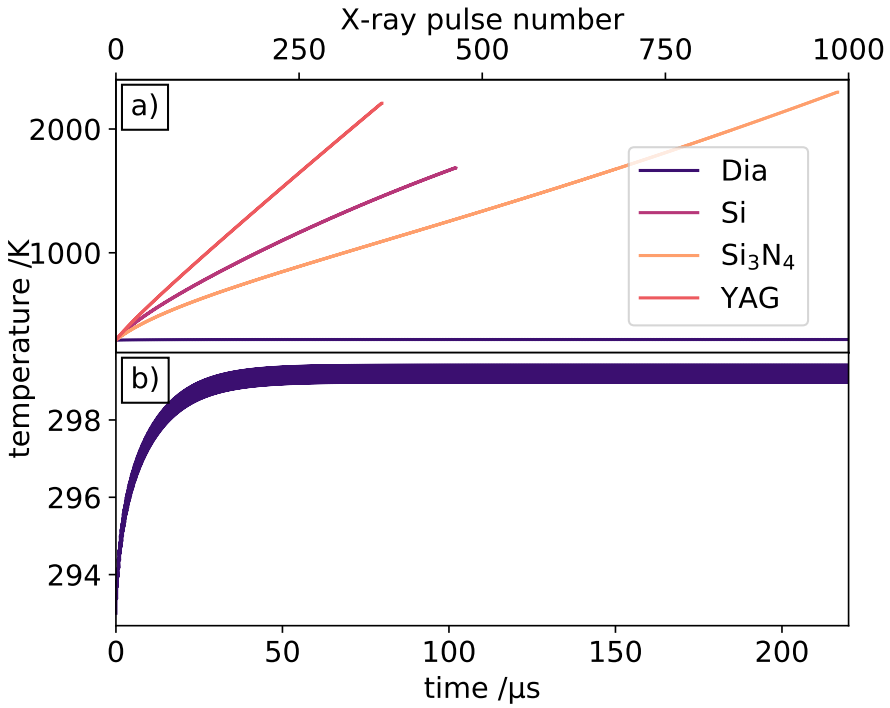

Figure 6.6.: Temperature increase in diamond, Si and Si_3N_4 with a thickness of 50 µm after partial absorption of 1000 X-ray pulses at 10 keV with pulse energies of 20 µJ and a Gaussian beam profile with a diameter of 100 µm FWHM. The FDE calculation in a) are stopped at the melting points of the different materials. The temperature evolution of diamond is enlarged in b) to make the temperature increase visible.

Si_3N_4 between the first and roughly the 150th X-ray pulse. In this region, the overall heat pile-up is slowing down. When the surrounding material is also heated up to a certain temperature, an equilibrium between the X-ray-induced temperature and the temperature diffusion rate to the surrounding material is reached. From this point on, the overall heat in the material increases linear. This can be observed in Si and Si_3N_4 until they reach their melting point and for YAG already after the first few X-ray pulses.
Diamond behaves special in this simulation as shown in Fig. 6.6 b) with a largely reduced temperature range on the y-axis. Unlike Si or Si_3N_4, diamond does not reach a diffusion equilibrium where a linear increase in temperature over many X-ray pulses can be observed. Due to its very high thermal conductivity, diamond reaches a temperature equilibrium after 280 X-ray pulses at a temperature of only 299 K. Under these X-ray conditions, an equilibrium is reached between the temperature gradient of the hottest central material volume and the surrounding cooler material, ending in a stable cycle between X-ray-induced heating and subsequent cooling by heat diffusion. These results of the FDE simulation suggest that neither YAG, Si nor Si_3N_4 are suitable to be used as a timing-tool sample for a MHz repetition rate XFEL. Further simulations are shown in Fig. 6.7 to investigate different X-ray energies impinging on a 50 µm thick diamond. Fig. 6.7 a) shows the temperature increase for 2700 X-ray pulses with an unattenuated and unfocussed X-

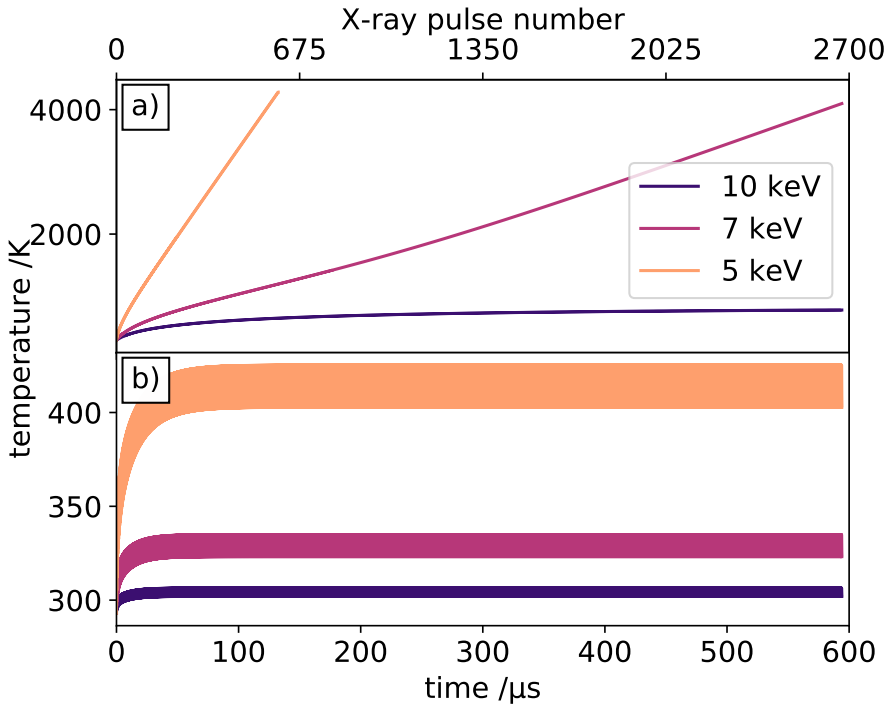

Figure 6.7.: X-ray-induced heat in 50 µm thick diamond after a full 2700 X-ray pulse train with internal 4.5 MHz repetition rate. In a) the heat pile-up for an unattenuated X-ray beam with 2 mJ pulse energy and a beam profile diameter of 500 µm FWHM for different X-ray photon pulse energies is shown. In subfigure b) the heat build up is simulated for attenuated beams after the monochromator with 20 µJ pulse energy and an FWHM beam diameter of 30 µm.

ray beam (E_{pulse} = 2 mJ, ⌀ = 500 µm FWHM). With these conditions, diamond can be used for X-ray photon energies above 7 keV. With the fully filled pulse train at 7 keV, diamond nearly reaches its melting point of ~4400 K [152]. Unfocussed pulse trains above 7 keV can therefore be viably used in diamond. The number of admissible X-ray pulses in a pulse train decreases below 7 keV, e.g., at 5 keV, the melting point of diamond is already reached after around 650 pulses.

Fortunately, less fluence is needed to operate a diamond-based self-referenced timing-tool, as will be shown later in Chapter 7 and 8. In Fig. 6.7 b), a full X-ray pulse train (2700 pulses) with X-ray pulse energies of 20 µJ and an FWHM beam diameter of 30 µm are assumed as reasonable conditions in which this diamond-based timing-tool can be operated. These conditions yield a peak intensity of ~2 J/cm^2, which is a reasonable upper boundary of the required X-ray pulse energy to operate a timing-tool (Tab. 4.2). Under these conditions, even at X-ray photon energies of 5 keV, diamond only heats up to 424 K over a full pulse train. Using higher X-ray photon energies is unproblematic, as shown in Fig. 6.7 b). Noticeably, even at 5 keV a stable equilibrium temperature is reached, which means that diamond could also be used as a diagnostics sample in a continuous wave mode XFEL facility at 4.5 MHz at these low fluences.

The results of the FDE simulations presented in this chapter are in agreement to similar three-dimensional simulations done by H. Sinn [25].

6.2. Self-referenced Timing-Tool Signal Simulation

To understand the emergence and fundamental concepts of the new self-referenced timing-tool scheme, a simulation is outlined in this Section, which aims to simulate the full performance and expected signal of the new approach. A transform-limited Gaussian laser pulse is propagated through the optical elements of the setup, interacting with the sample whose refractive index is altered by the X-ray pulse. The Section is divided into two subsections. In the first subsection, the expected X-ray-induced transient refractive index change in diamond is calculated by using two different methods. In the second subsection, the calculated transient refractive index change is used to simulate the shape and signal strength of the expected self-referenced timing-tool signal in diamond.

6.2.1. X-Ray-Induced Electron Density in Diamond

The absorption of X-ray photons in diamond depends on the thickness of the sample and the X-ray photon energy. As described in Section 3.2, the absorption probability is calculated with the Lambert-Beer-Law (Eq. 3.3), using the X-ray photon energy-dependent absorption length (Fig. 6.1). The number of actually absorbed X-ray photons is calculated from the total X-ray photon number in an X-ray pulse and the X-ray absorption cross-section of the absorbing sample. The photon number in an X-ray pulse is calculated with $E_{\text{pulse}}/E_{\text{photon}}$. For example, an X-ray pulse with 2 mJ and X-ray photon energies of 10 keV consists of 1.25×10^{12} photons.
Depending on the photon energy of the absorbed X-ray photon, a specific amount of secondary electrons are generated (Chapter 3.2.1). The pair creation energy in diamond is estimated to $E_{\text{pair}} = 12.2$ eV [74, 153] and can be used to calculate the number of secondary electrons by dividing the X-ray photon energy by this pair creation energy. The created electron density is calculated using the following set of equations [114]:

$$E_{\text{abs}} = E_0 \left[1 - \exp\left(\frac{d}{\cos(\theta_{\text{X-ray}})}\frac{1}{\mu_{\text{abs}}}\right)\right] \tag{6.6}$$

$$E_{\text{dens}} = \frac{E_{\text{abs}}}{A[d/\cos(\theta_{\text{X-ray}})]} \tag{6.7}$$

$$n_{\text{e}} = E_{\text{dens}}\frac{1}{E_{\text{pair}}} \tag{6.8}$$

The absorbed X-ray pulse energy E_{abs} is calculated with the X-ray pulse energy E_0, the sample thickness d, the incidence angle of the X-ray beam $\theta_{\text{X-ray}}$ (here normal incidence) and the material-specific X-ray absorption length μ_{abs}. The energy density E_{dens} in the absorbing material volume is then calculated by dividing the total absorbed energy E_{abs}

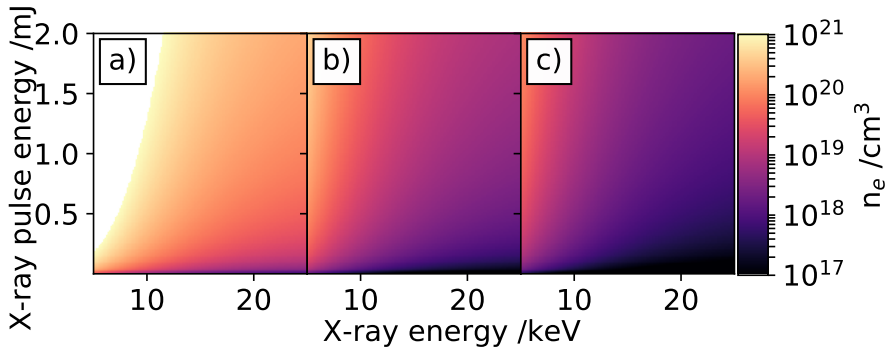

Figure 6.8.: X-ray-induced electron density after partial X-ray pulse absorption in a 50 µm thick diamond. Electron densities for three different X-ray beam diameters of a) 20 µm, b) 100 µm and c) 200 µm. are shown. The white area in a) indicate electron densities above 10^{21} electrons/cm^3.

by the irradiated volume. A flat-top X-ray beam profile, yielding a uniform electron density over the X-ray beam profile footprint, is assumed for these calculations. The electron density n_e following the electron cascade is calculated by dividing the energy density E_{dens} by the pair creation energy E_{pair}.
Calculations of X-ray-induced electron densities for X-ray pulse energies from 2 µJ to 2 mJ and for X-ray photon energies from 5 keV to 25 keV are shown in Fig. 6.8, using a flat-top beam shape with beam diameters of 20 µm a), 100 µm b) and 200 µm in c). Under these conditions, electron densities range from 10^{17} to potentially 10^{22} electrons/cm^3. From the discussion in Section 6.1, it can be deduced that the absorption of X-ray photons leading to an electron density above 1×10^{21}, would drive the diamond temperature close to its melting point. Thus, the region above this electron density is excluded (white area) in Fig. 6.8 a). The calculated electron densities will serve as a basis to calculate the X-ray-induced refractive index change in the next Section.

6.2.2. X-Ray-Induced Transient Refractive Index in Diamond

The refractive index of a material is directly related to its dielectric function via the relation $n^2 = \epsilon_r$. One of the most commonly used theoretical models in solid-state physics to describe the electro-optical properties is the Drude model (Section 3.3.2). Despite being formulated for metals, the model is often used to estimate the influence of 'quasi'-free-charge carriers (electrons and/or holes) in various semiconductors or isolators, also in the context of estimating the X-ray- or optical-induced transient refractive index [114, 121, 154]. This assumption may be valid because of the metal-like behavior of degenerate semiconductors or semiconductors with a large amount of excited electrons in the conduction band [66, 155]. These electrons can be thought of as free electrons and are therefore well-described by the Drude model.
As discussed in Chapter 3.3.2, the dielectric function of a material within the Drude model is described by Eq. 3.31. The transient refractive index change of the material

after irradiation with an X-ray pulse is then be calculated with

$$n(\omega)^2 = n_0(\omega)^2 - \left(\frac{w_p}{w}\right)^2 \frac{1}{1 + i/\omega\tau_e}\,, \tag{6.9}$$

where $n_0(\omega)$ is the usual frequency-dependent refractive index of the material. The plasma frequency ω_p depends on the effective electron mass. As described in Chapter 3.1, the effective charge carrier masses at the Γ point are characterized by the longitudinal m^*_{el} and transversal m^*_{et} effective electron mass. An effective mass is used in the calculation, which combines the transversal and longitudinal electron masses to $m^*_e = (m_t^2 m_l)^{1/3}$ [156]. The used properties, together with their values and sources, are listed in Tab. 6.1

In collaboration with the theory group of Michael Thorwart of the University of Ham-

Diamond properties used for Drude model	
$n_{0,400nm}$	2.46 [157]
m^*_{el}	1.56 [158]
m^*_{et}	0.28 [158]
ω_{400nm}	4709128918272133 rad/s [calculated]
μ_e	4500 cm^2/Vs [156]
τ_e	1.27 ps [calculated via $\mu_e m^*_e/e$]

Table 6.1.: Diamond properties used to calculate the refractive index change with the Drude model.

burg, we also use the Maxwell Garnett model of dielectric mixtures to estimate the X-ray-induced refractive index change in diamond (Section 3.3.3). A dielectric mixture of the diamond host material and electric polarons is used to calculate the effective dielectric function. The free electrons created by the incident X-ray pulses interact promptly with the bound electrons in the diamond crystal, distort their equilibrium distribution, and polarize the diamond lattice. This electronic polarization cloud screens the free charges and moves with them in the crystal. The free electrons plus the distortions of the bound charges in the lattice form an electronic polaron [88, 89]. The dielectric properties of this quasi-particle differ slightly from those of the diamond host lattice.

For the calculation of the effective dielectric function of a mixture of the diamond host material and the polaron gas, we used the values listed in table 6.2 to solve the Maxwell Garnett formula (Eq. 3.33). The static- and high-frequency polaron dielectric constants $\epsilon_{s,p}$ and $\epsilon_{\infty,p}$ are estimated to have nearly the same values as for the diamond host material. We assume only a small reduction of the overall dielectric constant of 1% within the electronic polaron, since only the fast electrons within the dielectric background will respond on a sub-ps timescale. The polaron relaxation time $\tau_{D,p}$ is also an estimate based on the Debye relaxation time in diamond. Due to the slightly reduced polarization of the distorted charge clouds within the electronic polaron, we estimate an increased Debye relaxation time by about a factor of 1.5 in comparison to that in bulk diamond. An electronic polaron size is around or smaller than the lattice constant. We chose its radius to

Diamond properties used for Maxwell Garnett model	
$\epsilon_{s,d}$	7.85 [160]
$\epsilon_{s,p}$	$0.99\epsilon_{s,d}$
$\epsilon_{\infty,d}$	5.56 [160]
$\epsilon_{\infty,p}$	$0.99\epsilon_{s,d}$
$\tau_{D,d}$	2.48 µs [160]
$\tau_{D,p}$	$1.5\tau_{D,d}$
a	$3.567\text{x}10^{-10}$m [161]
r_p	$0.80a$

Table 6.2.: Properties used to calculate the refractive index change in diamond after carrier injection with the Maxwell Garnett dielectric mixture formula.

about 80% of the lattice constant of diamond. Under these conditions, only nearby electrons are affected, and the crystal lattice itself remains free from major distortion [159]. Again, the dielectric function is used to evaluate the refractive index. The refractive index change is then calculated by subtracting the altered refractive index calculated with the Maxwell Garnett formula from the static frequency dependent refractive index.

The calculated X-ray-induced refractive index change Δn in diamond, calculated with the Drude and Maxwell Garnett formula, is shown in Fig. 6.9. The results are shown over a broad range of calculated electron densities from 10^{16} to 10^{21} electrons/cm^3. Due to the additional screening effects in the Maxwell Garnett model, the refractive index change is nearly two magnitudes smaller compared to the Drude model, where no coupling of the free electrons with the dielectric background is implemented.

A compilation of the X-ray-induced refractive index change for a 50 µm thick diamond sample, calculated for both models is shown in Fig. 6.10. The X-ray-induced electron densities are calculated as described in Section 6.2.1. The estimated refractive index change for X-ray photon energies from 5 keV to 25 keV and X-ray pulse energies ranging from 1 µJ to 2 mJ calculated with the Drude- and Maxwell Garnett model are shown in Fig. 6.10 a)-c) and d)-f), respectively. The X-ray beam profile is assumed to be a flat-top beam profile with beam diameters of d=20 µm (a) and d)), d=100 µm (b) and e)) and d=200 µm (c) and f)). A beam size of d=20 µm represents a timing-tool operated very close to the X-ray focus, while the other two beam diameters of d=100 µm and d=200 µm are describe the operation with more relaxed X-ray intensities. The white parts in Fig. 6.10 a) and d) indicate X-ray conditions where the absorbed X-ray photons would heat the diamond above its melting point as discussed in Section 6.1 and Fig. 6.8.

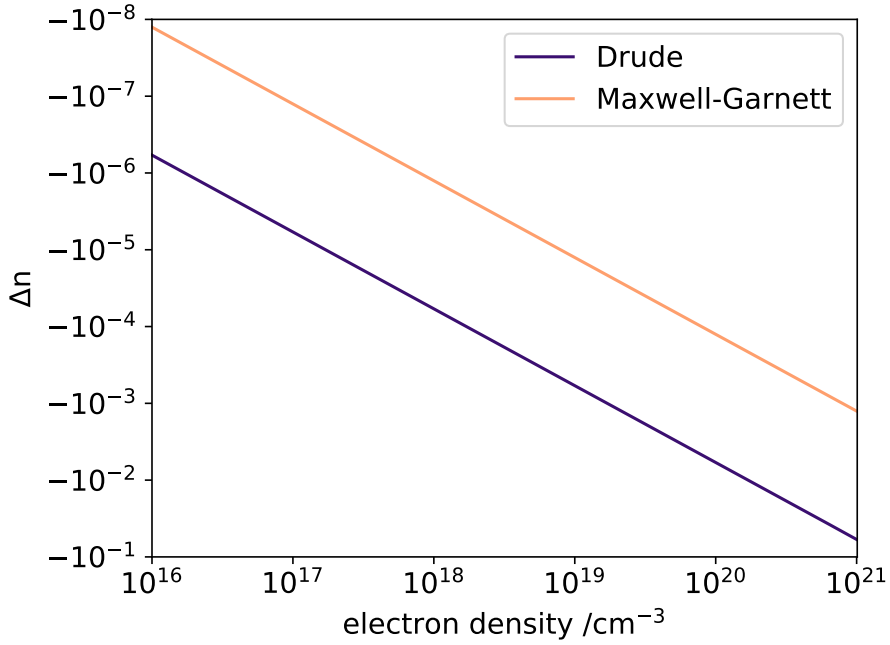

Figure 6.9.: Refractive index change in diamond due to carrier injection, calculated with the Drude model (blue) and Maxwell Garnett dielectric mixture theory (orange).

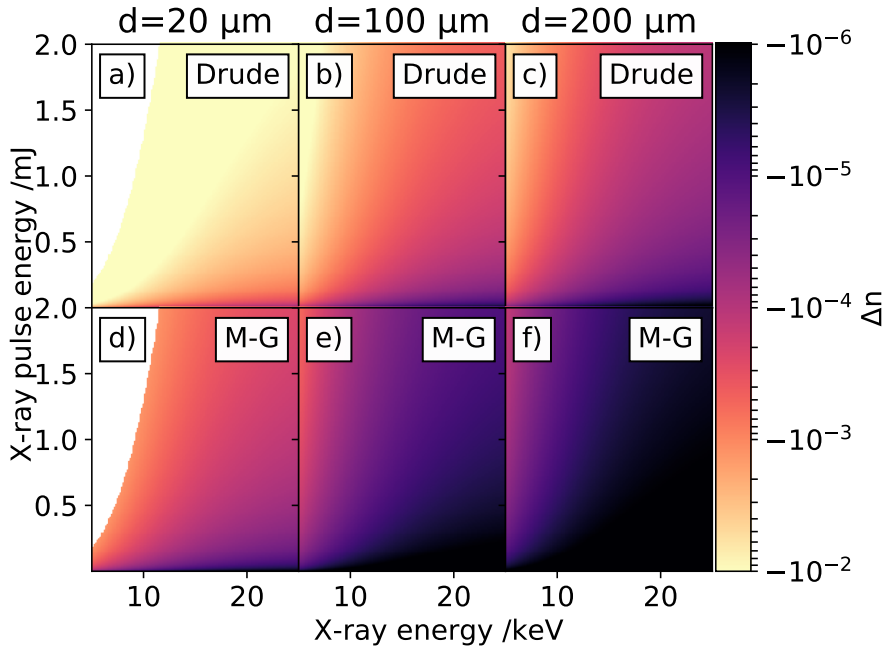

Figure 6.10.: X-ray-induced refractive index change in a 50 µm thick diamond, calculated for different X-ray photon energies and X-ray pulse energies, using the Drude (top row) and Maxwell Garnett (bottom row) models. Each column is calculated for different X-ray beam diameters (flat-top shape) of 20 µm, 100 µm and 200 µm.

6.2.3. Self-Referenced Timing-Tool Simulation

The expected self-referenced timing-tool signal with an imprinted X-ray arrival-time is simulated by combining the previously discussed methods. In this simulation, a laser pulse is propagated through the entire optical setup (Chapter 5). The interaction with the relevant optical elements and the interaction with the X-ray pumped sample are simulated by linear propagation of the laser pulse through each element. As described in Section 3.4.1, the linear propagation of a laser pulse through a dispersive material adds an additional phase to the original pulse and is implemented in the simulation up to the third order. The optical properties used for the different materials are listed in Tab. 6.3 and are directly derived from the Sellmaier equation of the individual materials [162, 163, 164].

The starting point of the simulation is a bandwidth-limited laser pulse with a Gaussian-shaped spectrum in the frequency domain, described through

$$|E(\omega)|^2 = A_0 \mathrm{e}^{-\frac{w-{w_0}^2}{4\Delta\omega\sqrt{2\ln 2}}}, \tag{6.10}$$

with the maximum amplitude A_0 at the central angular frequency w_0 and the FWHM of the spectral envelope $\Delta\omega$. The experimentally accessible physical observable in the frequency domain is the measurement of an optical spectrum $|E(\omega)|^2$, which is directly connected to the optical wavelength via $\lambda = 2\pi c/\omega$. The simulation results are shown in wavelength units.

The central frequency of the Gaussian pulse is set to $\omega_0 = 749.48\,\mathrm{THz} \times 2\pi$ (400 nm) and an FWHM bandwidth of $\Delta\omega = 93.69\,\mathrm{THz} \times 2\pi$ (50 nm). These values replicate roughly the laser conditions of the experimental data in Chapter 7. The spectrum of the bandwidth-limited pulse is shown in Fig. 6.11 a) and the corresponding time-dependent oscillating electric field in b) in blue. The inset in b) is an enlarged illustration of the oscillating time-dependent electric field of the bandwidth-limited pulse with a corresponding pulse duration of around 5 fs FWHM.

The X-ray arrival-time jitter is expected to be on the timescale of up to a few hundred of fs (Tab. 4.2). Therefore, the bandwidth-limited pulse is transmitted through a transparent dispersive medium, which chirps the pulse and stretches it in time, increasing

Material values used for timing-tool simulation at 400 nm								
Material	$\mathbf{n}_o$	$\mathbf{n}_e$	$\mathbf{GV}_o$ [c]	$\mathbf{GV}_e$ [c]	$\mathbf{GVD}_o$ [fs^2/mm]	$\mathbf{GVD}_e$ [fs^2/mm]	$\mathbf{TOD}_o$ [fs^3/mm]	$\mathbf{TOD}_e$ [fs^3/mm]
Fused-silica [162]	1.47		0.66		97.67		30.38	
BBO [163]	1.69	1.57	0.56	0.61	211.49	148.21	77.09	50.63
Diamond [164]	2.46		0.38		435.04		151.87	

Table 6.3.: Dispersion values of the used materials in the self-referenced timing-tool simulation. Values are deduced from the materials Sellmaier equation.

the accessible timing window. Propagating the bandwidth-limited pulse through a 1 cm thick fused-silica block chirps the pulse to an FWHM pulse duration of around 540 fs. The chirped pulse spectrum and its electric field are shown in orange in Fig. 6.11 a) and b), respectively. The spectrum of the chirped pulse is identical to the spectrum of the bandwidth-limited pulse, but its electric field is stretched in time. Therefore, its amplitude is decreased to preserve the pulse energy. In addition, the propagated optical pulse is temporally delayed by the group delay

$$\tau = \frac{d}{v_g}, \tag{6.11}$$

where d is the thickness of the dispersive medium and v_g the group velocity of the material. Using the second term of Eq. 3.47, the group velocity can be calculated with

$$v_g = \left. \frac{c}{n(\omega) + \omega \frac{\mathrm{d}n}{\mathrm{d}\omega}} \right|_{\omega=\omega_0}, \tag{6.12}$$

where c is the vacuum speed of light, n the refractive index of the dispersive material at the angular frequency ω_0 and $\mathrm{d}n/\mathrm{d}\omega$ the derivative of the refractive index at the central angular frequency ω_0.

The group delay is then sufficiently handled in the simulation by applying all dispersive effects from Eq. 3.47 up to the 3rd order. For visual reasons, all pulses are then time-shifted to be centered at t = 0 fs in all the following graphs.

The actual instantaneous wavelength, derived from the instantaneous frequency (Eq. 3.42) of the original bandwidth-limited optical pulse and the propagated chirped pulse, are shown as purple lines in Fig. 6.11 b) with the corresponding y-axis on the right-hand side. By propagating the optical pulse through the fused-silica glass block, the resulting optical pulse has a linear chirp in the frequency domain, arranging all frequencies (wavelengths) in time.

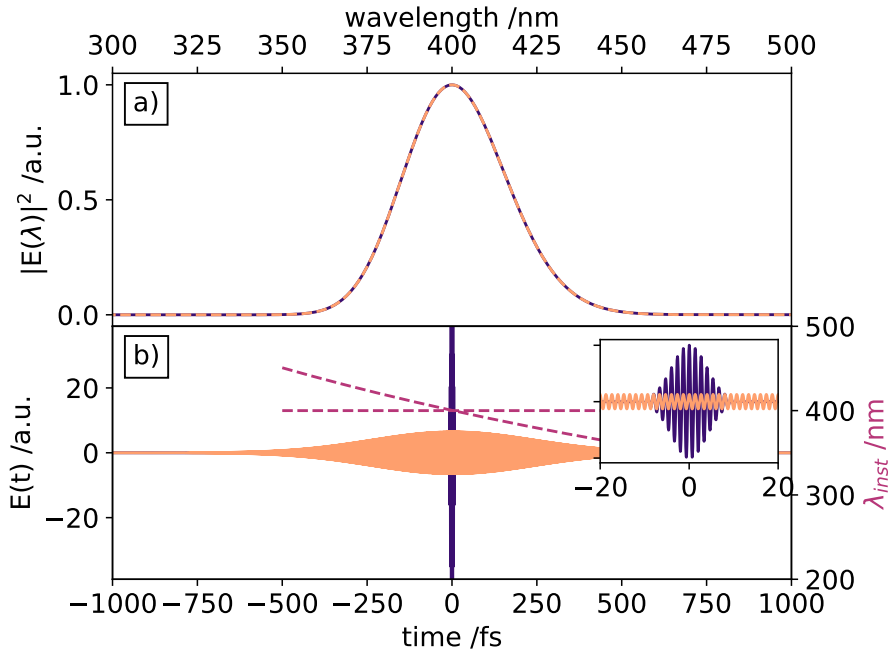

Figure 6.11.: Initial bandwidth-limited pulse (blue), which is used as a starting point for the timing-tool simulation. The optical pulse after linear propagation through 1 cm of fused-silica is shown in orange. The spectra of both pulses are shown in a). The time-dependent oscillating electric fields are shown in b). The purple lines show the instantaneous wavelength over the pulse duration of the two pulses. The bandwidth-limited pulse is unchirped, while the longer pulse has a linear chirp. The inset shows the bandwidth-limited pulse at an enlarged timescale of $\pm$20 fs.

The Common-Path-Interferometer

The most important part of the setup is the CPI. The CPI is responsible for extracting a background-free timing signal, thus, a zero signal in the absence of any X-ray-induced transient refractive index change in the sample and a background-free positive signal in the presence of an X-ray pulse. The influence of the alignment of both a-cut αBBO crystals is simulated next. The alignment of these two crystals must be very precise to yield a pure self-referenced. Two 5 mm thick a-cut αBBO crystals are used as BC in this simulation. The thickness influences the temporal separation of the two PC after the first BC and the synchronization of both PC behind the second BC.

The initial polarization of the bandwidth-limited optical pulse is set to 45° (see Fig. 5.2). The propagation of this initial bandwidth-limited pulse through the fused-silica block does not alter its polarization. The simulation neglects the weak laser-induced [165, 166] and stress-induced birefringence [167] in fused-silica, which could alter the polarization of the transmitted pulse, resulting in a weak elliptical contribution to the polarization. Assuming an ideal polarizer, the optical pulse is transmitted through the first polarizer without decreasing its amplitude since its polarization matches the transmission axis of the polarizer of 45°.

This optical pulse is transmitted through the first BC and is further chirped due to the dispersive properties of the BC. The birefringence of the crystal separates the optical pulse into two time-delayed horizontal $E_x(t)$ and vertical $E_y(t)$ PCs. When the optical axis of the BC is aligned to be exactly horizontal, equal portions of the original pulse will be split into the two PC. The temporal separation is determined by the thickness d of the BC and its ordinary and extraordinary refractive indices. The temporal separation $\Delta\tau$ of the two PC can be calculated to

$$\Delta\tau = d\left(\frac{1}{v_{go}} - \frac{1}{v_{ge}}\right), \tag{6.13}$$

with v_{go} and v_{ge} being the ordinary and extraordinary group velocities, respectively. With the chosen BC thickness, the two PC are almost completely separated in time and only residually overlap in their tails (Fig. 6.12). The horizontal PC, experiencing the extraordinary refractive index n_e of the BC, travels faster through the crystal than the vertical PC, experiencing the ordinary refractive index. In the region where both PCs overlap in time, the coherent sum of both PC delivers many different polarization states, depending on the phase shift between both PCs. This is shown in the inset of a), where the viewing angle is rotated such that the viewing angle is now along the time axis. This view reveals the major contribution of the horizontal and vertical PCs, but also the changing polarization in the region where both PCs are temporally overlapped. Due to the chirped nature of the optical pulse, the overlapped tails of both PC can not be in phase with each other, thus, generating a manifold of different polarizations. The spectrum of the optical pulse after transmission through the first BC is still identical to the original spectrum (Fig. 6.12 b). The extraordinary $E_x(t)$ (blue) and ordinary $E_y(t)$ (orange) PCs are shown in subfigure c). Due to the differences (Tab. 6.3) of the ordinary and extraordinary refractive indices, both PCs accumulate different amounts of additional phase, when transmitted through the first BC. Thus, they are not only temporally separated, but also their chirp and amplitudes are different.

By transmitting the time-sheared PCs through the second BC, both PCs can be temporally overlapped again. To synchronize both PCs precisely, the optical axis of the second BC needs to be rotated by 90° with respect to the optical axis of the first BC. In this particular case, the optical axis of the second BC needs to be vertical. Both pulses are propagated through the second BC, but now using the dispersive optical properties of the other crystal axis for each PC. The leading optical pulse, which was propagated through the first BC using the extraordinary refractive index, is now propagated through the second BC using the ordinary refractive index. The trailing optical pulse is accordingly propagated through the second BC, using the optical properties arising from the extraordinary refractive index. This perfectly overlaps both PCs in time again, since both optical pulses were propagated through the two BC, accumulating an identical amount of additional phase (Fig. 6.13).

When the temporal overlap is achieved with interferometric precision, meaning without

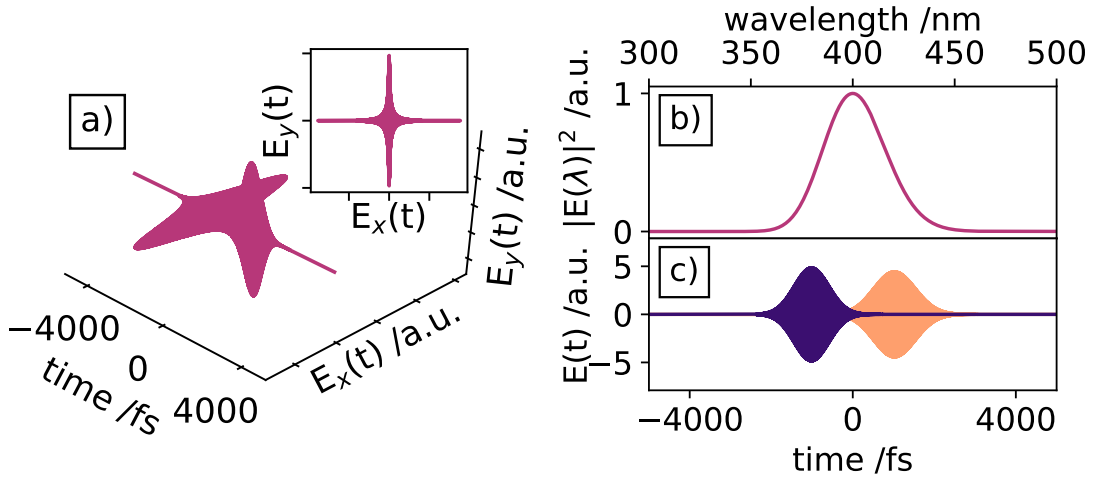

Figure 6.12.: Optical pulse after transmission through the first BC. The birefringence of the crystal creates a time-sheared pulse with a horizontal and vertical PC, shown in a). The overlapping tails create a circular polarization, illustrated in the inset of a). The spectrum of the time-sheared pulse is shown in b). The horizontal (blue) and vertical (orange) PC are shown in c).

any phase shift between both PCs, the same 45° polarization as before the first BC is recovered (Fig. 6.13 a). The inset displays the same situation along the time axis, confirming the perfect linear polarization when both PCs are exactly in phase after the second BC. The spectrum remains still unchanged (Fig. 6.13 b). The two temporally overlapped PCs are illustrated in c), while the inset shows more details in the region from -5 fs to 5 fs. Behind the second BC, the optical pulse arrives at the second polarizer. Since the second BC synchronized both pulses in time again and recovered the original 45° polarization, no light is transmitted through an ideal polarizer, as designed for a background-free setup. If the two PCs are not perfectly synchronized, a fraction of the resulting elliptical polarization is then transmitted through the polarizer. The polarizer transmission is calculated by using the Jones-Formalism [168].

The most sensitive part of the CPI is the alignment of the second BC, which overlaps the two PCs again with interferometric precision. To achieve a perfect synchronization between the two components, the thickness of the second crystal needs to be identical to the first one. By rotating the second BC around its optical axis, the effective thickness of the crystal can be changed, eventually matching the thickness of the first BC. The additional thickness by rotating the crystal is given by $\Delta d = d\cos^{-1}(\theta) - d$, where d is the thickness and θ the rotation angle. Assuming both BCs have the same thickness, the phase difference between both PCs through an additional rotation around the optical axis of the second BC is given by

$$\Delta\phi = \frac{2\pi|n_o - n_e|}{\lambda_0}\Delta d \ , \tag{6.14}$$

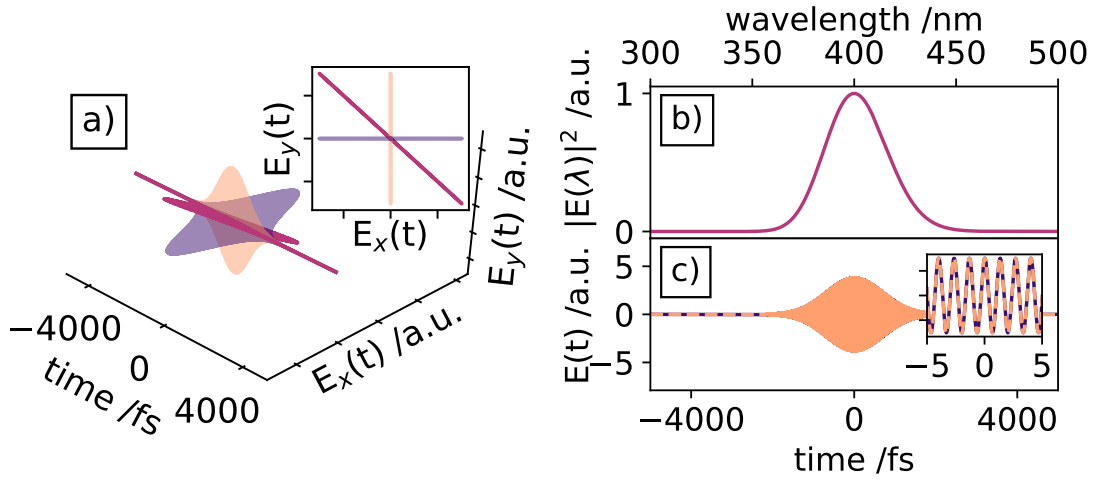

Figure 6.13.: Optical pulse after propagation through the second BC. Both PC (blue and orange) are exactly overlapped in time, recreating the original 45° polarized optical pulse (purple) in a). The inset shows the same 3d-illustration along the time axis, proofing the linear polarization of the coherent sum (purple) of both PC. The spectrum of the optical pulse is shown in b), while the horizontal (blue) and vertical (orange) temporal overlapped PC are shown in c).

where n_o and n_e are the ordinary and extraordinary refractive indices, λ_0 the central wavelength of the optical pulse and Δd the additional thickness caused by rotation of the crystal around its optical axis.

The influence of the rotation angle of the second BC on the transmitted optical light through the second polarizer is shown in Fig. 6.14. The rotation angle of the second BC around its optical axis is shown on the y-axis, which is directly correlated to the phase shift between the horizontal and vertical PCs of the optical pulse after transmission through the second BC. In this case, the phase shift describes the shift between the central wavelength of the two PCs. Only for a very narrow angle around 0°, almost perfect synchronization between the two PCs can be achieved, recreating the original 45° polarization. In this case, the second polarizer will not transmit any light. Only when the thickness of both BCs are exactly identical, the original linear polarization is retained. For all other rotational angles of the second BC, some spectral parts of both PCs are always out of phase, thus, a pure linear polarization can no longer be generated. With increasing rotation angles, the resulting spectra become increasingly complex. Due to the chirp and Gaussian-shaped envelope of the PC, integer multiple phase shifts between the two PCs at the central wavelength can not recreate a linear polarized pulse in all wavelength regions.

Four special phase shift angles between both PCs are displayed in Fig. 6.15: 0° (a), 90° (b), 180° (c) and 360° (d). The top row illustrates the coherent sum of the linear and vertical PC from a 45° viewing angle. In the middle row, the point of view is rotated, and the viewing angle is along the beam propagation axis, such that the polarization of the

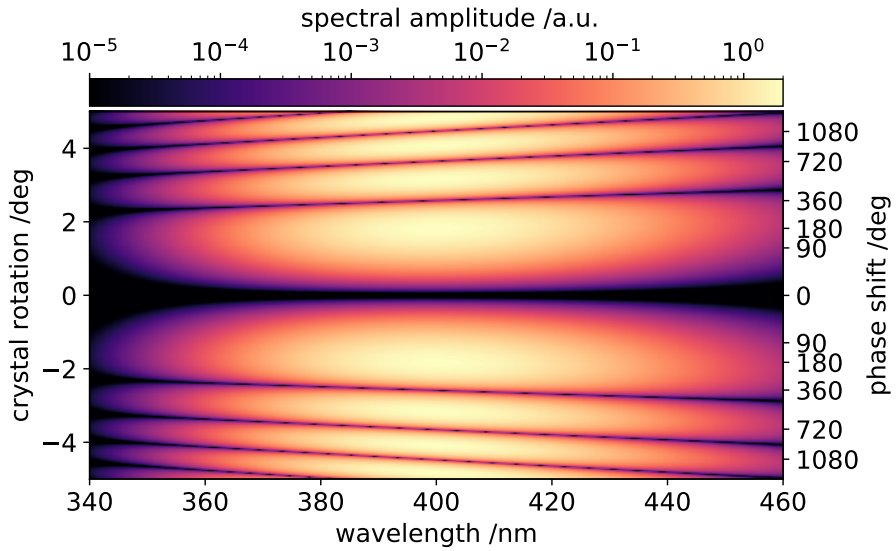

Figure 6.14.: Optical spectra behind the second polarizer for different rotation angles of the second BC. The original 45° polarization is only recreated for a very narrow rotational angle range around 0° of the second BC. All other rotation angles lead to a phase shift between both PC, thus, parts of the resulting optical pulse with the coherent sum of both PCs are always elliptically polarized.

optical pulse becomes clearly visible. The dashed line indicates the transmission angle of the second polarizer. The bottom row shows the spectrum of the optical pulse behind the second polarizer. The first column in Fig. 6.15 represents the desired alignment of the second BC. The phase shift between the horizontal and vertical PC is exactly 0°, meaning that both PCs are perfectly synchronized. Subfigures a) and a') show this linear 45° polarization. The transmission axis of the polarizer, shown as the dashed line in a') is perpendicular to the polarization, thus, no light is transmitted through the polarizer and no spectral intensity can be measured as shown in a''). A 90° shift between the two PCs delivers a circular polarization (Fig. 6.15 b and b') and yields a measurable spectrum behind the second polarizer with half the amplitude of the original optical pulse Fig. (6.15 b''). A 180° phase shift between the two PCs (Fig. 6.15 c and c') results in an almost linear polarization, but shifted by 90°. Due to the broad Gaussian-shaped spectrum of the chirped optical pulse with multiple frequencies, the phase shift of 180° can only be fulfilled at one single wavelength, thus, other spectral regions have deviating phases, resulting in elliptical contributions. Due to the 90° rotation of the (almost) linear polarization, nearly the complete original pulse is transmitted through the second polarizer (Fig. 6.15 c''). A full 360° phase shift of the two PCs at their central frequency recreates the original 45° polarization only at exactly the central wavelength. All other spectral parts of the chirped PC will be shifted more than 360° (lower wavelengths) or less than 360° (higher wavelengths) (Fig. 6.15 d and d'). Spectral regions with elliptical polarization are partially transmitted through the second polarizer, while at the central wavelength

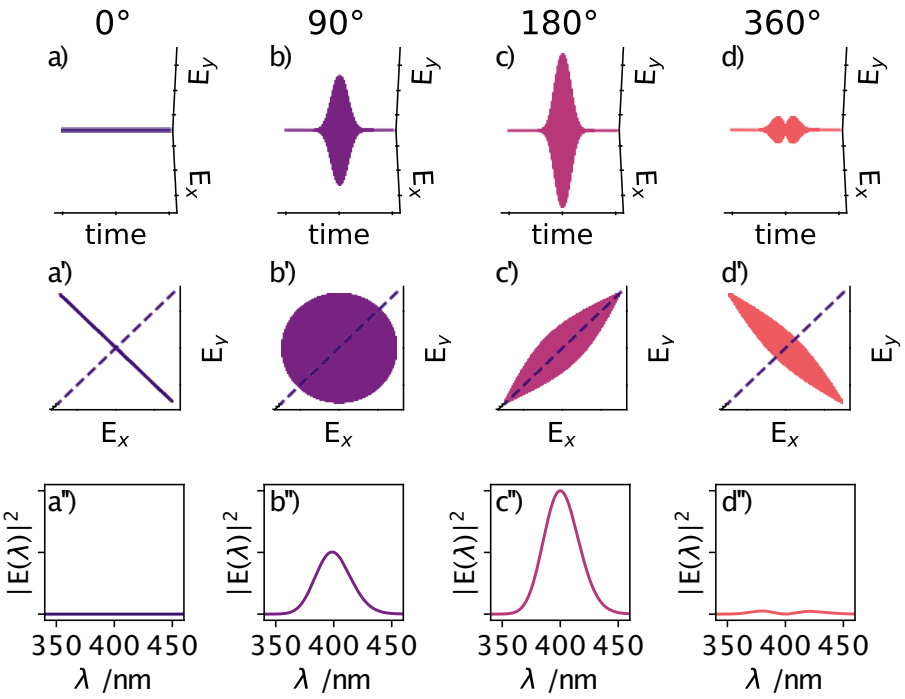

Figure 6.15.: Polarization after the second BC and spectra after the second polarizer for phase shifts of 0° (a), 90° (b), 180° (c) and 360° (d) between the two PC. The first row (a,b,c,d) illustrates the coherent sum of the PC from a 45° view, such that the 45° polarization in a) appears as a line. The second row shows the same polarized pulse from an angle along the time axis, clarifying its polarization. The transmission axis of the second polarizer is indicated by the dashed lines in each subfigure. The third row displays the observable spectrum of the optical pulse after the second polarizer.

of 400 nm, no light is transmitted (Fig. 6.15 d"). All other spectral regions have elliptic polarization contributions, leading to a measurable spectral intensity. Increasing the phase shift between the two PCs beyond 360° further increases the elliptic polarization contribution, which in turn increases the amplitude of the transmitted spectral regions. In addition, when the phase is shifted by multiple integers of 360°, the resulting optical pulse has multiple spectral regions where the polarization is linear polarized to 45°, but also more spectral regions where the optical pulse is almost linear polarized to -45° (Fig. 6.16).

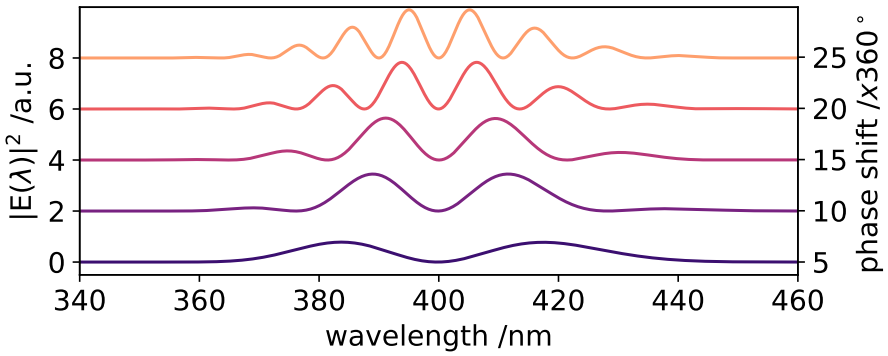

Figure 6.16.: Spectra behind the second polarizer for different large phase shifts between the two PC. The phase shifts between the two PC is indicated on the right y-axis as integer $\times$ 360°.

Implementing the X-Ray-Induced Transient Refractive Index

After having simulated the fundamental principle of the CPI in the subsection above, we introduce the diamond sample in the center of the CPI and simulate the influence of the X-ray-induced refractive index change on the transmitted optical pulse. The X-ray pulse imprints its arrival-time by changing the refractive index of the sample due to the additional created free electrons in the conduction band (Section 3.2.1 and Section 6.2.1).
Three effects, directly influenced by the transient refractive index change, are implemented in the simulation: i) change of the sample reflectivity, ii) phase change of the transmitted optical pulses and iii) the transient change of the sample absorption. Assuming normal incidence for the optical pulse on the diamond sample, the X-ray-induced change of reflectivity is calculated using the Fresnel formula [169]

$$\Delta R = \left|\frac{n_1 - n_2}{n_1 + n_2}\right|^2 - \left|\frac{n_1 - (n_2 + \Delta n)}{n_1 + (n_2 + \Delta n)}\right|^2 , \tag{6.15}$$

with n_1 the refractive index of air, n_2 the refractive index of diamond at the central wavelength of the optical pulse and Δn the X-ray-induced transient index change. The simulations from Section 6.2.1 suggest a transient refractive index change somewhere in the region of Δn = -10^{-6} to -10^{-4}. For Δn = -10^{-4}, the reflectivity R would decrease by $\Delta R/R = 0.0014\%$, leading to a very small increase in transmission. From the refractive index simulations, the imaginary part can be used to calculate the expected absorption change via the extinction coefficient and the Beer-Lambert law. For Δn = -10^{-4}, the expected absorption change is $\Delta A = 0.002\%$ at a central wavelength of $\lambda_0 = 400$ nm. Reflectivity and absorption influence the amplitude of the transmitted electric field.
The X-ray-induced phase change of the transmitted optical light is calculated with

$$\Delta\phi = \frac{2\pi\Delta n}{\lambda_0} d , \tag{6.16}$$

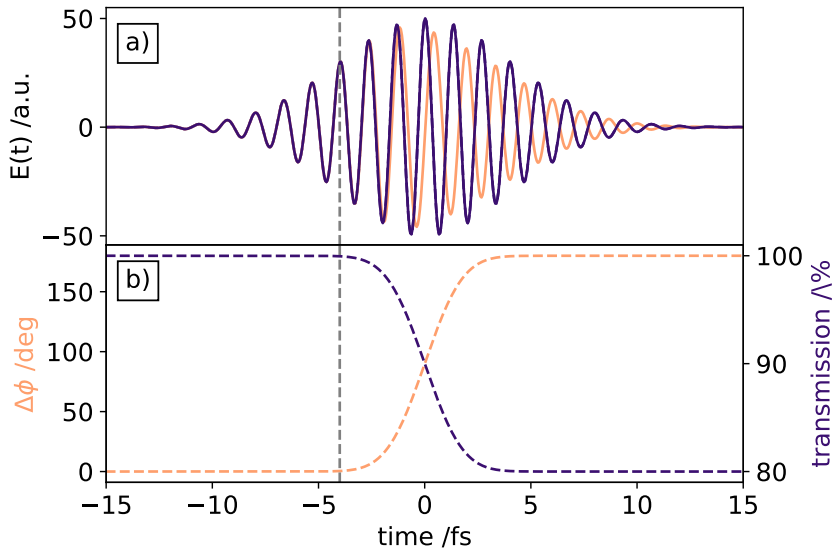

Figure 6.17.: Example of the phase ($\Delta\phi = 180°$) and amplitude ($\Delta A = 20\%$) manipulation of an electric field, shown in a). The blue curve shows the unperturbed oscillating electric field, while the orange curve shows this electric field with an amplitude and phase change, gradually applied over 2 fs FWHM period. The phase change (orange) and transmission change (blue) are shown in b). The gray vertical line indicates the onset of the phase and amplitude manipulation.

where Δn is the X-ray-induced transient refractive index change and d the thickness of the diamond sample. With $\Delta n = -10^{-4}$, the expected phase change is $\Delta\phi = 4.5°$, and we will see that this will give rise to a strong self-referenced arrival-time signal.

The X-ray-induced secondary electron cascade is not instantaneous and needs timescales between 10 fs to 100 fs in diamond (Section 3.2.1). Thus, the transient refractive index change occurs on the same fs-timescale. To apply this non-instantaneous change of the sample reflectivity, absorption and optical phase shift, a Gaussian error function of the form

$$f(t) = \frac{A}{2}\left[1 + \mathrm{erf}\left(\frac{t - t_0}{2\sigma\sqrt{2\ln 2}}\right)\right] \tag{6.17}$$

is used to describe the change of the X-ray-induced electron density and the following transient change of the optical properties. A is the amplitude, t_0 the center of the error function and σ the FWHM of the time-dependent rise of the function. The amplitude change is easily calculated by multiplying the electric fields of the individual PCs with an appropriately normalized $f(t)$, where the amplitude defines the amplitude change in percent. The onset of the error function marks the X-ray arrival-time. The FWHM of the error function is set to match the time duration of the X-ray-induced electron cascade, i.e., between 10 fs and 100 fs, depending on the incident X-ray photon energies.

The X-ray-induced phase change can not be applied by simple manipulation of the elec-

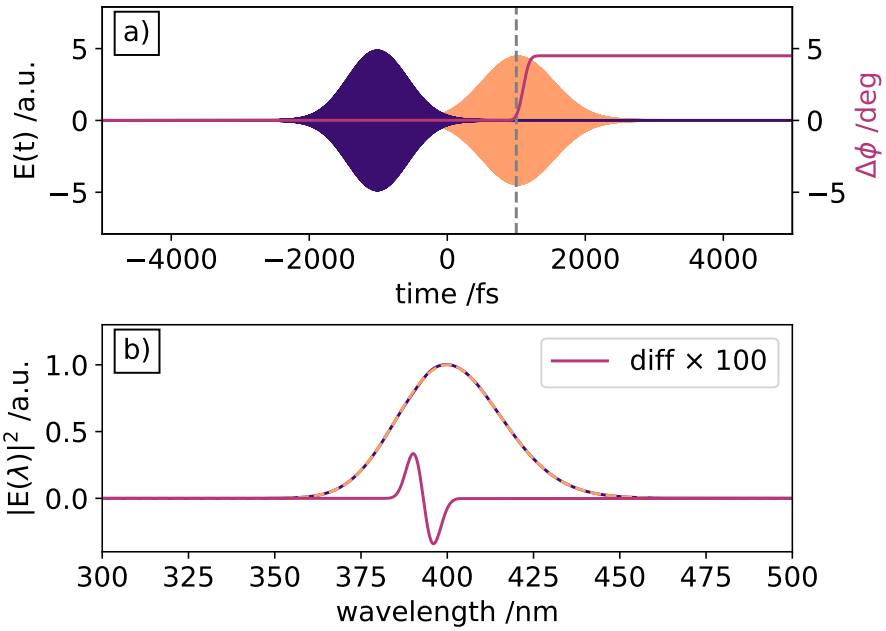

Figure 6.18.: X-ray-induced phase and transmission change in diamond, applied to the two PC of the optical pulse. In subfigure a) the leading horizontal (blue) and trailing vertical (orange) PC are shown. While the leading component is unaffected by the X-ray-induced phase change (purple), the trailing PC is only partially transmitted before the X-ray pulse arrives and causes a phase shift of the transmitted light. The spectra of the optical pulse before (orange) and after (blue) the transmission through the diamond is shown in b). The difference between both spectra is shown in purple and multiplied by 100.

tric field amplitude. The error function $f(t)$ is used to gradually change the phase of the oscillating electric field. To change the phase of both oscillating electric fields of the two PCs, both are multiplied with a phase-transfer function. The electric field with the phase change is obtained with

$$E_{\Delta\phi}(t) = E(t)\,\mathrm{e}^{-i\,f(t)}\,, \tag{6.18}$$

where $E(t)$ is the initial electric field and $f(t)$ the error function, described in Eq. 6.17. In this case, the error function describes the time-dependent X-ray-induced phase. Before the X-ray pulse arrives, the error function is set to zero. The amplitude of $f(t)$ defines the actual phase shift and is applied to the original oscillating electric fields of each of the two PCs.

An example of the amplitude and phase manipulation of the electric field at a certain point in time is shown in Fig. 6.17. The X-ray arrival-time is indicated by the gray dashed vertical line, which marks the onset of the phase and amplitude manipulation.
The actual application of the X-ray-induced amplitude and phase change in the diamond sample shows both orthogonal PCs and the time-dependant X-ray-induced phase shift (6.18 a). The electric fields of the two orthogonal PC and the time-dependent X-ray-

induced phase shift are shown in a). The leading horizontal PC passes through the diamond without experiencing any X-ray-induced refractive change. The arrival-time of the X-ray pulse and the subsequent X-ray-induced refractive index change is set to t_0=1000 fs, overlapping with the trailing vertical PC (gray line Fig. 6.18). The X-ray-induced refractive index change is set to $\Delta n = -10^{-4}$, yielding a nearly invisible change of the transmission $\Delta T \ll 0.01\%$, consisting of the X-ray-induced reflectivity change and X-ray-induced absorption due to the changed imaginary part of the diamond refractive index. The X-ray-induced phase shift is $\Delta\phi = 4.5^\circ$ (Eq. 6.16). The width of the error functions, applying the amplitude and phase manipulation, is set to 75 fs, according to the electron cascading times, estimated for X-ray photon energies of 10 keV (Section 3.2.1). The trailing (orange) PC is only partially transmitted through the diamond sample before the X-ray pulse arrives. Thus, parts of it are transmitted without experiencing the optical transmission and phase change, while the later fraction is subject to the X-ray-induced amplitude and phase change. The error function used to apply the phase change is shown in purple. The spectra of the optical pulse before the X-ray-induced refractive index change (blue) and after the X-ray-induced refractive index change is shown in Fig. 6.18 b). The difference between both spectra is shown in purple and multiplied by a factor of 100. The difference signal is dominated by an AC-like signal, originating from the X-ray-induced phase shift, which is a change of the instantaneous frequency in the temporal domain (Appendix A).

After applying the X-ray-induced amplitude and phase change to the optical pulse, it further propagates through the second BC which synchronizes both PCs in time again (Fig. 6.19 a). The viewing angle is again from 45°, to visualize the elliptical contribution to the coherent sum (purple) of both PCs. The first part of the optical pulse is linearly polarized at 45°, because both PC can be perfectly synchronized. Due to the X-ray-induced refractive index change in the diamond sample during the transmission of the trailing PC, all following parts can not be synchronized with the corresponding parts of the leading PC and lead to the elliptical polarization. The inset shows an alternative viewing angle along the beam propagation axis, clarifying the polarization states of the involved components. The horizontal (blue) and vertical (orange) PCs are linear polarized, coherently summing up to recreate the purple optical pulse with elliptical polarization contributions. The blue dashed line indicates the transmission axis of the second polarizer.

Both PCs are transmitted through the second polarizer, using the Jones formalism. As a result, the synchronized parts of both PCs are completely suppressed behind the polarizer, while parts of the elliptical polarization are transmitted. In the transmitted spectrum, a clear cut-off can be observed, indicating the X-ray arrival-time and the transition between the synchronized (blocked) and unsynchronized (partially transmitted) spectral parts of both PCs (Fig. 6.19 b). The higher wavelengths in the region above 410 nm are the part of the optical pulse whose polarization was linearly polarized to 45° before the second polarizer. The cut-off region around 400 nm is the spectral region, where the X-

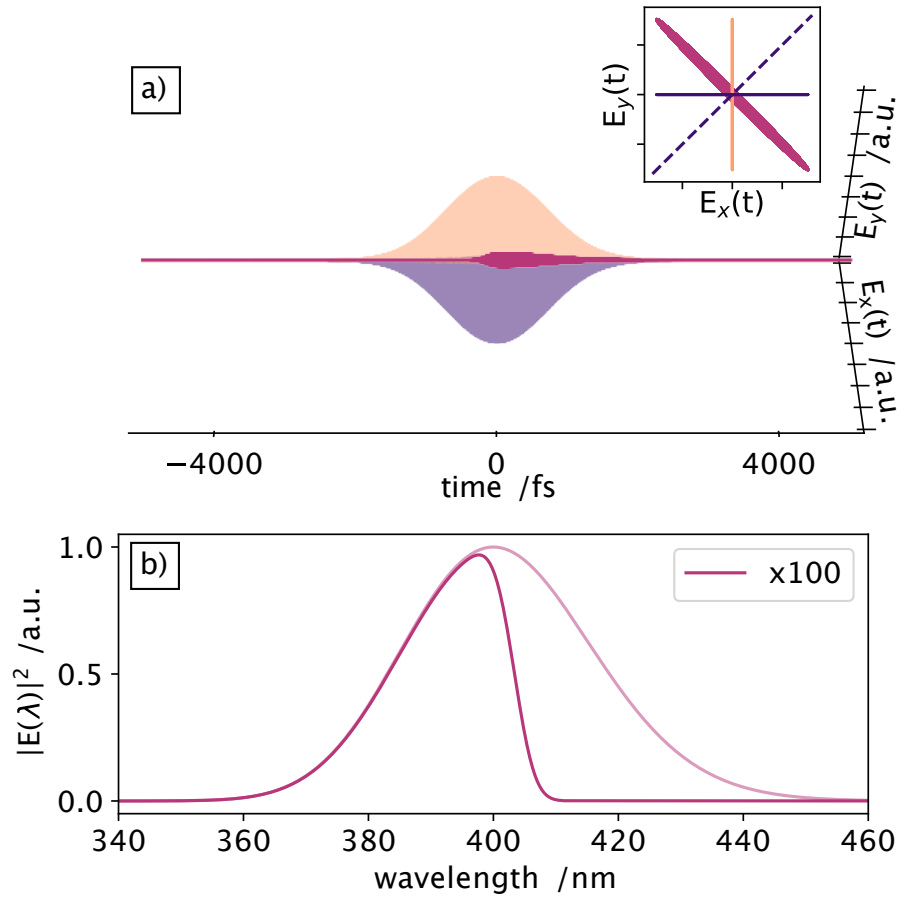

Figure 6.19.: X-ray-induced self-referenced arrival-time signal. The horizontal (blue) and vertical (orange) synchronized PC and the resulting optical pulse (purple) is shown in a) from a sideward 45° viewing angle. The inset shows the same situation from a different angle along the time axis as well as the transmission axis of the second polarizer (dashed blue line). The optical spectrum after transmission through the second polarizer is shown in b) and is multiplied by 100 to match the original spectrum of the optical pulse (shaded purple line).

ray pulse altered the phase of the trailing PC (see Fig. 6.18 a) and consequently changes the polarization after synchronizing both PCs from linear to elliptical.

The position of this cut-off region depends on the arrival-time of the X-ray pulse and its temporal overlap with the two PCs (Fig. 6.20). From -1500 fs to -500 fs, the X-ray pulse temporally overlaps with the leading horizontal PC, while between 500 fs and 1500 fs the X-ray pulse overlaps with the trailing vertical PC. For relative arrival-times between -500 fs and 500 fs, the X-ray pulse arrives midway the two PC, such that the entire first PC is not affected by the changed refractive index and the entire trailing PC is experiencing the changed refractive index.

The cut-off region, indicating the X-ray arrival-time, moves from higher wavelengths to lower wavelengths for increasing X-ray arrival-times. This is caused by the linear chirp of the optical pulse. Overlapping the X-ray pulse with the leading PC (-1500 to -500 fs)

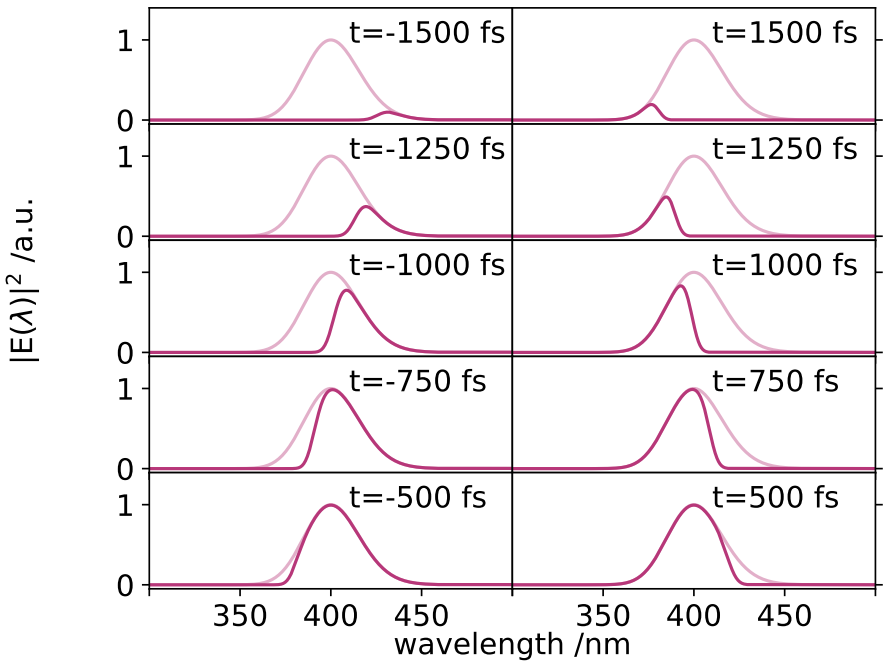

Figure 6.20.: Self-referenced arrival-time spectra for different X-ray arrival-times from -1500 fs to 1500 fs. The arrival-time spectra are shown as solid purple lines, while the full original spectra are indicated by the shaded purple lines. The arrival-time spectra are multiplied by a factor 100 to match the original spectral amplitude.

causes the zero signal to be in the lower wavelength regions and the non-zero part of the arrival-time signal in the higher wavelength regions. The reason is that the first part of the leading pulse is transmitted before the X-ray-induced refractive index change, while later parts of the leading PC and the whole trailing PC experience the refractive index change. These spectral parts of both PC can coherently recreate the original 45° polarization, thus, these spectral parts can not transmit through the second polarizer, generating a zero-signal in these wavelength regions. The spectral parts of the leading PC, which were transmitted through the diamond sample before the X-ray pulse changed the refractive index, experienced a different refractive index of the diamond sample compared to the same spectral parts of the trailing PC. Therefore, these parts are not perfectly synchronized after the second BC and the resulting polarization of the optical pulse has an elliptical component, which is partially transmitted through the second polarizer. Overlapping the X-ray pulse with the trailing PC creates a similar mirrored arrival-time spectrum. The non-zero part of the spectrum is now in the lower wavelength region, while the zero-signal part is now in the higher wavelength region. In this situation, only the spectral parts of the trailing PC, arriving after the X-ray pulse, experience a changed refractive index, thus, these spectral parts can not recreate the original 45° polarization of the optical pulse.

Influence of Temporal Polarzation Component Separation and Pulse Duration

Another experimental parameter, which influences the result of the arrival-time measurement, is the actual temporal separation of the orthogonal PCs with respect to the chirped pulse width. The temporal separation of the two PCs is defined by the thickness of the first BC and the difference between its ordinary and extraordinary refractive index. The pulse duration of the chirped optical pulse is controlled through the thickness of the transmissive glass block and the thickness of the first BC, together with their respective refractive indices. Three possible temporal configurations of the chirped optical pulses are discussed (Fig. 6.21). To illustrate the influence of the chirp of the optical pulse on the observed X-ray arrival signal, two different X-ray arrival-times are simulated with a temporal separation of 150 fs (indicated by the dashed purple line for the early X-ray pulse and solid purple line for the later X-ray pulse). The arrival-time spectra in the right column are multiplied by a factor of 100 to match the original spectrum, indicated as the shaded purple line. The cut-off edge position of the arrival-time spectra are marked by solid circles.

In the first configuration, both PCs are completely separated from each other, essentially generating two individual 1.5 ps long optical pulses with a temporal separation of 2.3 ps (Fig. 6.21 a). The two X-ray pulses are temporally overlapped with the trailing (orange) PC and introduce the transient refractive index change. These yield the arrival-time spectra shown in Fig. 6.21 a*).

Next, the pulse duration of the chirped optical pulse is increased, yielding temporally overlapping PCs (Fig. 6.21 b). The thickness of the BC is the same, such that the separation is identical. The two, 150 fs separated, simulated X-ray pulses are again overlapped with the trailing PC, but now the first PC overlaps as well with the X-ray pulses in its tail. Therefore, there are identical spectral regions affected by the X-ray pulse, both before and after its arrival. These spectral regions are synchronized behind the second BC. In between these two regions, spectral parts of the leading PC experienced the static refractive index, while the same spectral region of the trailing PC experienced the X-ray-altered refractive index. These spectral parts are not synchronized behind the second BC and yield an elliptic polarization. The leading and trailing parts of both PCs, which are synchronized behind the second BC, recreate the original 45° polarization and are blocked by the second polarizer. Thus, with temporal overlapping PCs and X-ray pulses arriving in this overlapping region, an X-ray arrival-time signal with two distinct cut-off regions is created, both equally imprinting the relative arrival-time of the X-ray pulse (Fig. 6.21 b*).

The two cut-off edges are more pronounced if the original optical pulse is chirped further, while keeping the temporal separation between the two PCs the same (Fig. 6.21 c). Hence, the width of the spectral region where both PCs experienced different refractive indices decreases. Therefore, the bandwidth of the arrival-time signal between the two cut-off edges is getting smaller (Fig. 6.21 c*)

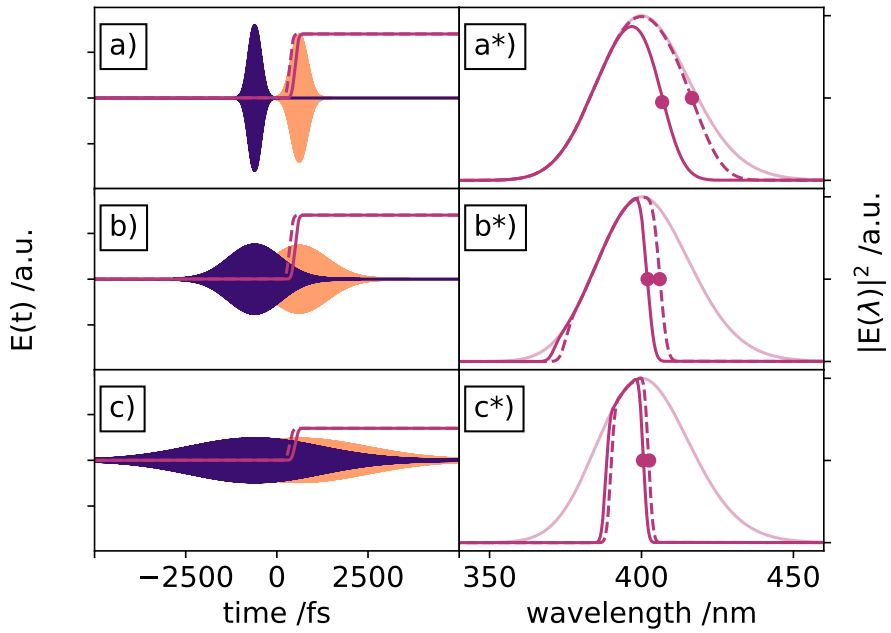

Figure 6.21.: Influence of the temporal optical pulse separation and optical pulse duration. Three different optical pulse configurations are illustrated. Two X-ray pulse arrival-times are simulated. The X-ray pulses are separated by 150 fs, indicated by the purple solid and dashed line. The first example shows the horizontal (blue) and vertical (orange) polarization entirely separated in time, shown in a). The resulting X-ray arrival spectra are shown in a*) for each of the two X-ray pulses. The position of the cut-off edge is indicated by purple dots. In b), the two PC are further chirped, partially overlapping. The two X-ray pulses arrive in a region where both PC are overlapped. The corresponding arrival-time spectra are shown in b*). The two PC are further chirped in c), which are almost completely overlapping and produce the two shown X-ray arrival spectra shown in c*).

Increasing the chirped pulse duration extends the measurement time-window. But increasing the chirped optical pulse duration, the spectral separation between identical temporally separated events is decreasing (Fig. 6.21 a* to c*). The two arrival-time spectra in each subfigure are created by two X-ray pulses, always separated by 150 fs. Using a relatively short optical pulse (500 fs), generates two arrival-time spectra, where the 150 fs pulse separation is translated into a shift of the cut-off in the arrival-time spectra of 12 nm. For a chirped pulse duration of 2000 fs, the 150 fs X-ray arrival-time separation translates to a shift of the arrival-time spectra of 4 nm. For a nearly 4000 fs long optical pulse, the resulting spectra are only shifted by 2 nm. This needs to be kept in mind for a real experiment because using too long optical pulses would severely reduce the time-resolution of the measurement.

Influence of the Spectral Intensity Distribution

So far, the initial optical pulse is assumed to be a bandwidth-limited Gaussian pulse, centered around 400 nm with an FWHM of 50 nm. In real-world experiments, the initial optical pulse is not even close to a bandwidth-limited Gaussian pulse but has a rather arbitrary spectral intensity distribution. Supercontinuum pulses are frequently used as optical probe beams in pump-probe experiments [170] [171]. These spectra can range over the entire optical visible region with rather complicated intensity distributions, depending on the non-linear generation process and initial laser properties. In the following, a typical supercontinuum spectrum, created with a 40 fs, 800 nm optical pulse in a Calcium Fluoride (CaF_2) crystal [170], is used as input into the self-referenced timing-tool simulation. The bandwidth of the initial spectrum would support an optical pulse with <10 fs pulse duration (Fig. 6.22 a*). Before using this optical pulse as input for the simulation, we set the pulse duration to 1.5 ps by propagating the pulse through an additional glass block (Fig. 6.22 a). As before, the chirped pulse is propagated through the first BC, generating a complex optical pulse described through a horizontal (blue) and vertical (orange) PC (Fig. 6.22 b). The X-ray pulse is set to arrive at t=0 fs (indicated by the vertical purple line in b) and introduces a phase shift of $\Delta\phi = 4.5^\circ$ on the transmitted PC.
The two synchronized PCs behind the second BC are shown in c). Due to the large spectral bandwidth, the optical pulse accumulates additional phase when transmitted through the second BC, increasing the overall pulse duration (Fig. 6.22 c). The final arrival-time spectrum is shown in c*) and is multiplied by a factor of 100 to match the original optical pulse spectrum (shaded purple). As shown before, two X-ray imprinted cut-off edges can be observed, indicating the relative arrival-time of the X-ray pulse. The structure of the cut-off edges is influenced by the spectral intensity distribution in this region, thus, the rising and falling edges can have additional features and may not be as clearly visible as shown before when using a Gaussian-shaped intensity distribution. Between the two cut-off edges, the original spectral intensity with all features is conserved.

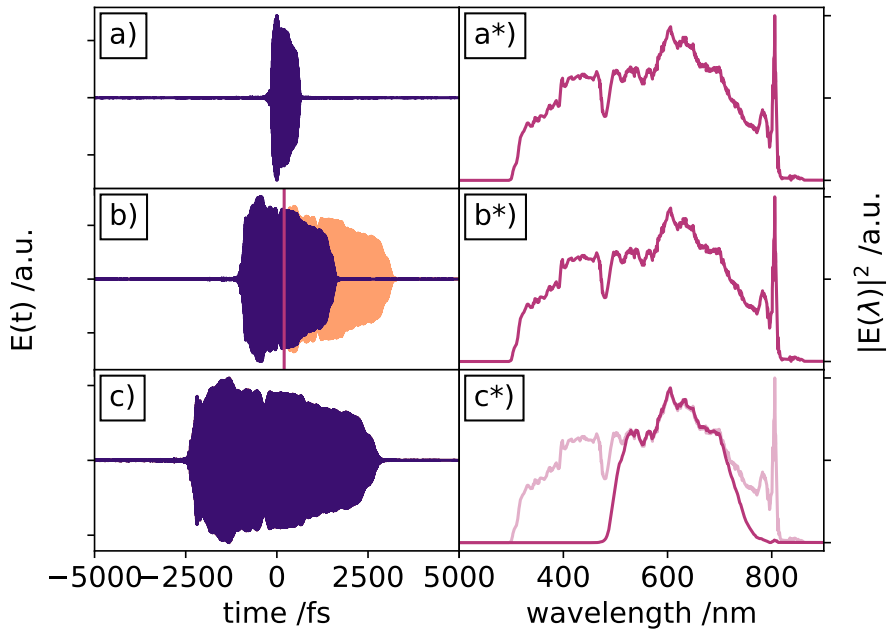

Figure 6.22.: Self-referenced arrival-time signal when using an optical supercontinuum pulse. The electric field and spectrum of the chirped supercontinuum pulse are shown in a) and a*). The horizontal (blue) and vertical (orange) PC of the optical pulse are shown in b), together with an X-ray arrival-time at t=0 fs, indicated by the purple vertical line. The spectrum before the second polarizer is shown in b*). The two synchronized PC after the second BC are illustrated in c). The self-referenced X-ray arrival-time spectrum is shown in c*) as the solid purple line, which is matched to the original supercontinuum spectrum, shown as the shaded purple line.

Arrival-Time Signal Dependence on X-Ray-Introduced Phase Shift

The simulations of the expected X-ray-induced refractive index change in diamond (Section 6.2.2) suggest an X-ray-induced refractive index change in the range of $\Delta n = -10^{-8}$ to -10^{-4} for the polaron based Maxwell Garnett model, and between $\Delta n = -10^{-6}$ and -10^{-2} for the simple Drude model. Assuming X-ray-induced refractive index changes in this range, the resulting possible phase shift, calculated with Eq. 6.16, covers a wide range from $\Delta\phi = 0.0045°$ to over 2000°.

To investigate the influence of the X-ray-induced phase shift on the shape of the self-referenced arrival-time spectra, these arrival-time spectra are simulated for varying X-ray-induced phase shift angles from 1° to 1280° (Fig. 6.23 a). With increasing X-ray-induced phase shift angles, the arrival-time spectra become more and more complicated. The cut-off edge, indicating the X-ray arrival-time, is shifted, despite keeping the X-ray arrival-time in the simulation at a static value.

In an actual arrival-time measurement experiment, the edge position will be used to determine the X-ray pulse arrival-time. Therefore, it is important that the edge position is always the same for identical X-ray arrival-times. For small X-ray-induced phase shifts,

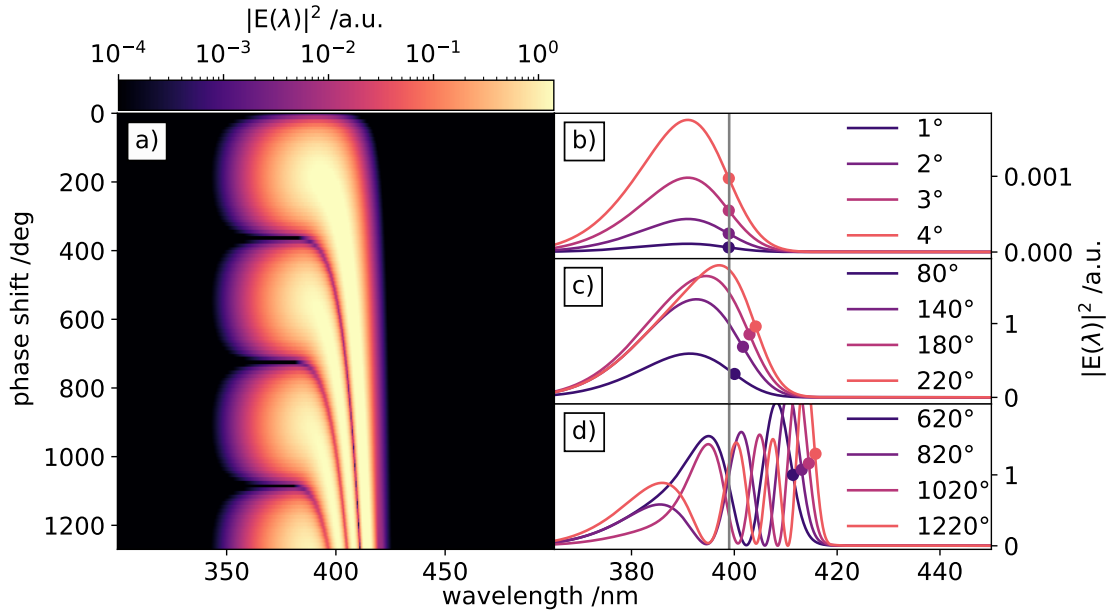

Figure 6.23.: Dependence of resulting X-ray arrival-time spectra on the X-ray-induced phase shift. A 2d-intensity image is illustrated in a), showing arrival-time spectra for identical X-ray arrival-times, but different X-ray-induced phase shifts from 0° to 1280°. The amplitudes of the spectra are color-coded using a logarithmic color map. Individual spectra for specific phase shift values are shown in b), c) and d). The gray vertical line indicates the cut-off edge position for arrival-time spectra with small X-ray-induced phase shifts. For large X-ray-induced phase shifts, the cut-off edge is slowly shifting (c) and the overall spectrum becomes quite complex (d).

the spectral position of the imprinted arrival-time cut-off edge is nearly unchanged (Fig. 6.23 b). The cut-off edge positions of the simulated arrival-time spectra are marked with solid circles and are determined by the minimum position of its first derivative. As a guide to the eye, a gray vertical line indicates these edge positions. This gray line is used in c) and d) as reference.

Increasing phase shifts angles, but still in the same cycle (0°-360°), result in steeper self-referenced arrival-time spectra. In these spectra, the cut-off edge position is not static anymore but moves with increasing phase shift angles (Fig. 6.23 c).

For very large phase shifts, larger than 360°, the spectra become very complex. For each further full cycle shifted ($n \times 360°$), a minimum in the arrival-time spectrum appears (Fig. 6.23 c). This is caused by the fact that the refractive index is not changed instantly but over a time span of up to hundreds of fs, depending on the X-ray photon energy. In this process, each time when the X-ray-induced phase shift is exactly 360°, the involved spectral regions of both PCs are perfectly synchronized behind the second BC and will recreate the original 45° polarization, and thus will not be transmitted through the second polarizer. If the refractive index change is large enough to shift the phase by multiple cycles, an equal amount of minima will be observed in the spectrum.

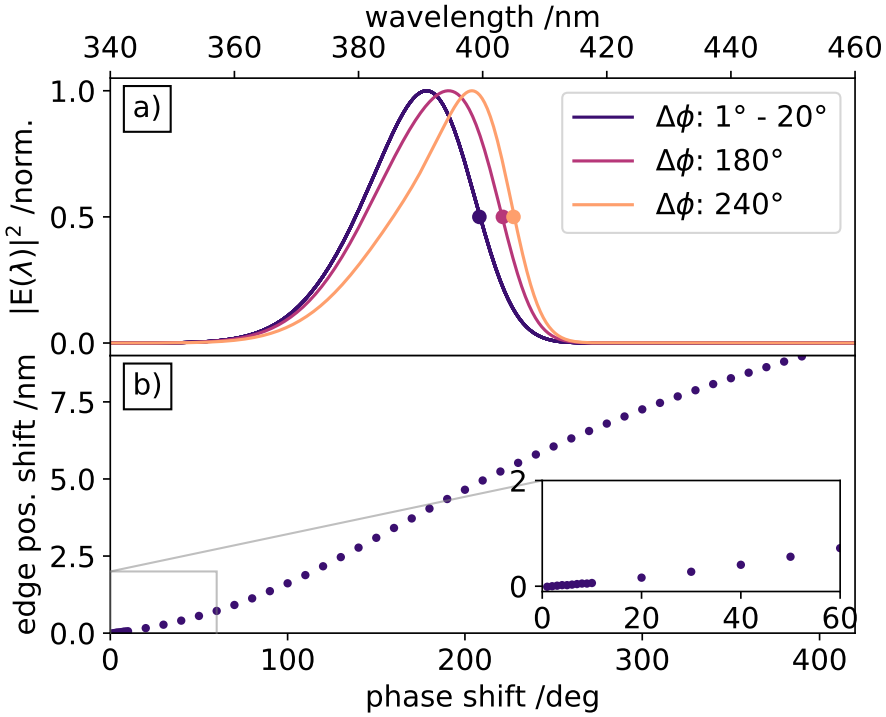

Figure 6.24.: Arrival-time spectra for different X-ray-induced phase shifts with constant X-ray arrival-times. Normalized spectra for X-ray-induced phase shifts of 1-60° (blue), 180° (purple) and 240° (orange) are shown in a). The cut-off edge position is indicated by the blue, purple and orange dots and are the turning points (1st derivative minima) of each spectrum. The cut-off edge positions drift for X-ray-induced phase shifts from 0° to 420° is shown in b).

Further analysis shows that up to an X-ray-induced phase shift of $\Delta\phi = 20°$, the cut-off edge position is nearly static (Fig. 6.24 a). All 20 simulated and normalized spectra for X-ray-induced phase shifts between 1° and 20° (blue) are nearly identical and show the same cut-off edge position. Higher X-ray-induced phase shifts such as $\Delta\phi = 180°$ (purple) and 240° (orange) result in a clearly shifted edge position. The spectral position shift of all cut-off edge positions as a function of the X-ray-induced phase shift is shown in Fig. 6.24 b). With increasing phase shifts, the edge position shifts increasingly to higher wavelengths. In an optimal operating condition, the self-referenced timing-tool would be operated in a regime, where the X-ray-induced phase shift is in the range $\Delta\phi \leq 20°$. In this region, the edge position shift is well below 0.5 nm and is almost negligible and experimentally not relevant with commonly used spectrometers.

7. X-ray Arrival-Time Measurement in Diamond at European XFEL

In this chapter, the results of the measurement campaigns at EuXFEL are presented. We successfully implemented the self-referenced X-ray arrival-time measurement setup to record X-ray arrival-times at the FXE beamline. The results demonstrate one of the first MHz-repetition rate arrival-time measurements at EuXFEL and are the only one using diamond as the interaction sample, making it a prime candidate for a nearly non-invasive durable setup at the beamline, even for extreme X-ray conditions.
First, a description of the used setup at the FXE beamline is provided. In the next step, the analysis process to determine the X-ray pulse arrival-times is described. Recorded X-ray arrival-times with the two different facility synchronization schemes (RFS and OS) are evaluated. The value of the X-ray-induced refractive index change in the diamond sample is investigated by comparing the experimental data with simulations and comparing the results to the predicted refractive index change by the Drude and Maxwell Garnett models. Finally, the required X-ray fluence to operate the self-referenced detection scheme is analyzed and an estimate for a lower boundary is given.

7.1. Experimental Setup and Conditions

The experiment was carried out at the FXE instrument at the end of the SASE1 photon tunnel [172, 173] of European XFEL [174, 23] with a fixed X-ray photon energy of 9.3 keV and a mean X-ray pulse energy of 300 µJ. We used a train filling pattern at a 1.128 MHz repetition rate and 120 pulses stored in each pulse train. The divergent X-ray pulses were collimated and (slightly) pre-focussed in the photon tunnel with a set of CRLs, 230 m away from the X-ray photon source in the undulators to match the horizontal acceptance of the deflection mirror M3. This mirror is placed 370 m behind the source point and diverts the X-ray pulses into the FXE instrument [174]. In the X-ray optics branch of the FXE instrument, an additional set of CRLs, 5 m upstream from the sample, was used to focus the X-ray pulses. We used lens 9 of the lens stack, yielding a nominal focus size at the sample position of around 170 µm FWHM [172, 173]. Due to an asymmetric prefocus in the SASE1 tunnel, the actual X-ray spot size at the sample positions was (150±5) $\times$ (30±5) µm^2.
The in-house developed EuXFEL optical pump-probe laser system was utilized, which is synchronized to the facility's main oscillator, and matches any chosen X-ray pulse pat-

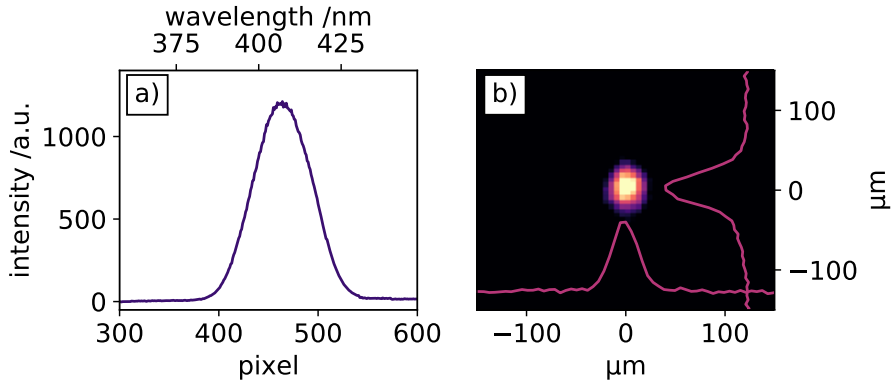

Figure 7.1.: The spectrum of the used optical pulses for the arrival-time measurements is shown in a). The beam profile in the focal plane, where the diamond sample is placed, is shown in b). The measured beam diameter as an FWHM of 40 µm.

tern of the facility [42, 175]. The optical laser can be synchronized using either the RFS or OS scheme (see Section 2.3). It delivers ultrashort 15 fs RMS pulses at its 800 nm central wavelength.

The entire optical setup of the self-referenced timing-tool was mounted on the multi-axis sample-stack at the nominal sample position, enabling the movement of the entire setup in all required degrees of freedom [172, 173]. The principal optical setup is described in Chapter 5. Instead of using the original 800 nm broadband laser pulses for the detection of the self-referenced X-ray arrival-times, we used a BBO-crystal to generate the second harmonic of the fundamental wavelength. The resulting optical pulse showed a spectrum centered around 405 nm with an FWHM bandwidth of around 20 nm (Fig. 7.1 a). The optical pulses were focussed to 40 µm FWHM diameter spot at the sample position (Fig. 7.1 b) with a pulse energy < 5 µJ.

Originally, we planned on using the fundamental broadband 800 nm optical pulses for the self-referenced X-ray arrival-time measurement. For still unknown reasons, we were only able to detect a very weak X-ray-induced self-referenced arrival-time signal with the 800 nm optical pulses. Due to the limited access of beamtime, we decided to frequency-double the fundamental laser pulses. With 400 nm broadband pulses, we immediately detected a strong self-referenced X-ray arrival-time signal. While the X-ray-induced refractive index change is almost identical for both probing wavelength regions, the X-ray-induced phase-shift is twice as large at 400 nm compared to 800 nm (Eq. 6.16), which increases the signal amplitude.

We used Nanoparticle Linear Film Polarizers (Thorlabs, LPVIS) with a contrast ratio of 1:100 000 at 405 nm and two a-cut αBBO crystals (custom-made by Eksma Optics) with a thickness of 5 mm for the CPI. The optical pulses were guided to a spectrograph (Andor Shamrock 193i) with an equipped 600 grooves/mm grating at a blaze wavelength of 500 nm. Various neutral density filters (Thorlabs ND Filter with AR coating) were placed in front of the entrance slit to protect the detector from optical saturation. When

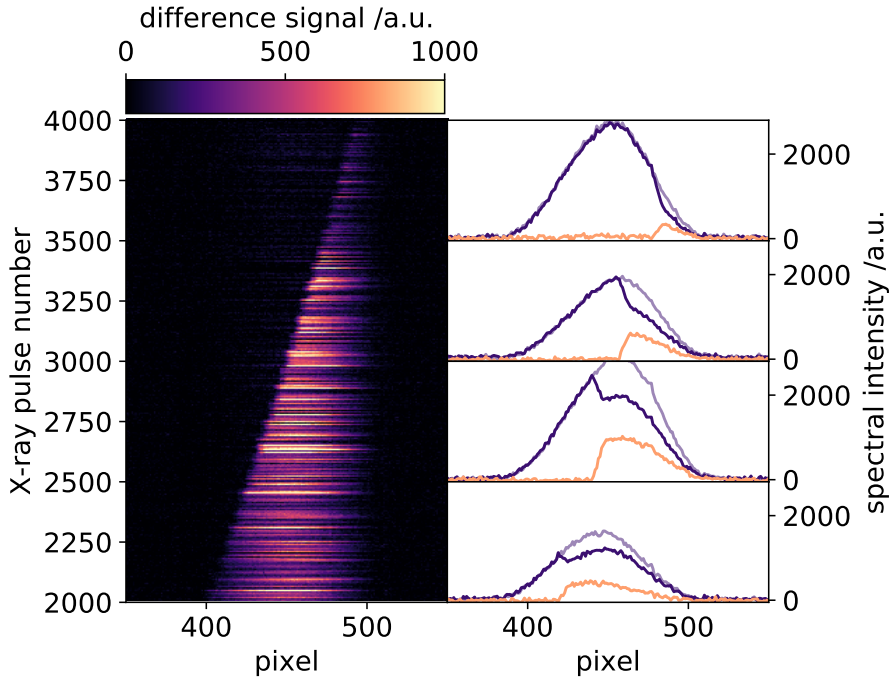

Figure 7.2.: Spectral encoding in YAG. The difference signals of a spectral encoding test measurement are shown in the 2-dimensional plot. The relative X-ray arrival-time is systematically scanned from early (top) to later (bottom) arrival-times. Four example measurements are extracted on the right-hand side. The actual spectral encoding spectrum is shown in blue, the reference spectrum shown in light blue, and the difference signal with the imprinted X-ray arrival-time is shown in orange.

not stated otherwise, the standard configuration during this experiment used a 1.5 OD filter (Thorlabs NE15A-A) in front of the spectrometer. For the detection of the individual spectra, we used a Gotthard detector [176] in the focal plane of the spectrograph, operated at 564 kHz, thus, recording every second X-ray pulse. In the near future, newer line detectors such as the Gotthard-II detector [177] and the KALYPSO detector [178] can capture every single pulse of a full pulse train at a 4.5 MHz repetition rate. In the meantime, a second Gotthard detector can be attached to a second output of the Andor spectrograph. These two Gotthard detectors can be operated to record the arrival-time spectra in an alternating fashion, doubling the data acquisition rate. Fused-silica glass blocks with a thickness of 0.4 cm or 2.5 cm were added to the 400 nm optical beam path to chirp the optical pulses to increase the temporal measurement window.

First, we performed a test measurement with the second polarizer removed from the setup. This transforms the setup into a (complicated) spectral encoding setup. We used a 100 µm thick YAG crystal to find the initial temporal and spatial overlap. YAG is known to produce a very strong X-ray-induced spectral encoding signal [115, 124], but at 10 keV it absorbs nearly 80% of the X-ray pulse intensity. The relative delay between the optical and X-ray pulses was systematically scanned to find the temporal overlap (Fig. 7.2).

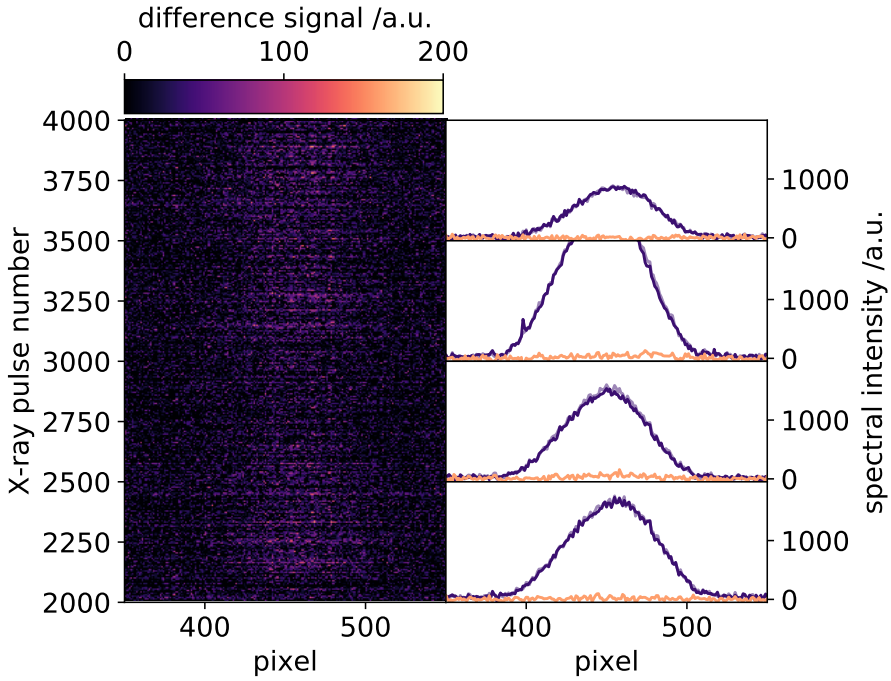

Figure 7.3.: Difference spectra of a spectral encoding time-delay scan in diamond. Difference signals of the entire measurements are shown in the left 2-dimensional image. On the four sub-panels on the right, four exemplary measurements are shown with the spectral encoding spectrum (blue), reference spectrum (light blue) and difference signal in orange. No X-ray-induced spectral encoding signal can be observed.

The 2-dimensional image on the left-hand side shows the resulting difference signals of an optical laser pulse transmitted through the sample without an impinging X-ray pulse and an optical pulse transmitted through the sample with an impinging X-ray pulse for changing relative delays. In the four sub-panels on the right-hand side, four individual measurements with different relative X-ray delays are shown. Since a reference of the optical spectrum is needed to evaluate the spectral encoding signal, we used the 100th optical pulse in each pulse train to measure the X-ray-induced spectral encoding signal (blue) and used the directly following 101st optical pulse (where the X-ray pulse train is unfilled) as the reference pulse (light blue). By subtracting the reference spectrum from the actual spectral encoding spectrum, an X-ray-induced transmission change can be observed (orange). A clear spectral encoding signal is visible in all of those measurements with a spectrally moving absorption feature due to the changed relative X-ray arrival-time.

Immediately after the spectral encoding study with the YAG sample, we exchanged the YAG sample and inserted a 50 µm thick CVD diamond attempting to directly reproduce the previous spectral encoding measurement. As expected, no spectral encoding signal is observed in diamond (Fig. 7.3), as it transmits almost 100% of the incoming X-ray pulse and the resulting transient absorption change of the transmitted pulse is very

weak. The X-ray-induced refractive index change is far too small to change the transmission of the optical pulses to be able to obtain a meaningful difference signal, which was also reported by Bionta et al. [115]. This justifies the use of the self-referenced detection scheme, enabling the use of the identical diamond sample and same X-ray beam conditions to measure the relative X-ray arrival-times.

7.2. Self-Referenced Arrival-Time Analysis

A robust algorithm is required to detect the cut-off edge position in each single arrival-time spectrum. Two measured raw self-referenced arrival-time traces are shown in Fig. 7.4. These two traces were selected because they show a rising (orange) and falling (blue) cut-off edge where the X-ray arrival-time was imprinted. Since each arrival-time spectrum posses only one cut-off edge, it is clear that the two PCs, generated by the first BC, are entirely separated in time (see Section 6.2.3). The blue spectrum shows a measurement where the X-ray pulse is overlapped with the leading PC, while the orange spectrum shows a measurement where the X-ray pulse is overlapped with the trailing PC.

To fully exploit the ultrashort optical and X-ray pulses and to enhance the time-resolution of pump-probe experiments, the X-ray-induced arrival-time signal needs to be reliably detected with the highest possible precision. The relative X-ray arrival-time can be defined by the cut-off edge position of the X-ray-induced arrival-time signal. We used our own algorithm to analyze the X-ray-induced arrival-time edge position.

The edge position is analyzed for each arrival-time spectrum by fitting a Gaussian error function

$$f(x) = A\left(1 + \mathrm{erf}\left(\frac{x - \mu}{\sigma}\right)\right) + C\ , \tag{7.1}$$

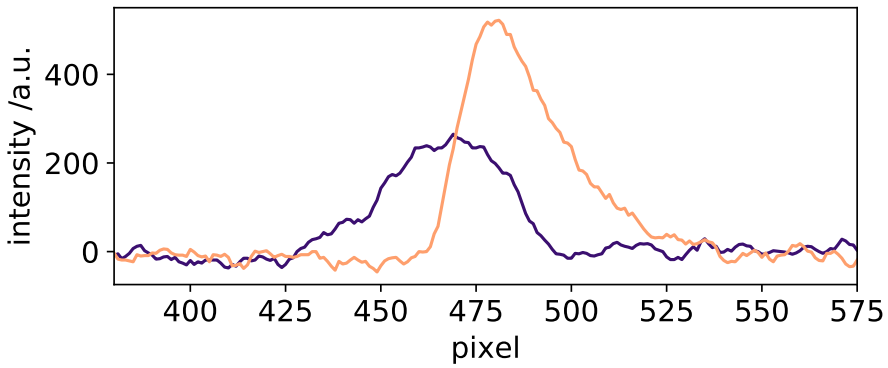

Figure 7.4.: Raw arrival-time data at EuXFEL. Two exemplary traces are shown, one with a leading edge (blue) and a trailing (orange) X-ray-induced arrival-time edge.

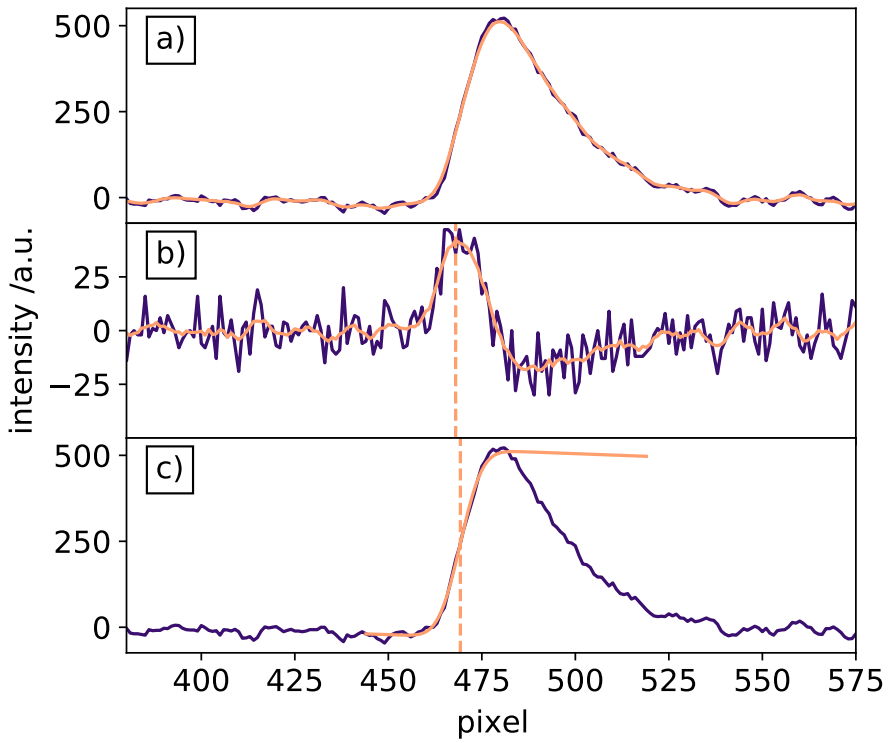

Figure 7.5.: Two-step algorithm to determine the X-ray-induced arrival-time position. A raw and smoothed spectrum is shown in a). The rough edge position is determined by using the first derivative of the smoothed experimental data b). These edge positions are used as starting conditions to fit the fitting function $f(x)$ to the raw experimental data c).

to the arrival-time signal. The fit function is a superposition of a Gaussian error function and a linear function C. The Gaussian error function describes the step-like self-referenced X-ray arrival signal (edge in the spectrum) and the linear function C is used to account for baseline slope variations. The Gaussian error function amplitude is defined by A, the center of the error function by μ and the width by σ. The edges are fitted using the fitting function $f(x)$ with Python and the fitting library LMFIT [179]. The fitting process itself is based on a Levenberg-Marquardt algorithm [180, 181]. Due to the high repetition rate at EuXFEL, the amount of spectra which need to be analyzed is quite large. The DESY Maxwell HPC cluster was used to parallel process the measured arrival-time spectra, decreasing the overall computation time tremendously.

For the analysis, we use a two-step algorithm (Fig. 7.5). In the first step, appropriate starting values for the fitting function $f(x)$ are determined. The most critical starting value is the edge position μ. The initial starting value is determined by identifying the position of the extrema of the first derivative. Therefore, the raw spectrum (Fig. 7.5 a), blue) is smoothed (Fig. 7.5 a) orange) with a Savitzky-Golay filter [182]. The first derivative of the raw experimental data (blue) is often too noisy. Instead, we use the first derivative of the smoothed data (orange) to estimate the edge position. The estimated edge position is indicated by the orange vertical dashed line. This estimated edge position is then used

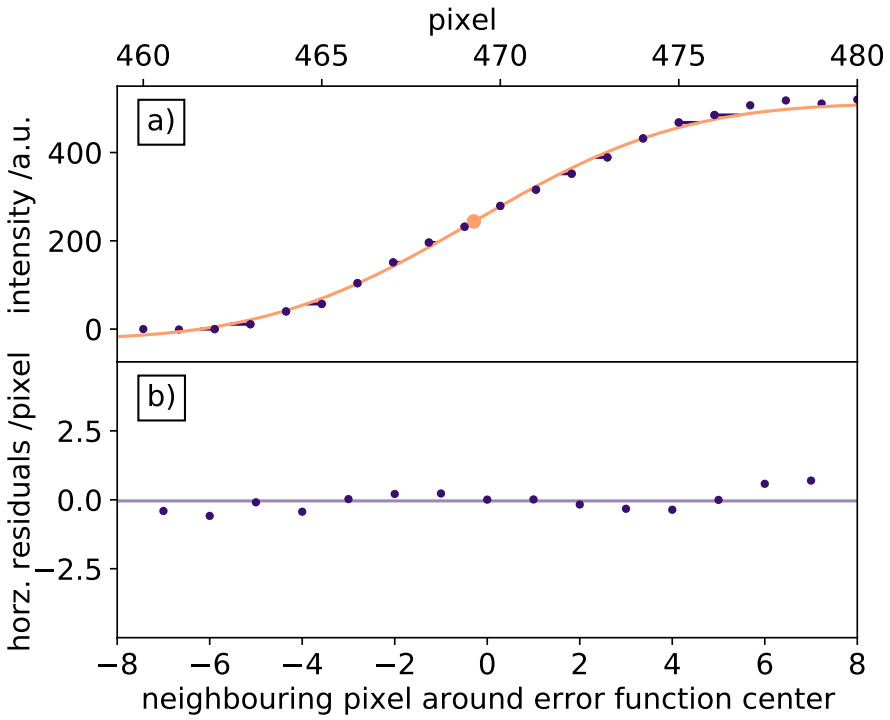

Figure 7.6.: Fit precision of the error function parameter μ. The horizontal distance of the experimental (blue) data points to the fit function (orange) is evaluated in a). The corresponding measured relative distances are shown in a'). The horizontal line indicates the mean value.

as a starting value for the Levenberg-Marquardt algorithm to fit the fitting function $f(x)$ to the raw data around the X-ray-induced arrival-time edge (Fig. 7.5 c). The new edge position of the error function is again indicated with a vertical dashed line.
The advantage of using the more complicated fitting function is that it can be applied directly to the raw data instead of using the smoothed data. The smoothed data is only used to determine the rough edge position, which is used by the Levenberg-Marquardt algorithm as starting condition to find a converging fitting result.

The most important variable is the center position μ of the fitting function. This determines the center position of the error function, which is used to determine the relative arrival-time of the X-ray pulse. The accuracy of this fitting value to the experimental data is an important measure of the achievable precision to determine the center μ of the error function. As a measure of the fit precision, we use the horizontal distance between the experimental data and the fitting function (Fig.7.6 a). The distance between the experimental (blue) data points around the estimated error function center and the fit function (orange) is measured. The center μ of the error function is indicated by the orange dot. The horizontal distances between the experimental data points and the fitting function are called fit residuals. These relative fit residuals are shown in b), together with their mean value indicated by the horizontal line. These values are stored together with the values for the fitting function parameters for each single arrival-time spectrum and are

used to determine the overall achievable time resolution (see next Section).
A major problem throughout the measurement campaign was the stability of the self-referenced arrival-time signal. In the analysis, we noticed huge variations of arrival-time spectra amplitudes. Many spectra had a Signal-to-Noise-Ratio (SNR) of below 2, making it impossible for the fitting algorithm to find converging fitting parameters. Therefore, these measurements were discarded in the analysis. This problem applied to about 40% of the total data. For these measurements, no X-ray arrival-time could be extracted due to very weak amplitudes of the arrival-time spectra. Thus, it is important to understand the source of these amplitude fluctuations.
Three exemplary pulse trains are shown in Fig. 7.7. In an optimal pulse train every single self-referenced arrival-time spectrum can clearly be measured with a sufficient SNR (Fig. 7.7 a). Unfortunately, in most of the pulse trains, there are arrival-time spectra with an amplitude too weak to be analyzed (Fig. 7.7 b). A minor portion of the pulse trains contained no analyzable arrival-time spectra at all (Fig. 7.7 c).
Assuming that the optical and X-ray beam size in the diamond sample remains unchanged, the main experimental contribution influencing the amplitude of the arrival-time signal is the varying pulse energy of the X-ray (and optical) pulses and the spatial overlap between both pulses. The X-ray pulse energy and the spatial position at the entrance of the FXE instrument is measured by its Intensity and Position Monitor [172], an arrangement of four diodes placed around the X-ray beam (left, right, top, bottom). These are used to pick up backscattered X-ray photons to determine the X-ray beam position by analyzing the measured intensity difference of the vertical and horizontal diode pairs. The X-ray pulse intensity is then given by the total signal of all four diodes. The calibration of the beam position monitor is accomplished by using one of the nearby Beam Imaging Units [172].
A compilation of X-ray pulse trains with their measured X-ray arrival-time spectra (left column), pulse energies (middle column) and spatial position (right column) is shown in Fig. 7.7. The orange arrival-time spectra in a)-c) correspond to the first five X-ray pulses within the pulse train. A general trend is that the X-ray pulse energies within a pulse train are increasing with subsequent X-ray pulses. The five weakest (orange) spectra correspond to the first five X-ray pulses in the pulse train with the lowest individual pulse energies (Fig. 7.7 a' and b'). The corresponding spatial positions of these X-ray pulses are illustrated in a") and b"). Especially in b"), the first five low-energy X-ray pulses are horizontally far away from the mean X-ray position. Thus, two main contributions deteriorate the arrival-time signal: i) low X-ray pulse energies yielding low self-referenced arrival-time spectra amplitudes, and ii) poor spatial overlap of the X-ray and optical pulses, also decreasing the spectral amplitude of the arrival-time signal.
These observations can not explain the case presented in Fig. 7.7 c)-c"), where no arrival-time signal is observed for any X-ray pulse in the entire pulse train. According to the X-ray pulse energies shown in c') and the spatial position of the X-ray pulses illustrated

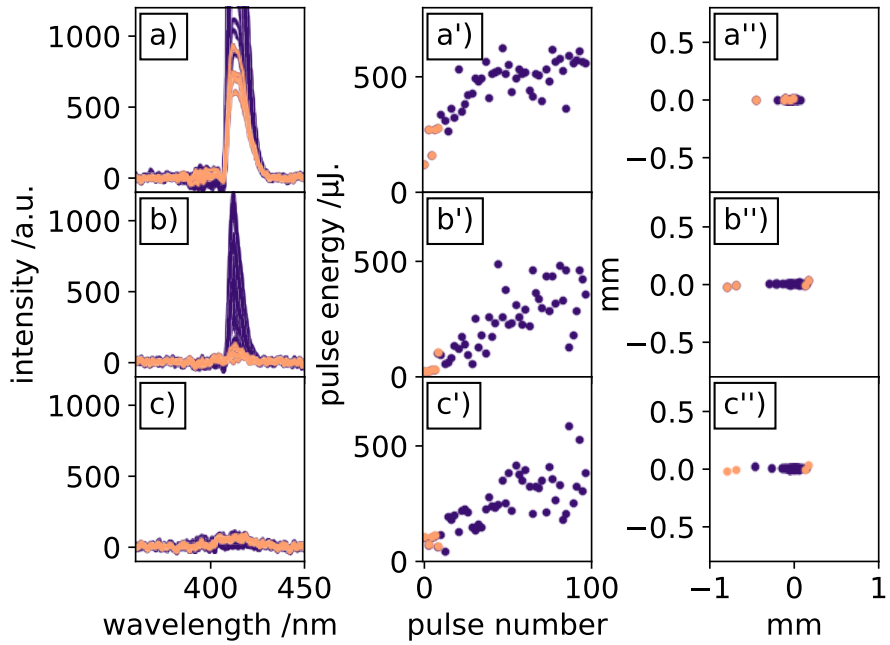

Figure 7.7.: Investigation of fluctuating signal strength of the arrival-time spectra. Few X-ray pulse trains have pulses where all X-ray arrival-time spectra show strong signals (a), most pulse trains have a mixture of very weak and strong arrival-time signals (b), while some show no arrival-time spectra at all (c). The X-ray pulse energies of the relevant pulses in the pulse train are shown in a')-c'), while the spatial jitter of the X-ray pulses at the IPM position is shown in a'')-c''). The first 5 X-ray pulses of the pulse train and their arrival-time spectra are shown in orange.

in c''), one would at least expect some arrival-time spectra with a detectable signal amplitude.

An additional influence on the arrival-time spectrum amplitude is the optical laser. By decreasing the pulse energy of the optical pulse, the overall amplitude of the obtained self-referenced arrival-time signal decreases due to a smaller X-ray-induced refractive index change. Unfortunately, we did not monitor the pulse energy of the optical laser during the measurements to correlate the optical laser pulse energies to the data shown in Fig. 7.7. To get an idea of the pulse energy stability of the optical pulses, a test run from before the actual experiment is shown in Fig. 7.8. In this graph, the maximal amplitude of 21 000 individual optical pulses is analyzed. The mean amplitude is at 242 a.u., while the minimum and maximum are at 105 a.u. and 412 a.u., respectively. These strong fluctuations in the spectral amplitude are related to smaller pulse energy fluctuations of the 800 nm fundamental laser pulse and are enhanced by the non-linear second harmonic generation process.

The few events, where not a single X-ray-induced arrival-time spectrum could be measured within a pulse train (Fig. 7.7 c), could be attributed to a combination of bad spatial overlap between the optical and X-ray pulses with very low pulse energy.

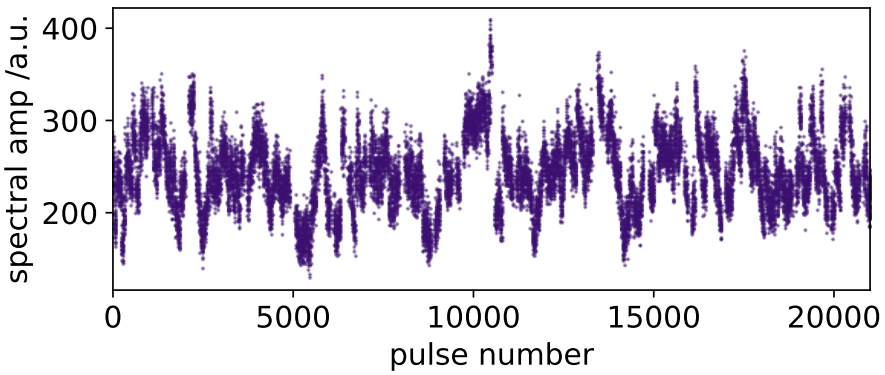

Figure 7.8.: Intensity fluctuation of 21 000 consecutive optical laser pulses.

The last possibility which would explain a zero-signal in the self-referenced arrival-time spectrum is a timing jitter event exceeding the temporal measurement window, defined by the temporal duration of the optical pulses. This would require a sudden jump in the synchronization between consecutive pulse trains on the order of 2-3 ps or more, which is very unlikely.

The X-ray beam pointing instabilities in our experiments have also been noticed by EuXFEL and its users, and a beam stability task-force was created to tackle these issues [183, 184]. A possible source of the spatial fluctuations is most likely identified to be vibrations of the M3 mirror, which is inserted ~570 m upstream of the experiment [174]. The spatial fluctuations of over 160 000 individual X-ray pulses show that the spatial fluctuations mainly occur on the horizontal axis (Fig. 7.9 a). The FWHM beamsize at the IPM position is smaller than 1 mm, depending on the X-ray energy [172, 173]. Thus, the horizontal jitter is larger than a whole beam diameter, while the vertical jitter is much smaller than a single beam diameter. In Fig. 7.9 b), the average distance to the center of gravity (COG) of the first to the last X-ray pulse within the pulse train is analyzed. Especially the first pulses of a pulse train are subject to large spatial fluctuations in the order close to a full beam diameter. After the first ten pulses, the spatial positions of the X-ray pulses gradually approach the COG. The average pulse energies of the X-ray pulses within the pulse trains show that the first X-ray pulses have an average pulse energy of 68 μJ, which increases throughout the pulse train until it reaches an average pulse energy of 443 μJ for the last X-ray pulses within the pulse train (Fig. 7.9 c). To exclude a correlation between low X-ray pulse energies and the spatial position far away from the COG, the average pulse energies of all X-ray pulses within a distance of 0.1 mm to the COG (orange) and a distance larger than 0.1 mm to the COG (purple) are also analyzed, showing no significant differences. These two observations from Fig. 7.9 b) and c) confirm the observed pattern in Fig. 7.7, where we assumed that the first X-ray pulses with an increased spatial fluctuation also have lower pulse energies. The reasons for these properties are not fully clear yet. It is also not understood if these patterns always occur generally or if these patterns are connected to a day-to-day variation in the configuration of the accelerator

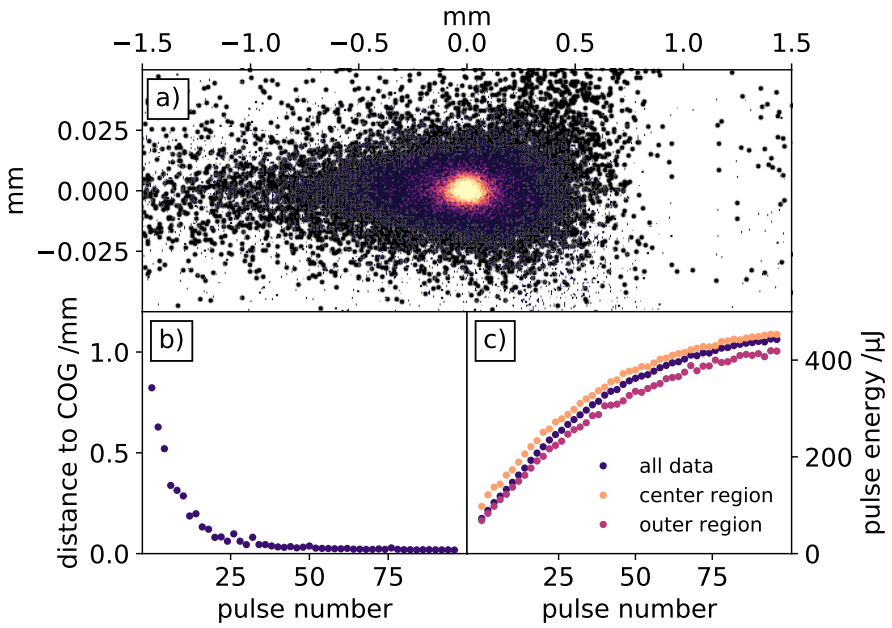

Figure 7.9.: X-ray pulse statistics of over 160 000 X-ray pulses. The spatial jitter, measured at the IPM position at the FXE beamline, is shown in a). The average distance of each pulse number within a pulse train to the COG is shown in b). The average pulse energy of the pulses within a pulse train is illustrated in c) for all pulses (blue), only the X-ray pulses with a spatial jitter less than 0.1 mm from the COG (orange) and with a spatial jitter more than 0.1 mm from the COG.

and undulator settings. These fluctuations are also under investigation by the facility.

The fluctuating pulse energies of the optical and X-ray pulses are commonly normalized with an intensity I_0 measurement. Pointing instabilities of the X-ray beam still need to be addressed and are harder to deal with. The used X-ray focus is often several times smaller than the optical focus size, such that X-ray pointing fluctuations of 2-3 times the beam diameter still yield a good overlap. For the operation of a timing-tool, or any X-ray-pump/optical-probe experiment, an X-ray beam pointing instability is an issue to be resolved.

7.3. Time Axis Calibration

To attribute a relative arrival-time of the X-ray pulses to the measured self-referenced arrival-time spectra, a calibration of the time axis is needed. Due to the linear chirp of the optical pulses, the wavelengths of the measured spectra are temporally aligned, longer wavelengths arriving earlier in time, followed by shorter wavelengths. With a pixel-to-time calibration, the pixel position of the cut-off edge of the X-ray-induced arrival-time spectra can be converted to a relative arrival-time.

To obtain the pixel-to-time calibration, we used the more precise OS of the EuXFEL facility. We performed a systematic delay scan where we changed the motorized delay stage,

used for the OS link between the EuXFEL facility and the optical pump-probe laser, in 200 fs steps, effectively shifting the optical and X-ray pulses by 200 fs with respect to each other.

A time calibration was recorded for each of the two used fused-silica glass blocks, which

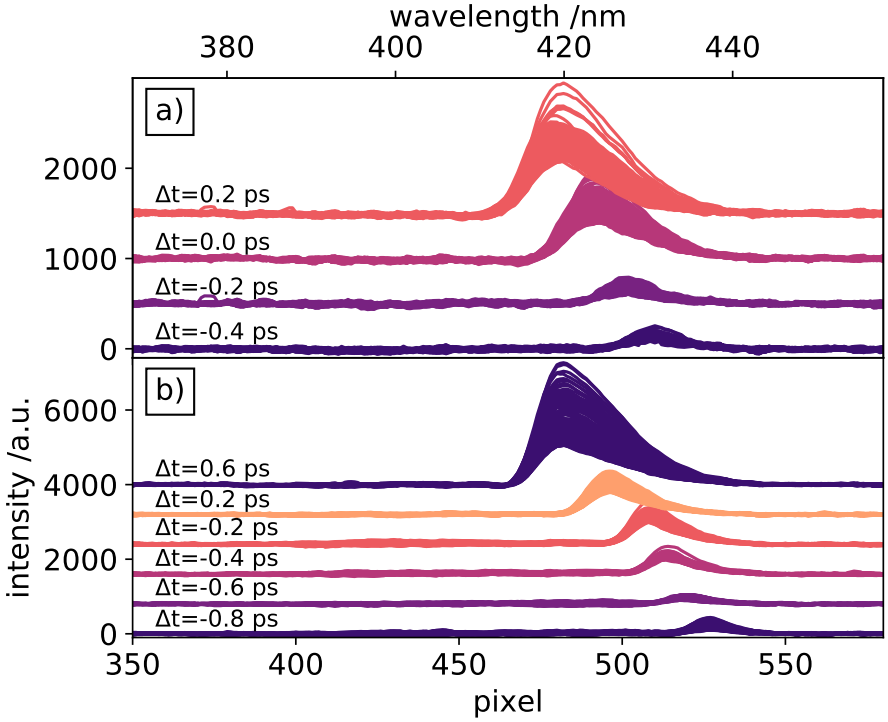

Figure 7.10.: Calibration of the time axis for the 0.4 cm (a) and 2.5 cm (b) thick fused-silica glass blocks, chirping the optical pulses. Each color shows thousands of individual arrival-time spectra at a fixed delay between the optical and X-ray pulses. The indicated time-delay between each collection was changed by an optical delay line.

added different amounts of chirp to the optical pulses. The two time calibration measurements are shown in Fig. 7.10 a) and b) for the 0.4 cm and 2.5 cm thick blocks, respectively. Thousands of individual self-referenced X-ray arrival-time spectra were recorded for each delay. Each color of the plots represents a different relative delay setting between the optical and X-ray pulses, such that the relative X-ray arrival-time changed and the cut-off edge, indicating the X-ray arrival-time, shifted its position. As described in Chapter 7.2, each arrival-time spectrum is individually analyzed, and the exact position of the cut-off edge is determined. For each delay setting, the average pixel positions of the first 50 X-ray pulse trains are analyzed (Fig. 7.11 a and b). The dashed line indicates the mean pixel position of all analyzed X-ray arrival-times at each delay setting. Using the mean pixel positions for each delay, a linear pixel-to-time calibration can be found for both calibration measurements (Fig. 7.11 c and d). For the 0.5 cm thick fused-silica block a calibration of (19±8) fs/pixel is found, while the time calibration for the 2.5 cm thick fused-silica is (28±5) fs/pixel. Using the very narrow bandwidth of only 20 nm FWHM

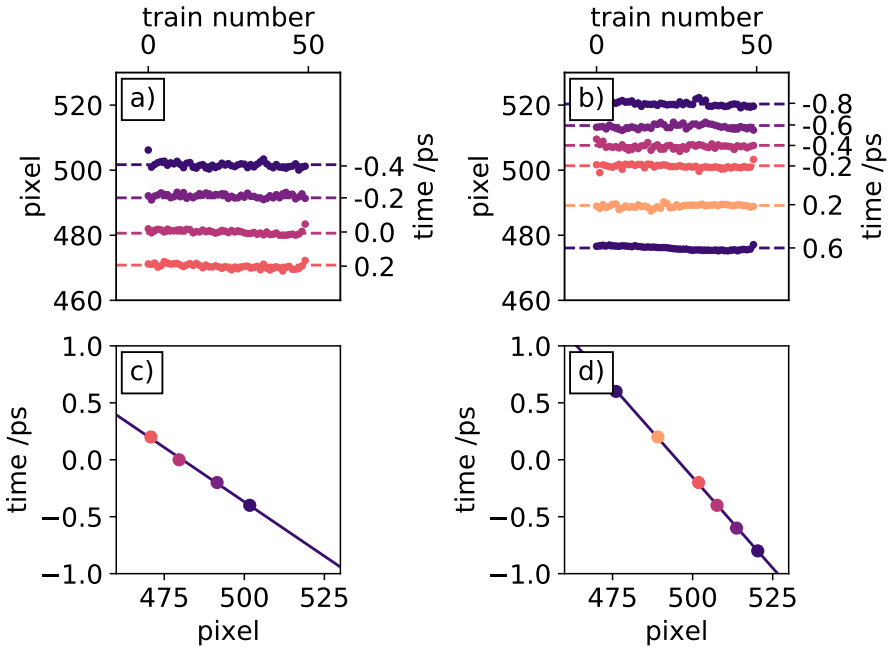

Figure 7.11.: Time axis calibration for the two fused-silica glass blocks. The average cut-off edge pixel positions of 50 X-ray pulse trains for each delay setting is shown in a) and b) for the 0.5 cm and 2.5 cm fused-silica glass blocks, respectively. The mean values for each delay are shown as vertical dashed lines. The linear calibration, using the mean values of each delay are shown in c) and d).

of the optical pulse justifies the linear approximation of the chirp, while for broad bandwidth optical pulses, a quadratic approximation would be more suitable.

7.4. Results of the Arrival-Time Measurements

During the actual X-ray arrival-time measurements we used the facility RFS as well as the OS scheme. For most measurements, the 0.4 cm fused-silica glass block was used to chirp the optical pulse. A comparison of X-ray arrival-times measured for 60 000 consecutive X-ray pulses using the RFS (blue) scheme and for 50 000 consecutive X-ray pulses using the OS (orange) scheme demonstrates the smaller timing jitter for the OS (Fig. 7.12). Already on the shorter timescales of roughly two minutes, larger fluctuations and timing drifts are observed for the RFS. These timing drifts are expected to become larger on longer timescales above 15 min [124].

A comparative analysis of the measured X-ray arrival-times with the RFS and OS shows the extracted X-ray arrival-times for six consecutive pulse trains (Fig. 7.13 a and b). Comparing both synchronization schemes reveals that the major source of the increased X-ray arrival-time spread of the RFS is the 10 Hz inter-train timing jitter, while within the pulse trains, the spread of the relative arrival-times is comparable. Evaluating the

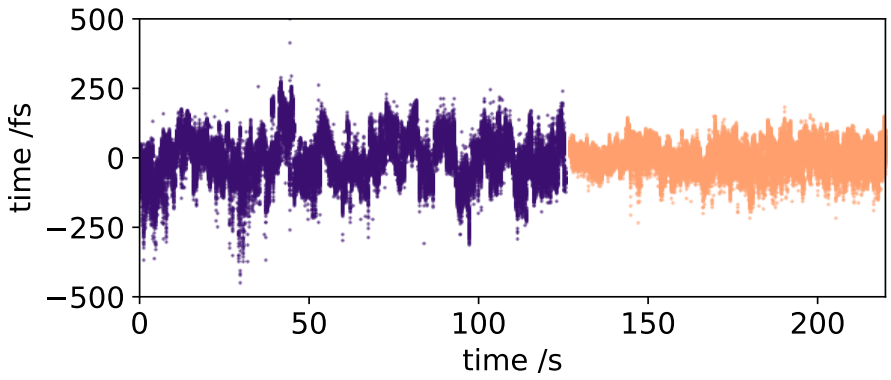

Figure 7.12.: X-ray arrival-times of 60 000 consecutive X-ray pulses using the RFS (blue) and 50 000 X-ray pulses using the OS scheme (orange).

relative arrival-times for all recorded X-ray pulses yields a timing jitter distributions of (161±19) fs FWHM for the RFS and (90±19) fs FWHM for the OS (Fig. 7.13 c and d). These values are in good agreement to the reported values by Kirkwood et al. [123] and Sato et al. [124]. Both of these reported measurements were performed at the neighbouring SPB beamline using the identical laser system but using a different sample and a conventional spectral encoding scheme setup.

An additional important feature is observed investigating the intra-train X-ray arrival-time pattern (Fig. 7.13 a and b). As a guide to the eye, a linear function is fitted to the X-ray arrival-times in each single pulse train. Distinct X-ray arrival-time drift pattern within the pulse trains are found for the two synchronization schemes. Using the RFS, the relative arrival-times of consecutive X-ray pulses within a pulse train are generally drifting to earlier arrival-times, while with the OS the X-ray arrival-time drift is reversed, and the intra-train arrival-times of the X-ray pulses are drifting to later arrival-times with increasing X-ray pulse numbers.

Further analysis of the intra-train drift pattern show that the average arrival-time drift using the RFS is negative and seems to be linear, while the average intra-train arrival-time drift with the OS is positive and approaches a maximum before reversing (Fig. 7.14). The reversing effect observed with the OS could be caused by the active stabilization loop of the OS. For both synchronization schemes, the relative timing jitter of the first few X-ray pulses is larger than for the following X-ray pulses, where the standard deviation (shaded areas) of the measured relative arrival-times are getting smaller. This correlates with the discussed and observed X-ray pulse energy and pointing fluctuations (Fig. 7.9), which also affects the first X-ray pulses of the pulse trains.

The intra-train arrival-time distributions for the RFS and OS are almost identical with 21 fs FWHM (RFS) and 23 fs FWHM (OS) (Fig. 7.14 left and right). This suggests that the OS is mainly stabilizing the 10 Hz inter-train timing jitter and actively prevents slower timing drifts.

An interesting investigation for the future, also regarding the EuXFEL stability task force,

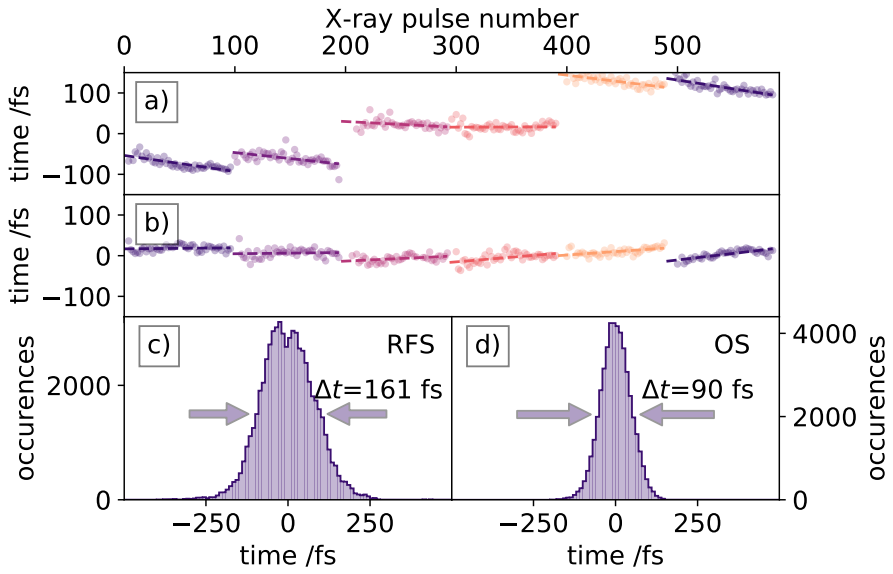

Figure 7.13.: Analysis of measured X-ray arrival-times. Relative arrival-times for six consecutive pulse trains are shown in a) using the facility RFS and in b) with using the OS scheme. In each of the pulse trains, a linear function is fitted to indicate the drift patterns of the relative arrival-times within a pulse train. The X-ray timing jitter distribution of the extracted X-ray arrival-times are shown in c) for the RFS and d) for the OS, yielding an FWHM timing jitter of 161 fs and 90 fs, respectively.

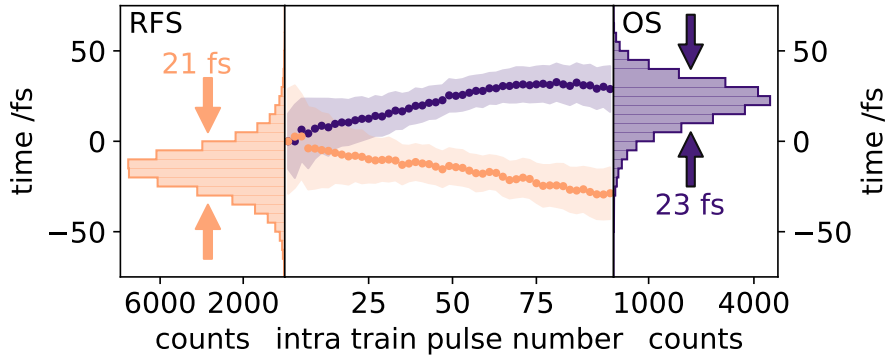

Figure 7.14.: Average intra-train X-ray arrival-times are shown in the main panel (center) using RFS (orange) and OS (blue). The average arrival-time of the first X-ray pulses of the pulse trains is arbitrary set to zero. The standard deviation for the averaged arrival-times of each pulse number is indicated by the shaded area. The side panels are showing the intra-train timing jitter distributions.

is the day-to-day evolution of these patterns. The X-ray pulses, arriving at the beamlines, are the results of a complicated and well-orchestrated interaction of hundreds of sub-systems in the photoinjector, the electron accelerator, and undulators with numerous variables, all influencing the final temporal, energetic, and spatial stability of the arriving X-ray pulses. Thus, minor changes in the settings of the machine could change the observed patterns. By observing these drift patterns and investigating their sources, even lower intra-train timing jitter seems to be achievable, increasing the possible time resolution. A first interesting follow-up measurement could be to repeat this campaign with a fully-filled X-ray pulse train. With the 100 X-ray pulses per pulse train, we only observed a small fraction of the, with 2700 fully-filled, possible pulse trains.

7.5. X-ray-induced Transient Refractive Index Change

We can probe the X-ray-induced transient refractive index change by evaluating the amplitude of the self-referenced arrival-time spectrum. To estimate the X-ray-induced transient refractive index change, we apply the simulation described in Chapter 6.2.3 to the actual experimental measurement. The X-ray-induced phase-shift, related to the transient refractive index (Eq. 6.16), is used as a free fitting parameter and determines the amplitude of the simulated self-referenced arrival-time signal. The fitting parameters of the simulation are set to reproduce the actual measured X-ray arrival-time signal. This way, the experimentally X-ray-induced refractive index change in the diamond sample can be estimated. Finally, we can compare the, in Chapter 3.3.2 and 3.3.3 described, Drude- and Maxwell Garnett model-predictions of the X-ray-induced refractive index change with the actual experimental results.

In order to simulate the arrival-time signal, the full spectrum and amplitude of the original 400 nm optical pulse are required. This reference spectrum is measured by rotating the second polarizer P_2 parallel to the first polarizer. Thus, in absence of an X-ray pulse, the complete optical pulse is transmitted. To protect the Gotthard detector from oversaturation, an additional neutral density filter (Thorlabs NE20A-A) with transmission of 0.05% at 410 nm was placed in front of the spectrometer. Hence, the original full bandwidth laser spectrum has a roughly 2 000 times higher amplitude than the experimentally measured self-referenced arrival-time signal. In this configuration we recorded over 5 000 individual laser pulses (Fig. 7.15 a). The optical reference spectrum, representing the original optical laser pulses, is fitted with a multi-Gaussian function (purple) and is used as the optical reference spectrum in the simulation.

As discussed before, the amplitudes of the measured X-ray arrival-time signals depend on the optical and X-ray pulse energy, as well as the spatial overlap between these two pulses. For this reason, we limit the analysis of the measured arrival-time spectra to those with the largest (0.5%) amplitudes of all the measured self-referenced arrival-time spectra

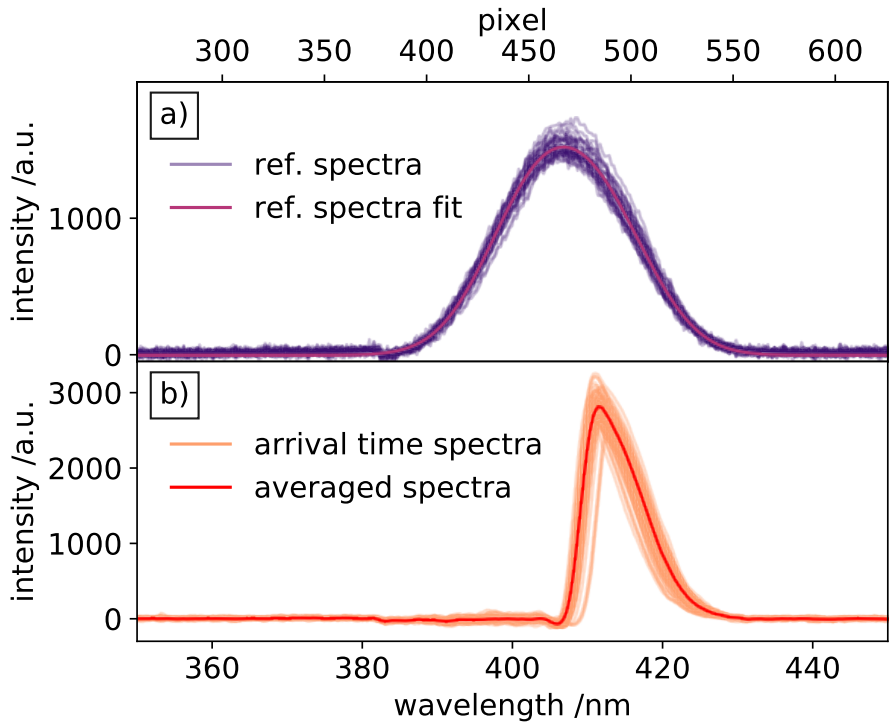

Figure 7.15.: Optical reference spectra (a) and X-ray arrival-time spectra (b), used for the analysis to estimate the X-ray-induced refractive index change. The purple multi-Gaussian fit is used for the timing-tool simulation as the optical reference spectrum. The orange self-referenced arrival-time spectra are a subset of all measured spectra with the 0.5% highest amplitudes. The red curve shows the average of these top amplitude spectra.

(Fig. 7.15 b). This ensures that the spatial overlap, as well as the optical and X-ray pulse energies were maximized for these pulse pairs. For the estimation of the X-ray-induced refractive index change, the simulation is scaled (by changing the variable of the X-ray-induced refractive index change) to fit the averaged arrival-time spectrum (red) of those experimentally measured arrival-time signals.

With the knowledge of the amplitude and spectral intensity distribution of the original optical pulse, a transform-limited electrical field of this optical pulse is generated and used as the starting point of the simulation. The thickness of the fused-silica glass block chirping the optical pulse, the relative arrival-time of the X-ray pulse, and the X-ray-induced refractive index change are adjusted to reproduce the experimentally measured X-ray arrival-time signal. As described in Chapter 6.2, the chirp of the optical pulse defines the wavelength-to-time mapping and, therefore, the slope of the cut-off edge. The relative arrival-time marks the spectral position of the cut-off edge and the X-ray-induced refractive index change defines the amplitude of the simulated arrival-time signal (relative to the incident optical pulse amplitude). The thickness of the a-cut αBBO-crystals (5 mm) and the electron cascading time of 100 fs are fixed values and are not changed during the simulations.

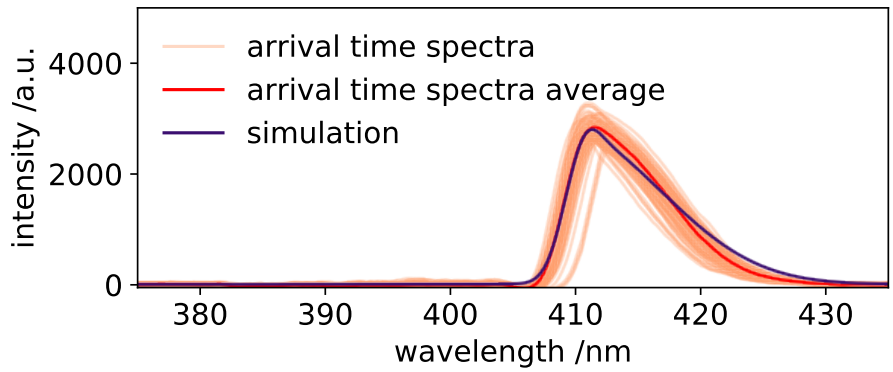

Figure 7.16.: The experimental data is reproduced with a simulation (blue), using the actual experimental conditions. The red curve is the selected arrival-time spectrum, recreated by the simulation and representing the average all top 0.5% arrival-time spectra (orange).

A simulation, adjusted to reproduce the measured X-ray arrival-time signal is illustrated in Fig. 7.16. The simulated arrival-time spectrum, shown in blue, recreates the red averaged arrival-time spectrum. The X-ray-induced refractive index change used in the simulation to match the amplitude of the experimentally determined spectrum is $\Delta n_{exp}=-5.7\times10^{-5}$. Since we used the averaged arrival-time spectrum of the most intense measured arrival-time spectra, it is reasonable to assume that the X-ray pulse energy was very close to the maximum measured X-ray pulse energy of 640 µJ and perfect spatial overlap between both pulses was achieved. Thus, the optical pulse was overlapped with the central region of the X-ray beam profile, where the X-ray-induced refractive index change is maximized. With the actual X-ray beam size at the sample position of (150±5) × (30±5) µm^2, the transient refractive index in the diamond sample can be calculated (Fig. 7.17). The fluence distribution is depicted in subfigure a) with a maximum fluence of 12 J/cm^2 in the center. Due to the Gaussian beam profile, the refractive index change of the diamond sample is not uniform over the footprint of the X-ray beam. The calculated X-ray-induced refractive index changes, using the experimental conditions described before, are shown in subfigure b) for the Drude model and c) for the Maxwell Garnett theory.

The theoretical estimated transient refractive index change in the center of the beam profile is $\Delta n_D=-2.9^{+0.1}_{-0.08}\times10^{-3}$ and $\Delta n_{MG}=-6.3^{+0.29}_{-0.26}\times10^{-5}$ for the Drude- and Maxwell Garnett theory, respectively. The estimated lower and upper errors are calculated for an X-ray beam size uncertainty of ±5 µm in both (vertical and horizontal) dimensions.

These theoretically predicted values of the X-ray-induced refractive index change are used for additional simulations to compare the experimentally obtained refractive index change with the theoretically predicted ones (Fig. 7.18). The simulation reproducing the experimental data (blue), uses the experimentally obtained refractive index change of $\Delta n_{exp}=-5.7\times10^{-5}$. The same simulation with identical experimental parameters, but now using the calculated X-ray-induced refractive index with the Drude and Maxwell

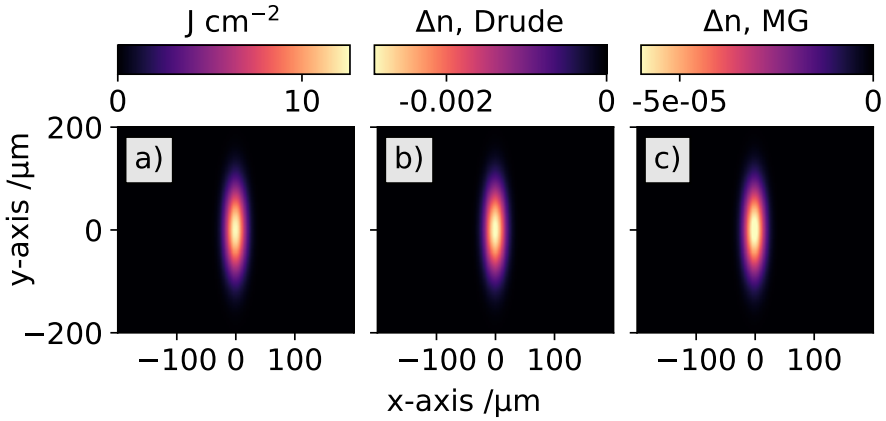

Figure 7.17.: The experimentally used X-ray beam profile at the sample position is shown in a) with a maximal power density of 12 J/cm^2 in the center. From the fluence profile, the expected X-ray-induced refractive index change can be calculated across the beam profile, shown in b) and c), using the Drude and Maxwell Garnett theory, respectively.

Garnett models are shown in purple and orange, respectively. The shaded area indicates the range of uncertainty due to the uncertainty of the experimental X-ray laser focus size in the diamond sample. The simulation recreating the experimentally measured data with the X-ray-induced refractive index change Δn_{exp} and the simulation using the X-ray-induced refractive index change Δn_{MG} of the Maxwell Garnett calculation are almost identical. The experimental data are well within the error bounds (orange shaded area) of the Maxwell Garnett model, suggesting good description of the X-ray-induced refractive index change. The Drude-Theory overestimates the X-ray-induced refractive index change by more than an order of magnitude. This result suggests that the old but rather simple description of the Maxwell Garnett theory of dielectric mixtures can deliver quite precise predictions of the X-ray-induced refractive index change in diamond and probably many other materials. As far as we know, this is also one of the first measurements directly probing the transient X-ray-induced refractive index change.

7.6. Detection Limits of the Setup

The experimentally obtained X-ray-induced refractive index change of $\Delta n_{exp}=-5.7 \times 10^{-5}$ and the calculated refractive index change of $\Delta n_{MG}=-5.9 \times 10^{-5}$ induce a phase-shift of 2.57° and 2.79°, respectively. This is sufficiently small to not be influenced by the phase-shift dependent cut-off position drift, which occurs for phase-shifts larger than 20° (see Section 6.2.3).

With the Maxwell Garnett model predicting the X-ray-induced refractive index change quite accurately, it can be used to estimate the minimum required refractive index change to be able to detect a self-referenced X-ray arrival-time spectrum with the used experimental conditions. To determine the X-ray arrival-time, the position of the cut-off edge

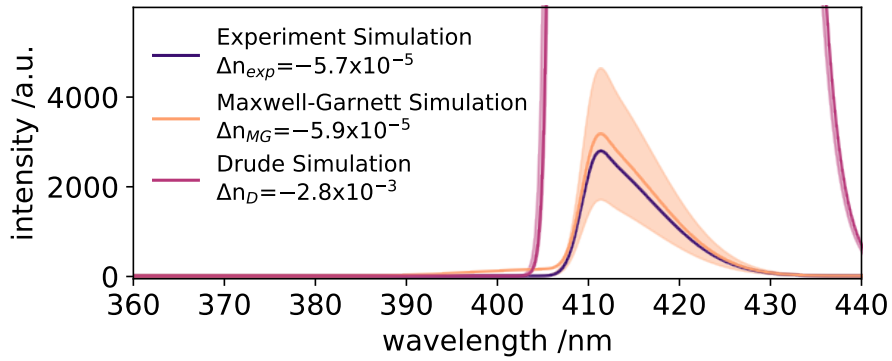

Figure 7.18.: Simulated X-ray arrival-time spectra. The simulation, which was tailored to the experimental data is shown in blue. Using the same simulation parameters, but using the theoretical calculated X-ray-induced refractive indices create the purple (Drude) and orange (Maxwell Garnett) arrival-time spectra. The shaded areas indicate the uncertainty of the X-ray-induced refractive index due to the uncertainty of the X-ray spot size in the diamond sample.

in the self-referenced arrival-time spectrum needs to be detected. The algorithm, described in Section 7.2, requires self-referenced arrival-time spectra with a SNR ≥ 2 to reliably detect the position of the cut-off edge. An example of such a low-amplitude arrival-time spectrum is shown in Fig. 7.19. The blue spectrum is experimentally measured. The blue shaded area indicates the noise of the spectrometer. The orange curve is a simulation recreating the experimental data with the orange shaded error bar indicating the uncertainty of the X-ray-induced refractive index change due to the X-ray focus size. The X-ray-induced refractive index change determined by the simulation is $\Delta n_{\text{exp\_min}} = -1 \times 10^{-5}$. This corresponds to an X-ray pulse energy of 90 µJ and a peak-fluence in the center of the Gaussian beam profile of 2 mJ/cm^2. The underlying Gaussian elevation, indicated by the purple dashed line, is the residual leakage of the original optical pulse through the crossed polarizers of the CPI.

The Gotthard detector was operated at about 20% of its saturation level throughout the experiment [185]. Operating it closer to saturation would increase the sensitivity to detect smaller X-ray-induced refractive index changes. The timing-tool simulation is used to estimate the minimal required X-ray-induced refractive index change needed, to still detect a timing signal with the used optical setup. The originally used optical laser intensity (dashed line) is up-scaled (solid line) (Fig. 7.15 a), such that the most intense experimentally measured arrival-time spectrum (Fig. 7.15 b) is scaled to reach the saturation of the detector (Fig. 7.15 c). By doing so, the originally smallest detectable arrival-time spectrum (orange Fig. 7.15 b) is now also up-scaled, such that it exceeds the minimal required SNR of 2 by a large margin. Therefore, the minimal detectable X-ray-induced refractive index change decreases.

The amplitude of the incoming reference optical pulse can not be scaled infinitely due

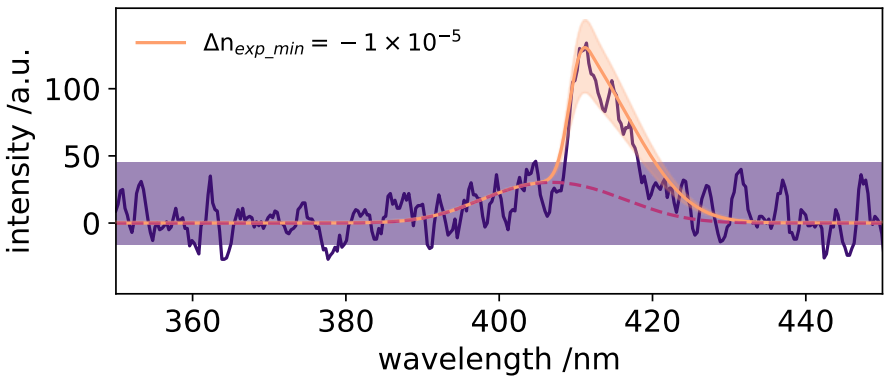

Figure 7.19.: X-ray arrival-time spectrum (blue) with the smallest required amplitude to fulfill a SNR greater than 2. The noise floor is indicated by the blue shaded area. The simulation reproducing the experimental spectrum is shown in orange. The error bars are indicated by the orange shaded area. The leakage of the original optical pulse through the crossed polarizer setup can be recognized and is implemented in the simulation and indicated by the purple dashed line.

to the leakage of the original optical pulse through the crossed polarizer setup. Scaling the amplitude of the original optical pulse such that the refractive index change of Δn_{exp}=-5.7×10^{-5} creates an arrival-time spectrum with an amplitude close to the saturation level of the detector, would cause leakage through the crossed polarizers, larger than the amplitude of a SNR $\approx$ 2 arrival-time spectrum (Fig. 7.21 a). The minimal measurable spectrum with a SNR of 2 is shown in blue. The leakage of the original laser pulse is shown as the purple dashed line with an amplitude larger than the amplitude of the arrival-time spectrum. With the used Nanoparticle Linear Film polarizers and its contrast ratio of 100 000 : 1, it would be impossible to obtain a background-free self-referenced arrival-time spectrum with such a low signal amplitude under these experimental conditions. Recently, polarizers with a contrast ratio of 1 000 000 : 1 can be commercially obtained [186]. With these polarizers, the leakage of the original optical pulse could be sufficiently suppressed, such that a self-referenced arrival-time spectrum with a SNR $\approx$ 2, could be measured background-free, even with the up-scaled amplitude of the original pulse (Fig. 7.21 b). The orange simulation, recreating the experimental spectrum (blue), using the up-scaled optical pulse, has an X-ray-induced refractive index change of Δn_{min}=-4.3×10^{-6}. This value marks the smallest possible refractive index change Δn to obtain a background-free X-ray arrival-time spectrum with the used setup and experimental conditions. Increasing the amplitude of the original laser pulse would increase the leakage through the crossed polarizers, such that a background-free measurement would not be possible anymore.

With the Maxwell Garnett model predicting the experimental data accurately, a forecast for the parameter range of the application of the currently used setup can be given. The main question of an operational timing-tool would be if a reliable X-ray arrival-time

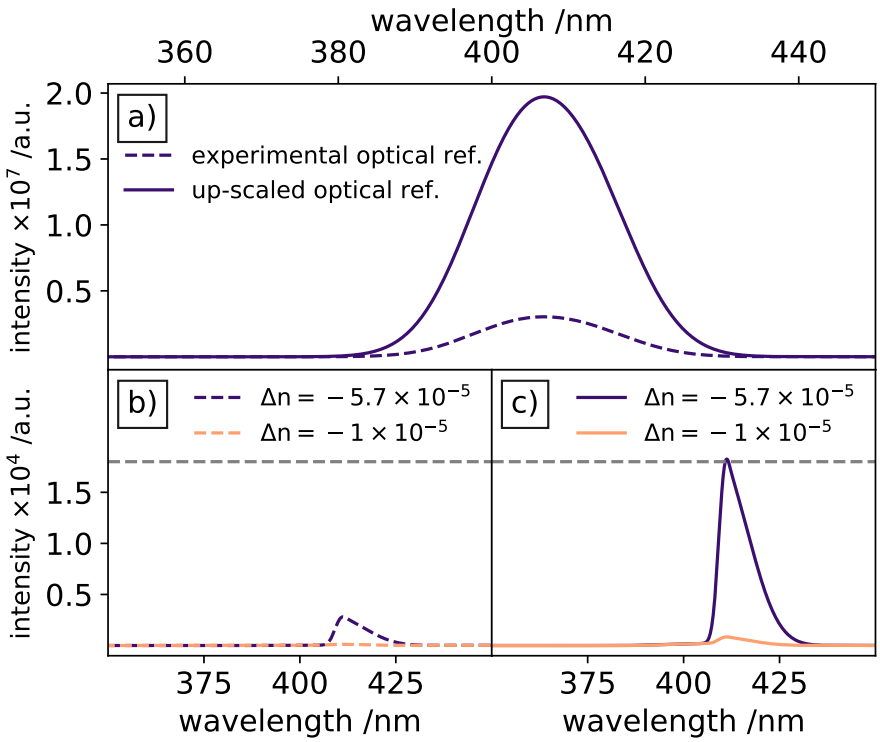

Figure 7.20.: Arrival-time spectra experimental sensitivity simulation. The originally used optical reference pulse used for the simulations is shown as the dashed line in a), while an up-scaled reference spectrum is shown as the solid line. Simulated arrival-time spectra, using the original reference spectrum, recreating the experimental data are shown in b), simulated for an X-ray-induced refractive index change of Δn_{exp_max} (blue) and Δn_{exp_min} (orange). Using the same refractive index change but with the up-scaled reference spectrum, the resulting arrival-time spectra have an increased amplitude, fully exploiting the dynamic range of the detector. The gray dashed line indicates the saturation level of the Gotthard detector.

spectrum can be measured. Typical X-ray pulse energies for experiments at EuXFEL can range up to 4 mJ, and the FWHM beam diameter can range anywhere between roughly 5 µm and 1 000 µm, depending on the focussing conditions and position of the timing-tool setup at the beamline. Two simulations for the expected SNR of the self-referenced timing-tool for a wide variety of pulse energies and X-ray diameters are shown in Fig. 7.22. The simulation in subfigure a) shows the simulation with the actually used experimental conditions, where a minimal required X-ray peak-fluence of 1.5 J/cm^2 (in the center of the Gaussian beam profile) is required to achieve a self-referenced X-ray arrival-time spectrum with a SNR ≥ 2. As explained above, operating the detector closer to its saturation level and using state-of-the-art polarizers can increase the sensitivity of the setup tremendously. With such an optimized setup, the sensitivity to detect X-ray-induced transient refractive index changes in the order of $\Delta n = -4 \times 10^{-6}$ seems to be a reasonable expectation, and the required X-ray peak-fluence would be reduced to 0.03 J/cm^2. This required peak-fluence is comparable, and even smaller, to reported

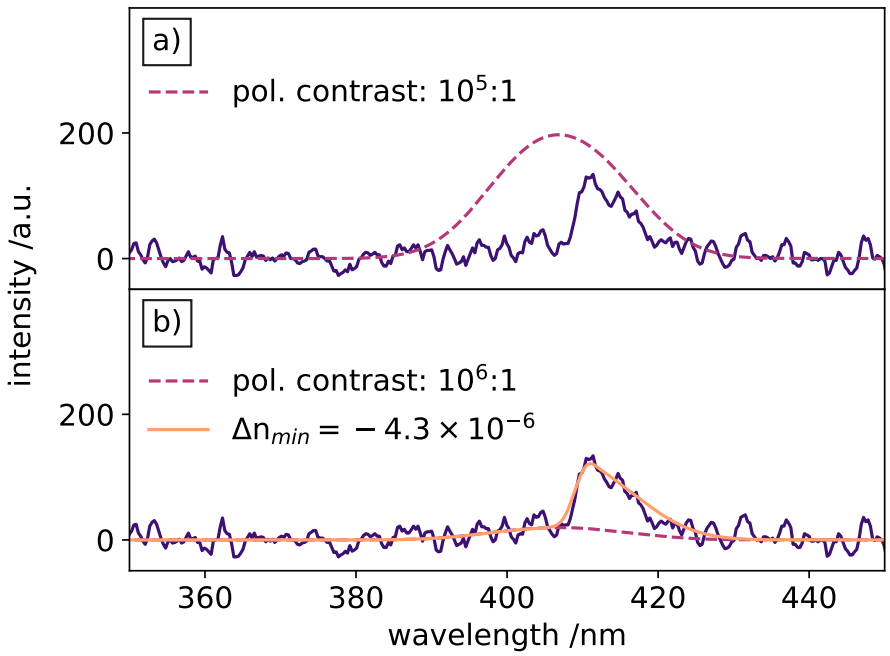

Figure 7.21.: Simulated self-referenced X-ray arrival-time spectra with an up-scaled amplitude of the reference optical pulse. A spectrum with the minimal required SNR is shown in blue. The used Nanoparticle Linear Film polarizers leak a substantial amount of the original laser pulse (purple dashed line), such that the amplitude of the leakage is higher than the experimental spectrum shown in a). With the best obtainable polarizers, the leakage can be suppressed enough to vanish within the noise floor as shown in b). A simulated self-referenced X-ray arrival-time spectrum, using this high contrast ratio polarizer, is shown in orange.

fluences in historical milestone spectral- and spatial encoding schemes as well as THz-streaking experiments, as listed in Section 4 Tab. 4.2.

7.7. Conclusion

In this chapter, the working principle of the self-referenced timing-tool scheme to determine the relative X-ray arrival-times was demonstrated. It was shown that with this scheme, it is possible to reliably use diamond as an interaction sample to measure the relative X-ray arrival-times. This paves the way for the reliable usage of timing-tools at intense MHz repetition X-ray facilities.

During the measurements, large pointing and pulse energy instabilities were observed, which were systematically described and analyzed in this chapter. The reports of our experimental team, as well as from other users, led to the formation of a EuXFEL internal task force to further investigate these instabilities.

The relative arrival-time jitter of the X-ray pulses at the FXE instrument was analyzed, revealing the superior synchronization of the OS scheme with a relative inter-train timing

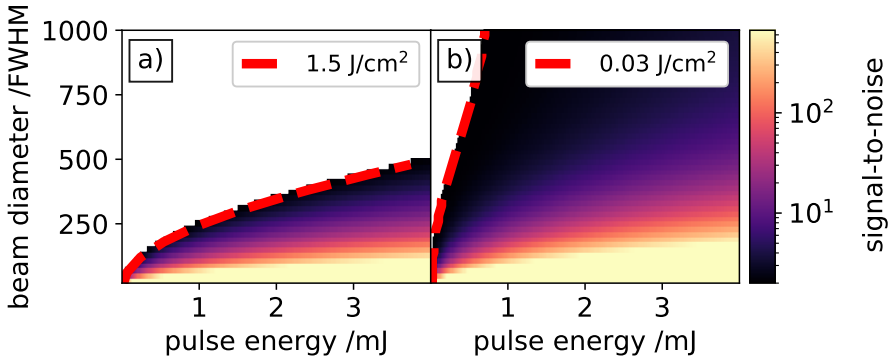

Figure 7.22.: Simulated SNR of the self-referenced arrival-time detection scheme at 9.3 keV for different X-ray pulse energies and FWHM beam diameters. Simulated SNR using the actual experimental conditions are shown in a). Expected self-referenced arrival-time spectra SNR with a fully optimized setup are shown in b). The red dashed line marks values where the SNR equals 2. The white area marks all conditions where the SNR would be below 2.

jitter of 90 fs FWHM. The MHz intra-train timing jitter was measured to be only 23 fs and 21 fs FWHM for the OS and RFS, respectively. This result suggests that the OS is stabilizing the timing jitter on the timescale of pulse trains (10 Hz), while the intra-train (MHz) timing jitter is identical for both synchronization schemes. Different intra-train drifting pattern could be observed with the RFS and OS scheme. While we used an X-ray train pattern with only the first 100 X-ray pulses filled, we could already observe differences in the intra-train drifting pattern when using the RFS and OS. Understanding the source of these drifting patterns could decrease the X-ray timing jitter even more.

Finally, the minimal required X-ray peak-fluence for the self-referenced detection scheme in diamond was discussed and estimated to be 0.03 J/cm^2 when slightly tweaking the experimental setup and using state-of-the-art polarizers. With the current setup, the required peak-fluence was determined to be 1.5 J/cm^2.

8. In-Situ X-ray Arrival-Time Measurement at SACLA

This chapter is based on our recently published paper [187], where we report the results of an XFEL beamtime at the SACLA facility in Japan. The main scientific goal for the measurement campaign was to demonstrate the capabilities of the self-referenced timing-tool as an in-situ method, measuring the arrival-time on the actual sample used in a scientific experiment. We used a free-flowing liquid flat-sheet as an interaction sample to measure the relative X-ray arrival-times, something not possible with other conventional timing-tool schemes. This new detection scheme was used together with the standard SACLA spatial encoding timing-tool located up-stream in the X-ray optics hutch. With the trusted SACLA timing-tool data, we benchmark the measured arrival-times of our timing-tool.
We show that with the self-referenced detection scheme, the actual scientific sample can be used as the timing-tool sample, thus, possibly enhancing the overall time resolution of the experiment by obtaining the arrival-time data directly from the sample. This is especially interesting for liquid chemistry experiments, where free-flowing liquid jets are used for sample delivery. These liquid jets can fall victim to mechanical vibrations caused by the high pressures these jets are operated on, nearby vacuum pumps, or other disturbing devices. Depending on the setup of the liquid jet, oscillations of hundreds of μm are possible, thus, introducing additional uncertainty of the timing jitter between optical and X-ray pulses when they are not in a co-linear geometry.
An actual liquid chemistry experiment was prepared by using a 100 mM aqueous solution of sodium iodide as the target sample. Similar to the X-ray-induced electron cascades described in Section 3.2.1, the X-ray pulses generate hot electrons which thermalize within 300 fs as solvated electrons [188] and change the refractive index of the sample.

8.1. Experimental Details

The fundamental concept of the detection scheme was discussed earlier in Section 5. In this Section, the details of the experimental setup used at SACLA beamtime are discussed.
The experiments were performed in the EH2 experimental hutch of SACLA. Throughout the experiment, the repetition rate was set to 30 Hz and the photon energy to 5.2 keV with a pulse energy of about 300 μJ. The X-ray beam spot size on the sample was set to a circular radius of 100 μm FWHM using the SACLA CRLs [189].

The facility's optical laser system is based on a commercial Ti:sapphire chirped-pulse amplification system operating at 1 kHz (Micra and Legend Elite, Coherent Inc.), which delivers pulses with a pulse duration of 25 fs FWHM and 15 mJ of pulse energy at a central wavelength of 800 nm [190, 118]. The laser is synchronized to the accelerator's main clock by locking the oscillator cavity length using a commercial standard phase-locked loop [190] (Synchrolock-AP, Coherent Inc.). A pulse picker selects a subset of optical pulses from the 1 kHz laser amplifier output to match the X-ray repetition rate. The time-delay between the optical laser and X-ray pulses can be freely adjusted with sub-ps accuracy using a trigger and clock delay module (84DgR5C01, CANDOX Systems Inc.), while for fs-tuning, an additional optical delay line is used.

In our experiment (Fig. 8.1), we split the optical laser beam into three paths. The first one was used for the operation of the facility's standard spatial encoding timing-tool (TT), which is based on the spatial encoding technique and located a few meters upstream of the experimental hutch EH2 [120, 121]. The second beam was used for exciting the actual sample in our pump-probe geometry and for calibration of our self-referenced timing-tool. For this, the 800 nm optical pulse was frequency-doubled (second-harmonic generation, SHG) in an appropriate BBO crystal, creating a wavelength of 400 nm. Finally, the third beam was used for the simultaneous measurement of the X-ray/optical relative arrival-time at the sample position and on the sample itself using our novel detection technique.

For the self-referenced detection scheme, 800 μJ of the 800 nm fundamental laser pulses were focused inside a 1.5 m long tube filled with 1.5 bar pressurized Argon gas, generating a broadband supercontinuum pulse [191] ranging from 300 nm to over 1000 nm. The Argon cell and several transmissive optics in the beam path exhibit sufficient dispersion to chirp the continuum to a pulse duration of approximately 2.5 ps within a narrow wavelength range of about 330-450 nm. The supercontinuum laser pulse was guided over an optical delay line (Δt_{SC}) to the self-referenced setup. To prevent linear or non-linear sample excitation by the supercontinuum pulse, a BG38 color glass filter was used to block the 800 nm contribution from the supercontinuum, reducing the supercontinuum pulse energy to below 1 μJ.

The broadband optical pulses were guided through the CPI to generate the self-referenced X-ray arrival-time signal. Two Glan-Taylor (GT10, Thorlabs) calcite polarizers were used. The two BC were two a-cut α-BBO crystals with a thickness of 5 mm.

The X-ray-induced arrival-time signal was guided to a spectrograph (Shamrock 193i, Andor) with a focal length of 193 mm and a 600 grooves/mm grating with a blaze wavelength of 500 nm. The dispersed optical spectrum was recorded with a 2-d detector (Opal, Adimec) camera with 1920×1080 pixels and a pixel size of 5.5 × 5.5 μm. The high dispersion of the chosen grating limited the recorded spectral region to about $\Delta\lambda \approx$100 nm around the 400 nm region of the spectrum and ultimately defined the temporal window of the measurement as well as its final time resolution.

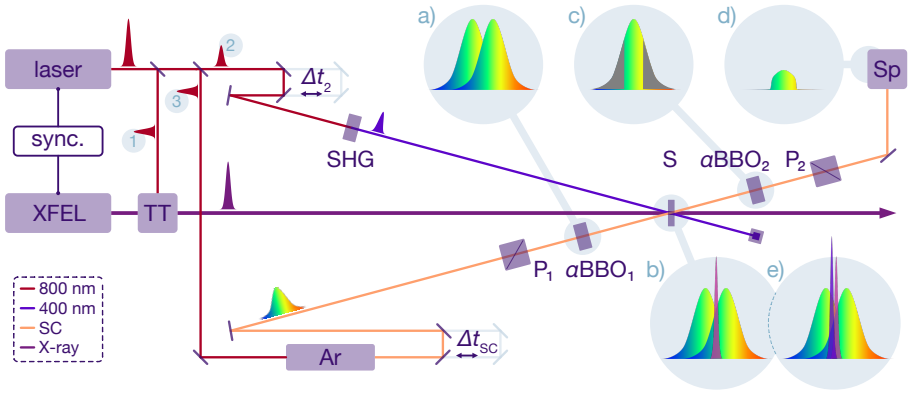

Figure 8.1.: Self-referenced timing-tool setup at SACLA. The 800 nm beam of the SACLA's synchronized optical laser is split into three branches: the first beam (1) is guided to the facility's standard timing-tool TT, the second beam (2) is frequency-doubled (second-harmonic generation, SHG) and serves both for temporal calibration of the timing-tool without X-ray pulses and for sample excitation with X-ray probing pulses. The third beam (3) is used to measure the relative X-ray arrival-times with the self-referenced detection scheme. As described before, the CPI consists of the polarizers (P1) and (P2) and the BC αBBO_1 and αBBO_2. The optical pulse with its two different PC, created by αBBO_1, is shown in a). The two PCs overlapped with the X-ray pulse in the sample are shown in b). c) shows the two temporally synchronized PCs behind the second BC, where the grey areas indicate spectral regions of both PCs, which are perfectly synchronized. The residual self-referenced timing signal behind the second polarizer observed in the spectrometer (SP) is shown in d). In e), the two broadband PCs overlapped with the X-ray pulse, and the additional intense optical pump pulse are shown.

In an actual scientific experiment, where the self-referenced timing-tool was used as an in-situ arrival-time monitor, the intense 400 nm laser pulse was used to excite the sample. Thus, an additional optical pulse arrived together with the supercontinuum timing-tool pulse and the X-ray pulse in a narrow time window.
The wavelength calibration of the spectrometer was accomplished using a Princeton Instruments IntelliCal calibrated Ar/Hg light source. Due to the small wavelength region dispersed onto the camera of only ~100 nm and the very limited preparation time before the experiment, the pixel to wavelength calibration was estimated to be linear. This simplification does not affect the analysis of the experiment, since the wavelength is only used to have a more familiar physical value than just the plain pixel positions. The whole analysis uses pixel positions, which are directly converted to a temporal axis (see below), never using the wavelength values.

Sample Delivery System

Using a free-flowing liquid jet removes the X-ray-induced heat pile-up problem. Instead of ensuring that the used sample can withstand the X-ray pulses at a high repetition rate, another approach is to refresh the actual sample for every single X-ray pulse.

The free-flowing liquid jet system (Microliquids GmbH) used in this experiment produces a thin flat-sheet of liquid based on the collision of two round jets under an angle of 30° to 50°, each with selected diameters of a few tens of μm. Adjustments of the backing pressure, the selected nozzle diameter, the relative position and angle of each nozzle, each influences the shape, thickness, flatness, and stability of the sheet, allowing a precise control as a function of the viscosity of the liquid. With this, adjustable flat-sheet thicknesses from about 2 μm to 300 μm are realized.
We used two glass nozzles with 100 μm diameter orifices and a volumetric flow rate of up to 80 ml/min (corresponding to flat-sheet speeds up to 60 m/s, see below) using a commercial HPLC pump (Shimadzu LC-20AP). This yields a leaf-shaped area of the flat-sheet of about 0.5×5 mm^2 (horizontal $\times$ vertical dimensions) with a thickness of below 20 μm. This type of liquid jet had been originally developed for soft X-ray spectroscopy experiments [192], including appropriate vacuum environments (approx. 10^{-3} mbar). However, we optimized the setup for ambient conditions or helium atmospheres as typically used at hard X-ray instruments. In particular, due to the high volumetric flow rate, a catcher assembly enables the recycling of the sample solution with a minimum total sample volume of approximately 50 ml.
Before the start of the arrival-time measurements, we determined the thickness of the flat-sheet liquid jet with a commercial device. It is based on confocal achromatic imaging, where the light of a broadband source (typically an LED) is imaged through a chromatic lens yielding a dispersion of monochromatic light along the z-axis, i.e., the foci of different wavelengths are dispersed along the z-axis. Placing a thin transparent optic in this longitudinally dispersive region, here the thin flat sheet jet, only two single wavelengths will be efficiently reflected into the chromatic lens, one from the front and the other from the backside of the sample. These two wavelengths are then imaged through a filtering pinhole, which suppresses all other wavelengths not focused on the sample. A spectrometer determines both back-reflected wavelengths. The thickness of the sample is then calculated by the relationship between the chromatic focusing distances and the recorded wavelengths. We used a commercial device for this purpose (TopSens CCS Prima with optical pen CL0-MG140, Polytec/STIL), which allows thicknesses measurements in the range from sub-5 μm up to 100 μm at a working distance of approximately 2.7 mm. Because of the size of the device and the short working distance, this method can not be used as an online thickness monitor and is used only prior to the actual arrival-time measurements to set the desired sheet thickness. The measurement of the thickness of the flat-sheet liquid jet yield a sheet thickness of (10 ± 1.4) μm (Fig. 8.2). The integration time of each data point is 100 ms, meaning that only the average thickness of the liquid jet in this time period can be measured.

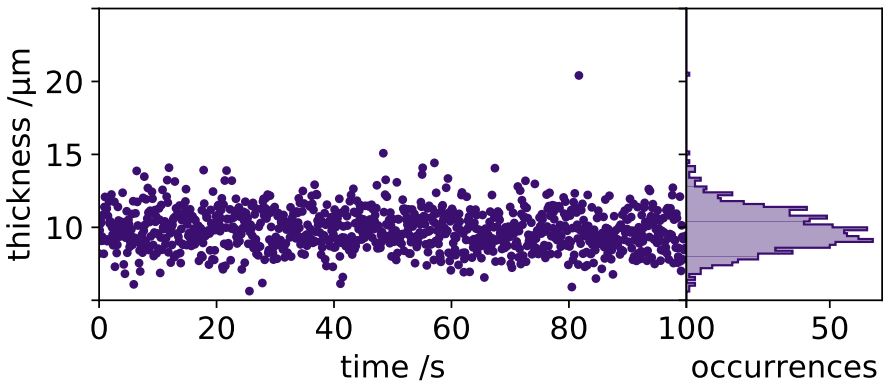

Figure 8.2.: Thickness measurement of the colliding liquid jet, using confocal achromatic imaging.

MHz Compatibility of the Flat-Sheet Liquid Jet

The flow speed of the liquid jet is an important figure of merit for the MHz intra-train repetition rate at EuXFEL and for the upcoming LCLS II. Ideally, the irradiated volume by the X-ray and the optical laser pulses are completely replaced for every shot. For instance, for a maximum spot size of 25 µm for both beams on the colliding jet, a linear flow speed of 25 m/s is required to accommodate a repetition rate of 1 MHz, corresponding to an X-ray pulse spacing of 1 µs in case of LCLS II and similar for the 1.1 MHz special pulse pattern offered at EuXFEL. For 4.5 MHz, the maximum intra-train repetition rate at EuXFEL, corresponding to a minimum X-ray pulse spacing of 222 ns, jet flow speeds beyond 113 m/s are required. Together with colleagues from the SFX/SPB Instrument and the Sample Environment Group at EuXFEL, we investigated the flow speed of the colliding liquid jet already in Hamburg in our laser lab. To determine the flow speed of our liquid jet, we treated the flowing sheet with an intense ultrashort laser pulse, evaporating the sample volume in the focus of the laser pulse and generating a small hole. The speed of this hole is then tracked as it flows with the remaining sample volume of the liquid jet. For this investigation, laser pulses were striking the liquid sheet at a 3 kHz repetition rate, and the evolution of each impact was tracked with a fast recording camera and a nanosecond gated camera using the time-lapse technique. In a first series of measurements, the back-illuminated liquid jet is imaged using a microscope objective onto the sensor of a high-speed camera (Photron FastCam SA4), which is capable of recording up to 500,000 frames per second with a minimum shutter opening time of 1 µs. This records the development of one single impact over a time period of up to 110 µs. In a second series of measurements aiming to record the faster nanosecond timescales, we used a gated image intensifier (Hamamatsu C9538-03). It records triggered images at fixed time-delays, thus, averaging over several individual shots for each time point. We reconstruct a time-lapse movie using different time-delays over a range of 50 µs. Both techniques combined delivered a movie of the laser-induced distortion flowing along with the liquid jet. (Fig.

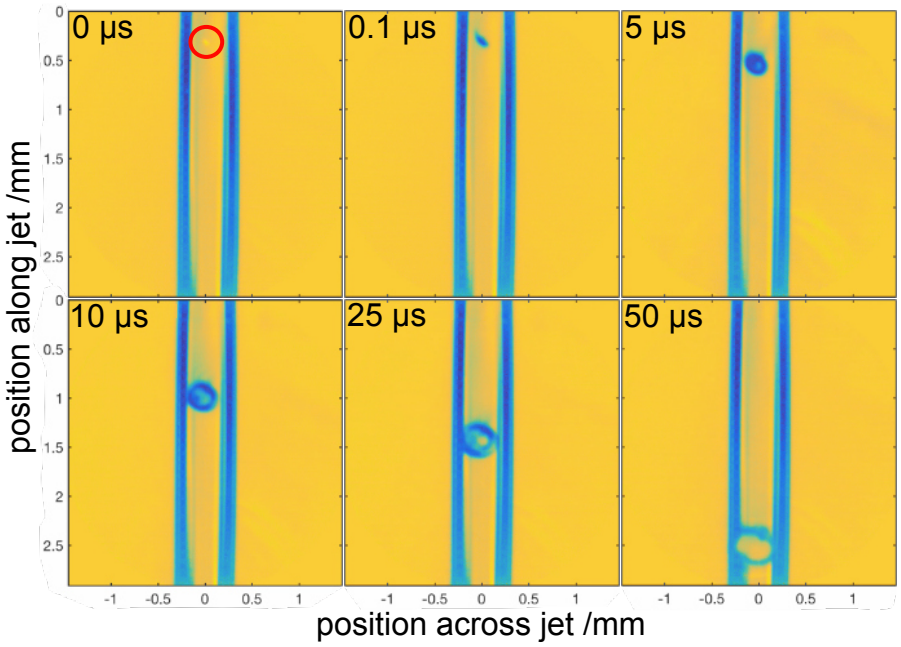

Figure 8.3.: Colliding jet flow speed measurement. Focussed 1 mJ, 800 nm, 35 fs strike the liquid jet at 0 µs. The distortion on the liquid jet is tracked over time as it propagates together with the liquid of the jet.

8.3). The pixel to length calibration was obtained by inserting a variable line grating at the position of the liquid jet.
By analyzing the traveled distance of the laser-induced distortion in Fig. 8.3, a flow speed of our liquid jet system (under stable conditions in terms of thickness and flatness) of up to approximately 50 m/s was measured. This flow speed was recorded using a flow rate of 70 mL/min and two 100 µm nozzles to form the flat-sheet jet. Using a slightly higher flow rate of 80 mL/min, the liquid jet speed reached 60 m/s but showed minor instabilities in flatness. While this only approaches the required refreshment rate at 4 MHz, it already fulfills the requirements for experiments at LCLS-II (1 MHz) and at EuXFEL for 0.5 MHz, 1.1 MHz and 2.25 MHz modes. With additional efforts, it is within reach to utilize fully refreshed samples at EuXFEL's highest intra-train repetition rate of 4.5 MHz with the colliding jet system.

8.2. Raw Data Analysis

The data for our timing-tool and the SACLA spatial encoding timing-tool were both recorded with identical 2d detectors, described in the setup section in this chapter. Both detectors have 1920×1920 pixels. The detector for the self-referenced signal is mounted in the focal plane of the spectrograph, thus, the spectrum is dispersed in a line focus over the detector. The spectrum is dispersed over the entire horizontal axis of the detector but covers only the pixel rows between 750 and 850. These pixel rows are vertically integrated (summed over) to retrieve the self-referenced spectrum (Fig. 8.4 a).

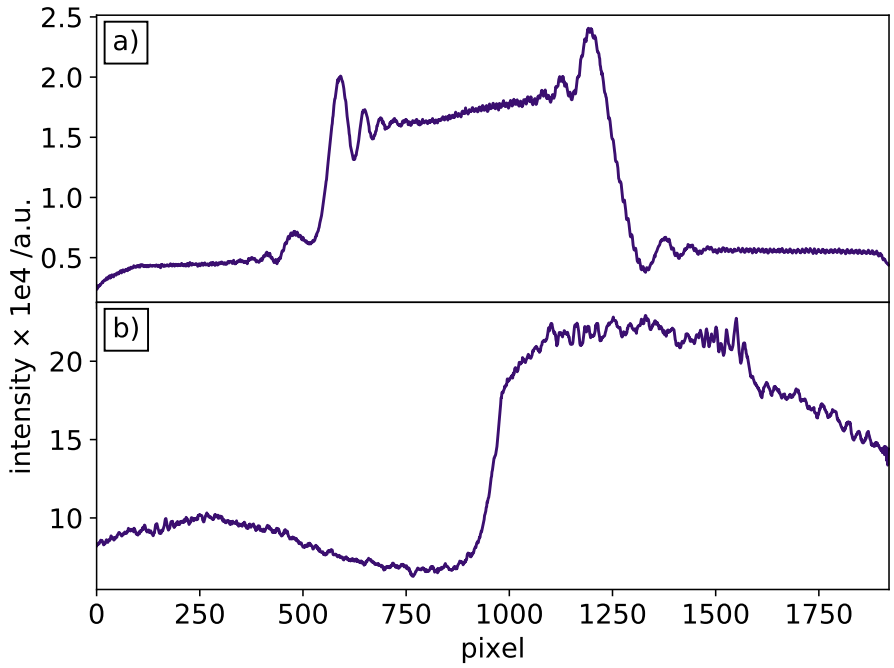

Figure 8.4.: Raw arrival-time data from the self-referenced timing-tool a) and SACLA spatial encoding timing-tool b). Both traces are integrated over multiple pixel rows of the 2d detectors.

We used the SACLA spatial encoding scheme (described in Section 4.3.1) in its standard configuration [120]. To increase the X-ray intensity on the spatial encoding GaAs sample, the X-ray pulses were focussed in the vertical direction with an elliptical X-ray mirror, impinging on the sample in a 45° angle, yielding a footprint on the sample of 3 (V) × 780 (H) μm^2. The optical laser with a normal incidence angle was focussed with a cylindrical lens to a beam size of 400 (V) × 2000 (H) μm^2 to cover the X-ray footprint on the GaAs sample. The transmitted optical beam profile was imaged on the dedicated 2d pixel detector. The imaged beam profile covered the entire detector in the horizontal direction and the pixel rows from 140 to 200. These rows were integrated vertically to retrieve the spatial encoding arrival-time signals (Fig. 8.4 b).

The self-referenced arrival-time spectrum shows two distinct cut-off edges, each indicating the same X-ray arrival-time (see Section 6.2.3). Observing both cut-off edges is a clear indicator that the X-ray pulse arrived in a temporal window, where both PCs are partly overlapped. The pronounced oscillations, especially in the cut-off edge regions, are caused by thin-film interferences [91] when transmitting the two broadband PCs through the 10 µm thick liquid jet. The background is close to 5000 counts, and the slight linear slope of the background intensity from lower to higher pixels numbers is caused by imperfect shielding of the ambient light. The intensity drop in the pixel region from 0 to 100 and 1800 to 1920 is caused by the frame of the detector, such that these pixel regions are in a shadow caused by the detector housing frame where the ambient light leakage can not reach these pixels.

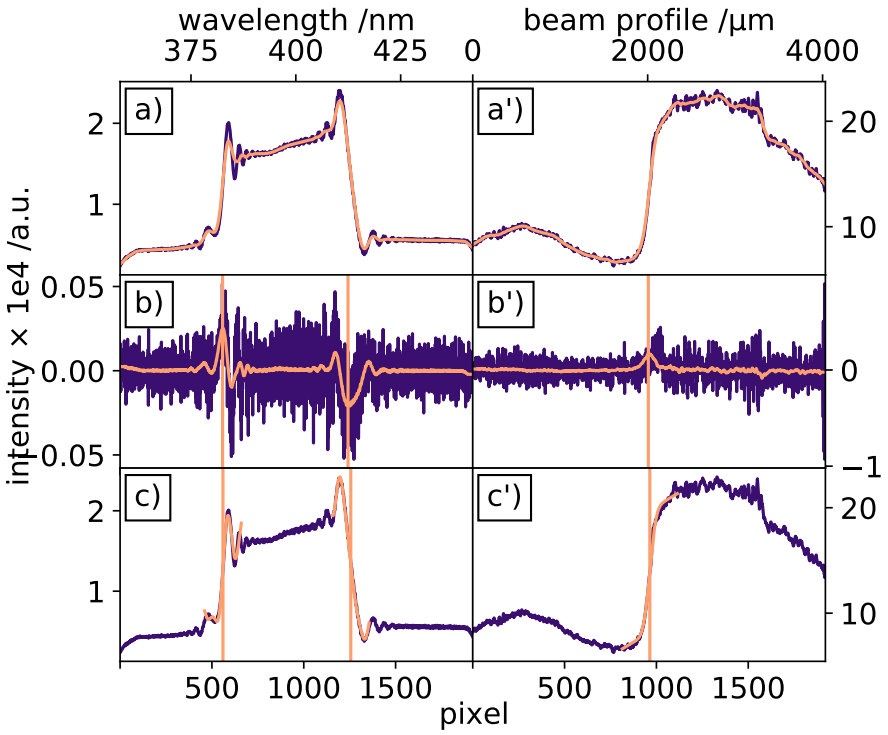

Figure 8.5.: Raw arrival-time data from the self-referenced timing-tool a) and SACLA spatial encoding timing-tool b). Both traces are integrated over multiple pixel rows of the 2d detectors.

To analyze the relative X-ray arrival-time, the X-ray-induced arrival-time signal in the self-referenced detection scheme and the SACLA spatial encoding scheme need to be reliably detected with the highest possible precision. Both arrival-time signals are defined by a noticeable step-like edge. In the case of the self-referenced arrival-time spectrum, even two edges can be used to determine the X-ray arrival-time. We used our own algorithm to analyze the X-ray-induced arrival-time edge position, also for the SACLA spatial encoding timing-tool, instead of the SACLA standard analysis tool [193]. This should ensure that both arrival-time signals are analyzed the same way and should prevent a systematical error caused by two different analysis schemes.
We use the same edge detection algorithm as discussed before in the EuXFEL experiment in Section 7.2. The only difference is an altered fitting function:

$$g(x) = A_1 \left(1 + \mathrm{erf}\left(\frac{\mathrm{x} - \mu}{\sigma}\right)\right) + A_2 \sin\left(\nu(x - \phi)\right) + C \ . \tag{8.1}$$

In addition to the fit function $f(x)$ used in the EuXFEL experiment, we added a sine function to the fitting function to be able to take the thin film interferences into account. As before, the Gaussian error function amplitude is defined by A_1, the center of the error function by μ and the width by σ. The amplitude of the sine-function is defined by A_2, its frequency by ν and its phase by ϕ. An example of the edge fitting algorithm on the

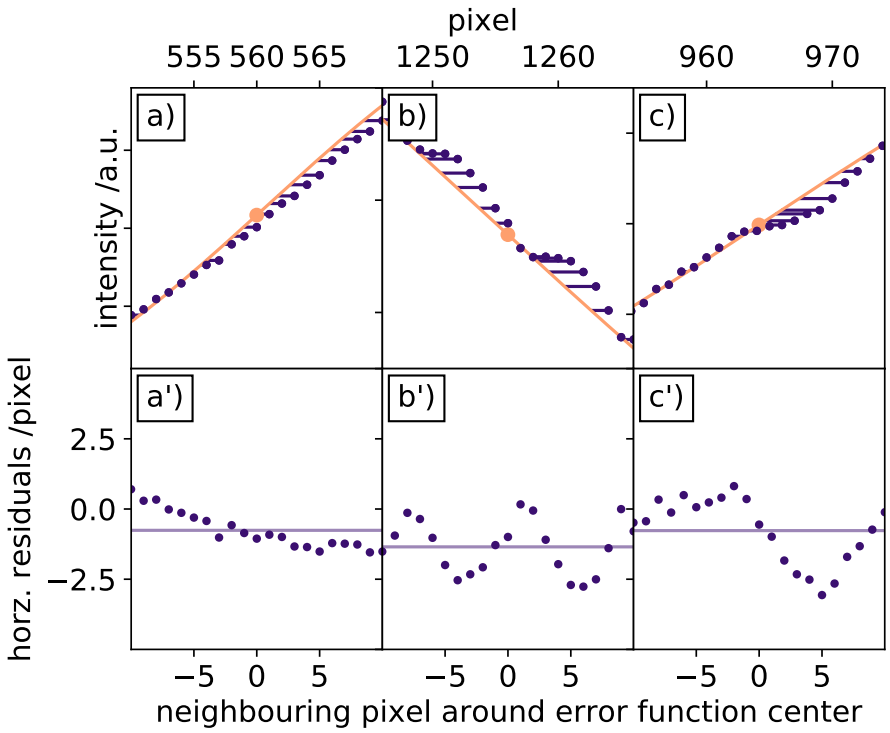

Figure 8.6.: Fit precision of the important error function parameter μ. The horizontal distance of the experimental (blue) data points are measured to the fit function (orange) for the two self-referenced edges a) and b) and the SACLA timing-tool c). The corresponding measured relative distances are shown in a′), b′) and c′). The horizontal line indicates the mean value.

SACLA experimental data is shown in Fig. 8.5. A self-referenced arrival-time spectrum (blue) is shown in a) together with the smoothed spectrum (orange). The corresponding arrival-time trace measured with the SACLA spatial encoding scheme with the identical X-ray pulse is shown in a′). The first derivatives of the two arrival-time signals are shown in b) and b′) for the self-referenced detection scheme and SACLA spatial encoding scheme, respectively. In the self-referenced detection scheme, both X-ray-induced arrival-time edges are detected using the maxima (left) or minima (right) of the first derivative of the smoothed signal. Identically, the position of the X-ray-induced arrival-time signal in the SACLA spatial encoding scheme is detected.
The final fits, using the fitting function $g(x)$ and the estimated edge position from b) and b′) are shown in Fig. 8.5 c) and c′).
Similar to the EuXFEL experiment, we determine the horizontal residuals by measuring the horizontal distance between the determined fit function and the experimental data as illustrated in Fig. 8.6 a) and b) for the two self-referenced edges and for the SACLA spatial encoding tool in c). These fitting residuals are later on to estimate the time-resolution of the experiment.

8.3. Temporal Calibration

The arrival-time for each X-ray pulse can be retrieved by precisely determining the pixel position of each rising and falling edge of the self-referenced signal. For this, it is required to map each pixel, and thus each wavelength of the optical spectrum, into the time domain. The resulting time axis is a function of the chirp of the supercontinuum pulse. To reliably convert the wavelength, or detector pixel values, to a time axis, we operated the self-referenced setup with the optical 400 nm excitation laser pulse instead of the X-ray pulse. Since the supercontinuum and the 400 nm pulses originate from the same source, their residual timing jitter is sufficiently small, such that the arrival-time between these pulses can be precisely adjusted with an optical delay line (Δt_2 in Fig. 8.1). We recorded 50 individual self-referenced spectra for each of the 48 different time-delays in 100 fs steps. Four time-delays are illustrated in Fig. 8.7 a). Polycrystalline SiO_2 with a thickness of 90 µm was used for these calibration measurements instead of the flat-sheet jet. The solid sample was chosen for its simple setup but also to eliminate additional timing jitter, which potentially could be caused by a fluctuating liquid jet. In stark contrast to the X-ray-induced self-referenced timing signal illustrated in Fig 8.4 a), the laser-induced self-referenced timing signal looks quite different. Due to the large bandgap of SiO_2 of 8.9 eV [194], a non-linear absorption of the 400 nm optical pump pulse is needed to generate one conduction band electron. Thus, the optical-induced electron density is smaller than the X-ray-induced electron density, resulting in a weaker, nearly not detectable, self-referenced timing signal. The optical signal is dominated by two coherent artifacts at the spectral position where the leading and trailing edges of the X-ray-induced signal would be expected and is non-existent in between.

The mapping of pixel positions into the time domain is determined by evaluating the edge position for each spectrum of a single time-delay, using the fitting function $g(x)$ to fit the edges, as shown in Fig. 8.7 b). The analysis of the horizontal residuals of all 50 shots at one fixed time-delay yields an error of Δcen_l=2.2 pixels FWHM for the leading edges and Δcen_t=1.5 pixels FWHM for the trailing edges, as indicated by the vertical shaded rectangles in the example in Fig. 8.7 c). Together with the small variations of the error function edge position μ for the individual fits, the overall uncertainty of the time calibration accumulates to $\pm$6 fs FWHM. The entire set of calibration points for converting a pixel position into a relative time-delay for each leading and trailing edge is fitted with a second order polynomial function, where time zero is set arbitrarily to the center of the detector. It delivers the calibration curve shown in Fig. 8.7 d), which results in the aforementioned wavelength-dependent accuracy to determine a time point within a precision of $\pm$2 fs for any given detector pixel for the self-referenced detection scheme.

The simultaneously recorded data of the established SACLA timing-tool based on spatial encoding has been analyzed using the same fitting function $g(x)$. The time calibration of the spatial encoding timing scheme is purely geometrical, and we used the numbers and calibration reported by [189].

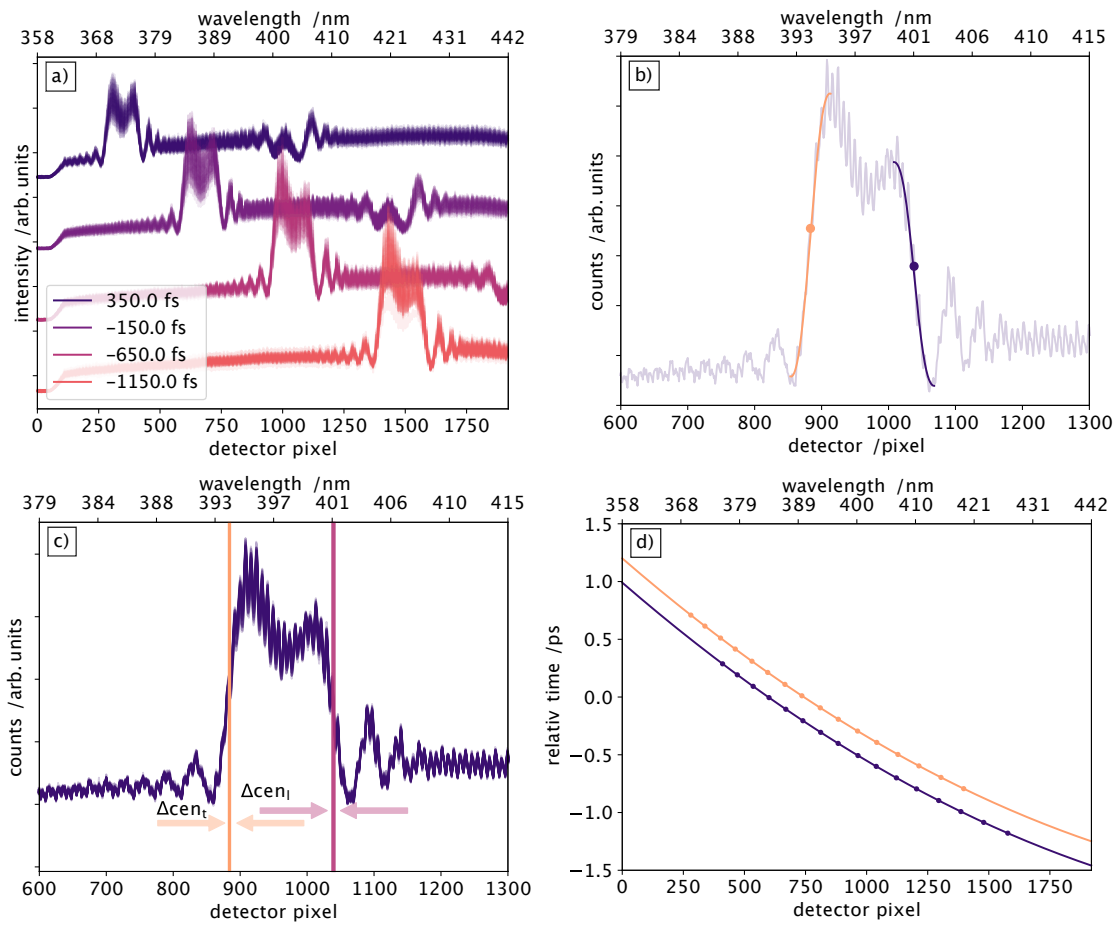

Figure 8.7.: Calibration of the self-referenced arrival-time measurement setup, utilizing a 90 µm thick SiO_2 plate and laser pulses at a wavelength of 400 nm from the same laser system as used for the supercontinuum generation. At each time-delay of the 400 nm and the supercontinuum pulses, 50 spectra are recorded, which move from higher to lower wavelengths for increasing time-delays a). Both edges of every recorded spectrum are fitted using $g(x)$, with one example shown in b) with leading (blue) and trailing (orange) edge fits. The center of the error function is indicated by the dot. The 50 individual pump-probe events produce an edge position within an error 2.2 pixels as indicated by the vertical rectangles in c). The mapping of detector pixels to fs time-delay results in a calibration function d).

8.4. Experimental Results

The goal of this experiment is to demonstrate the reliable operation of the self-referenced timing-tool scheme in the same sample used for a scientific experiment, in particular that it delivers accurate arrival-time information for each X-ray pulse in the presence of an additional intense laser stimulus. Therefore, this scheme will provide additional flexibility towards a variety of samples used in liquid phase structural dynamics experiments. Since we expected an increased sensitivity of this scheme to extract accurate arrival-time information, we attenuated the X-ray pulses with a 50 µm thick Al foil, transmitting only

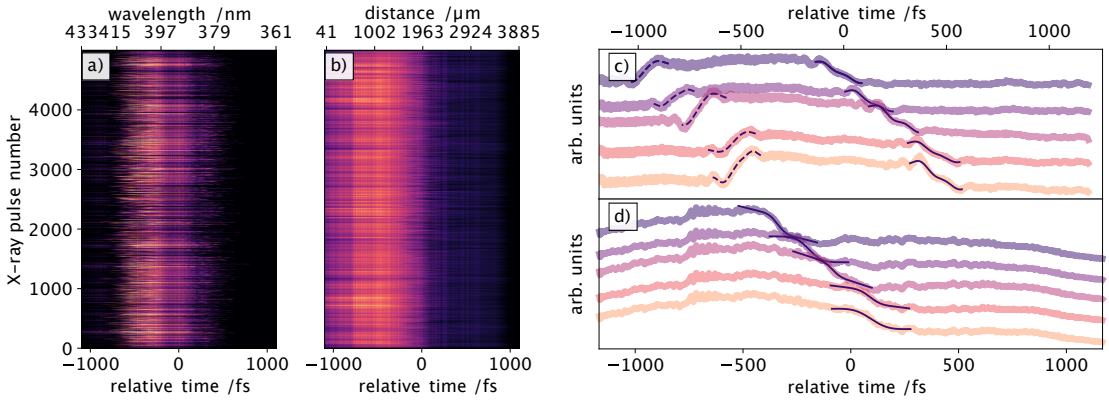

Figure 8.8.: Raw data of the self-referenced timing-tool a) and SACLA spatial encoding timing-tool b), recorded for 5000 X-ray pulses at low fluence. Five consecutive X-ray arrival-time measurements and corresponding edge-fits for the self-referenced timing-tool c) and SACLA timing-tool d) are shown, displaying the same sequence of events from top to bottom traces, revealing the same overall trend.

10% of the original pulse energy, i.e., 30 µJ. Even under these conditions and using a relatively large X-ray spot size with an FWHM diameter of 100 µm on the sample and a corresponding X-ray peak-fluence in the center of the Gaussian beam profile of 0.2 J/cm^2, a clear timing signal is measured, as shown in Fig. 8.4 a). With the reported [121] dimensions of the line focus for the SACLA spatial encoding scheme (3 µm × 780 µm), using roughly two percent of the original X-ray pulse energy, split from the main beam with a Si grating, the X-ray peak-fluence of 0.3 J/cm^2 is a little bit higher than the 0.2 J/cm^2 used for the self-referenced detection scheme.

In this comparison study, 5000 individual X-ray pulses were recorded with each timing-tool. All 5000 individual spectra / arrival-time traces are shown in Fig. 8.8 a) and b) for the self-referenced detection scheme and SACLA spatial encoding timing-tool, respectively. Each row is a single arrival-time trace with the x-axis already converted to a relative time, using the pixel to time calibration. In Fig. 8.8 c) five consecutive arrival-time measurements with the self-referenced timing-tool are shown together with the edge fits to determine the edge positions. The corresponding X-ray pulse arrival-time measurements with the SACLA timing-tool are shown in Fig. 8.8 d) together with the determined edge fits. For these five X-ray pulses a clear matching pattern in the measured arrival-times can be observed when comparing both measurement methods.

To further investigate the performance of the self-referenced timing-tool, we analyze the correlation of the measured relative arrival-times obtained with the two timing-tools. The correlation between two measurements X and Y can be quantified by the Pearson correlation ρ [195, 196]

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \ , \tag{8.2}$$

where $\mathrm{cov}(X, Y)$ is the covariance of the two measurements and σ_X and σ_Y their standard deviation. A Pearson correlation of $\rho_{X,Y} = 1$ corresponds to a perfect correlation and a value of 0 to a completely uncorrelated measurement. We use this relation to quantify the correlation between the self-referenced and the spatial timing-tools in a direct comparison study.

As shown in Fig. 8.9, we observe a excellent correlation with a Pearson correlation value of 0.98, for 89% of the data, when comparing the arrival-times measured with both timing-tools. The remaining 11% of the data did not deliver a trustworthy measurement of the timing signal, mainly due to occasional low X-ray pulse energy shots for the spatial encoding scheme, but also due to occasional and random disturbances of the flat-sheet liquid jet, which scattered the optical beam away from the spectrometer entrance. Those X-ray arrival-time measurements, where the parameter defining the center position of the fitting function, in either of the two timing-tools, had an error larger than 3σ, were rejected from the analysis.

In a perfect correlation, i.e., a Pearson value of 1, all data points would lie on the orange diagonal line in Fig. 8.9, such that any deviation from this line is a measure for the uncertainty between the two schemes, resulting to about 39 fs FWHM (inset in Fig. 8.9). A major contribution to the residual timing jitter can be caused by the flat-sheet liquid jet. In such systems with connections to powerful liquid pumps, mechanical vibrations can never be fully prevented. Spatial oscillations of the liquid flat-sheet along the X-ray axis can introduce an additional timing jitter, not recognized by the remote SACLA timing-tool. In our configuration, with an angle between the X-ray and optical pulses of 10°, a spatial oscillation of $\pm 350\,\mu$m would be enough to introduce the measured residual timing jitter.

The relative arrival-time distribution itself (X-ray timing jitter) measured by the self-referenced timing-tool and SACLA spatial encoding deliver the histograms shown in Fig. 8.9 with almost identical FWHM values of (513 $\pm$ 22) fs and (510 $\pm$ 39) fs for the self-referenced timing-tool and the spatial encoding scheme, respectively. The relatively large error in this particular measurement is caused by the low X-ray pulse energy, which in turn generates timing signals with a rather small amplitude. Thus, the fitting function residuals are larger. In addition, we confirmed that without the Al attenuator the self-referenced signal strength increases by the same factor of ten as the X-ray pulse intensity, thus, confirming operation in the almost linear regime as described in Section 6.2.3.

We believe that the self-referenced measurement should deliver a more precise arrival-time information for a pump-probe experiment since it operates directly on the sample, thus, also taking into account small fluctuations of the sample position. This could be demonstrated in an experiment, where the selected sample develops a very steep fs response upon photo-excitation. Correcting the measured time trace for every timing point from each timing-tool would then deliver a signal with possibly different rise times. The timing-tool yielding the shorter rise time would then quantify its increased accuracy. Al-

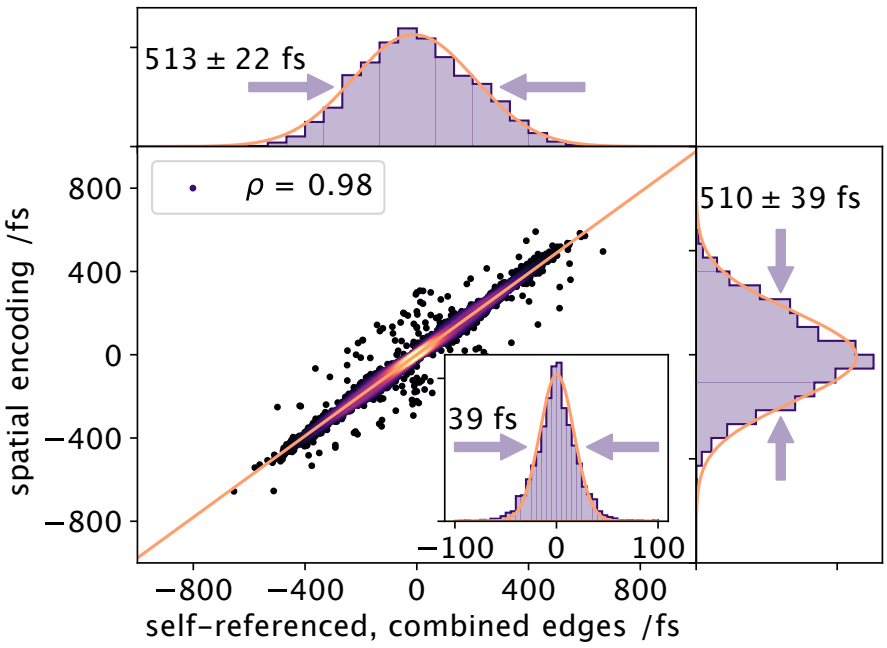

Figure 8.9.: Fit precision of the important error function parameter μ. The horizontal distance of the experimental (blue) data points are measured to the fit function (orange) for the two self-referenced edges a) and b) and the SACLA timing-tool c). The corresponding measured relative distances are shown in a'), b') and c'). The horizontal line indicates the mean value.

though such a study could not be performed in the current experimental campaign, we first confirmed its utility for this situation as a stand-alone timing-tool at the sample position. We guided an additional intense ($\approx$30 J/cm^2 peak-fluence) 400 nm optical beam onto the liquid jet together with the X-ray pulses to replicate a traditional experiment to investigate ultrafast dynamics. In this arrangement, the X-ray pulse, which is used as a pump pulse for the arrival-time measurement, simultaneously would serve as the probe pulse during a liquid chemistry experiment, where the dynamics in the sample are initiated with the additional intense 400 nm pulse. The X-ray arrival-time monitor then needs to be operational even in the presence of that additional intense 400 nm pump pulse, which could potentially distort the arrival-time measurement.

To investigate the influence of the additional 400 nm stimulus, we performed another arrival-time measurement with fixed timing between the X-ray and self-referenced timing-tool pulse, while scanning the additional intense pump pulse from $-$1,500 fs to 1,000 fs with the dedicated mechanical delay stage. As before, 5000 X-ray shots were recorded with the self-referenced detection scheme, shown in Fig. 8.10 a). For each time-delay setting, 200 X-ray arrival-times were recorded. The X-ray-induced arrival-time regions are framed in orange (leading edges) and purple (trailing edges). The intense 400 nm pulses act similar to the X-ray pulses, such that each pulse creates its own distinct plateau shape in the supercontinuum pulse used for the self-referenced timing-tool, due to the altered transient refractive index of the sample. The leading (pink) and trailing (gray-blue)

edges of the 400 nm stimulus timing signal are indicated in a) for each single pulse. Since the 400 nm and supercontinuum pulses used for the timing-tool, are generated from the same source, they are practically jitter-free, thus, the recorded timing jitter of the 400 nm induced timing signal should be very small. In consequence, the X-ray-induced and optically induced timing signals can overlap during a time-delay scan, which is most apparent when their respective edges cross each other, i.e., when both stimulating pulses arrive simultaneously.

In Fig. 8.10 a), three distinct delay times for the 400 nm pump beam are indicated with white boxes. At each of these delay settings, either the trailing edge, both edges, or the leading edge of the X-ray-induced timing signal overlaps with one or both edges of the optically induced timing signal. This initially leads to increased uncertainty in the determination of those X-ray arrival-times, shown in the right panels b) - g) in Fig. 8.10. The distributions in the right panel show, similar to the inset in Fig. 8.9, the residual timing jitter between the self-referenced timing-tool and the SACLA timing-tool. We found the accuracy limited to about twice the individual uncertainty of approximately 20 fs, however, further analysis can mitigate this increased uncertainty, either by simply selecting the unaffected edge to analyze or by adding more detailed knowledge into the edge shape to distinguish the 400 nm stimulus timing signal from the X-ray timing signal, when both are overlapped. The latter is only absolutely required at the point in time where the X-ray and the 400 nm pulses arrive simultaneously. In all other cases where the X-ray pulse and 400 nm stimulus pulse are separated by more than $\approx$20 fs, we always find an undisturbed edge for a precise arrival-time determination, as shown in Fig. 8.10 d) and e).

We analyzed the arrival-times of the 400 nm optical pulses for each delay in a similar way we analyzed the X-ray pulses, with the resulting edge positions indicated by the dots in Fig. 8.10 a). The edge position temporal jitter is shown in Fig 8.10 h) for all leading (pink) and trailing (gray-blue) edges within the single delays. When the 400 nm stimulus is set to arrive approximately 1 ps earlier or later in time compared to the X-ray pulses, the arrival-time jitter of the 400 nm pulses is 7 fs FWHM and by this similar to the calibration error without X-ray pulses. As the relative time-delays between X-ray and 400 nm stimulus approach each other, it becomes more and more challenging to disentangle the respective timing signals, which leads to increased uncertainty in measuring the X-ray pulse arrival-times as described before, but also in measuring the 400 nm stimulus arrival signals. The maximal uncertainty, when both pulses are exactly overlapped in time, is approximately 45 fs FWHM. A double Gaussian function was fitted to the data in Fig. 8.10 h) with the two peaks centered at the mean edge positions of the leading and trailing self-referenced X-ray-induced timing edges. The widths of the two Gaussians are fixed to the experimentally measured FWHM arrival-time jitter of the X-ray pulses of 513 fs.

In addition, one can even use the 400 nm stimulus timing signal imprinted in the X-ray arrival-time measurement to obtain the actual detector pixel to time mapping. This could

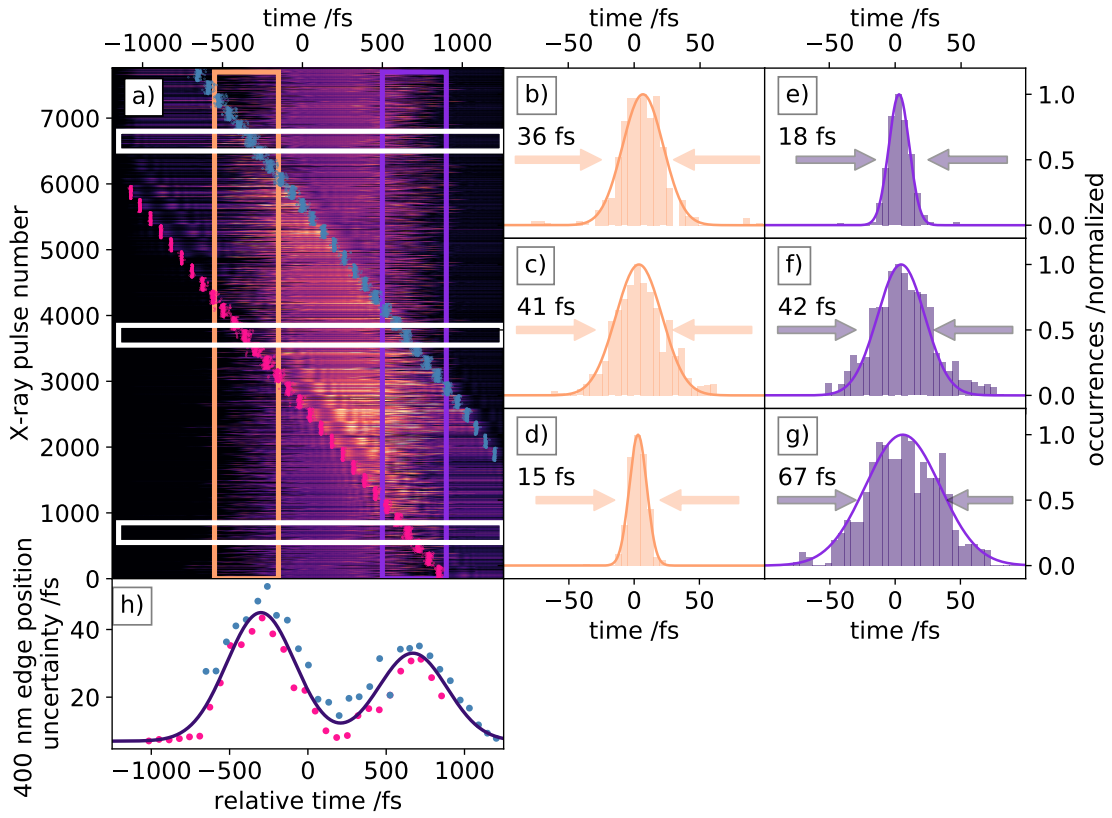

Figure 8.10.: Raw data (a) of the self-referenced timing-tool with an additional intense 400 nm stimulus for an actual liquid chemistry experiment. The 400 nm pump pulses were delayed from -1500 fs to $+1000$ fs with respect to the X-ray arrival-times. The additional timing signals induced by the 400 nm stimulus are indicated by the pink (leading edges) and gray-blue (trailing edges) dots. The X-ray-induced leading and trailing edge regions are indicated by the orange and purple framed area. Residual timing jitter between the leading (b-d) and trailing (e-g) X-ray-induced self-referenced edges and the SACLA spatial encoding timing-tool at selected time-delays. The selected delays are indicated by the white boxes. For b) and e) shots 6600-6800, for c) and f) shots 3600-3800, and for d) and g) shots 600-800 were analyzed. The precision with which the 400 nm induced leading (red) and trailing (green) timing edges can be determined is shown in h).

potentially be useful in certain measurement campaigns, where data is accumulated over a long time and delay scans are repeated to gain more statistics. In such a case, a change in the chirp of the supercontinuum could potentially be recognized during the measurement, and the time calibration could be adjusted accordingly without the need to abort the actual experiment.

Another notable feature of the recorded self-referenced pattern of the supercontinuum is that it contains in-situ information about the actual flat-sheet thickness of the free-flowing liquid jet. This information is obtained within the ps-short time-span of the supercontinuum passing through the flat-sheet jet. Close to the leading and trailing edges, additional oscillations caused by thin-film interferences are observed (e.g. Fig. 8.4 a), which we used to extract the liquid jet thickness. The thickness of the liquid sheet can be extracted by the analysis of the observed interference pattern near the sharp edges, using the Swanepoel method [197]. Counting the clearly visible fringes near the self-referenced edge positions, the jet thickness can be calculated with

$$d = \frac{(m_i - m_j)\lambda_i\lambda_j}{2(n_i\lambda_i - n_j\lambda_j)} \,, \tag{8.3}$$

where $(m_i - m_j)$ is the number of fringes counted between the two wavelengths $\lambda_{i,j}$ with the associated refractive index $n_{i,j}$ at those wavelengths. The retrieved jet thickness for each individual X-ray pulse is shown in Fig. 8.11. In contrast to the jet thickness measurement with the commercial device discussed in Fig. 8.2, the jet thicknesses retrieved directly from the experimental data indicate a jet thickness of (14 ± 2.4) µm. This is in close agreement with the (10 ± 1.1) µm extracted from the commercial thickness sensor, although measured independently under slightly different conditions. The latter measurement had been carried out during the experiment's setup phase approximately 24 hours before the timing-tool studies commenced. Its smaller mean value indicates changed conditions overnight, which also underlines the need to have as much in-situ information as possible during such an experiment and yields an average value over a much longer measurement period of 100 ms. The higher RMS fluctuation of the jet thickness retrieved from the experimental data can be explained by the fact that this measurement represents the jet thickness measured within a picosecond, while the commercial device needs 100 ms for a single measurement, thus, averaging the jet thickness over this time-span. Since the interference pattern used for the thickness measurement is taken exactly during the pump-probe shot and at the very same lateral position as the time-resolved laser-pump/X-ray-probe measurement, the presented data Fig. 8.11 represents hereby the most accurate values for the real liquid jet thickness during such a pump-probe study. This information can be useful in additional a posteriori corrections of the longitudinal dimension of the interacting sample volume.

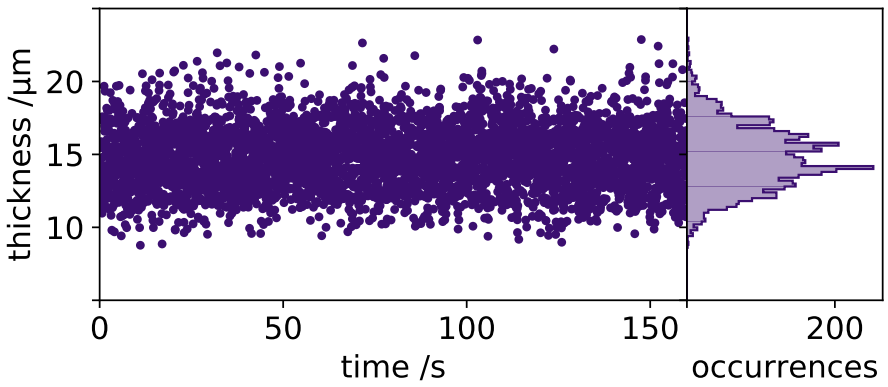

Figure 8.11.: Thickness measurement of the colliding liquid jet, using the self-referenced experimental data.

8.5. Conclusion

We have implemented a novel scheme to measure the relative arrival-time between hard X-ray pulses from an FEL and optical laser pulses directly on the sample that was simultaneously under investigation in a time-resolved X-ray pump/optical probe experiment. We determined a timing accuracy of 22 fs at worst, i.e., for the lowest X-ray pulse energies. This is sufficient to reliably correct the different arrival-times in comparison to the inherent timing jitter between the FEL and optical laser pulses at the SACLA facility under the circumstances of our experiment. The correlation with the existing SACLA timing-tool is excellent, while both methods yield arrival-time distributions (timing jitter) of around 500 fs FWHM, even in a measurement where we attenuated the X-ray pulse intensities by a factor of ten. The small difference in the observed arrival-time distributions can be explained by additional jitter sources between both optical laser paths to the two arrival-time monitors, which are located a few meters apart from each other. Therefore, we expect a more reliable timing precision when measured directly on the sample of interest as in our case. In a second measurement, it was demonstrated that an additional intense optical laser pulse can serve as a sample stimulus in a liquid-phase structural dynamics experiment, while the self-referenced timing-tool approach still delivers accurate in-situ arrival-time information. A full description of the involved electron cascading and propagation of the optical pulses and details of the estimated change of refractive index upon X-ray and potentially simultaneous intense optical irradiance is subject of a future study. In order to evaluate the ultimate performance of our timing-tool, one would need to compare the corrected timing data in a time-delay scan with a time-dependent X-ray-probe signal with all timing-tools available. In an experiment, based on photo-emission of an electron, one would measure a sharp rise after time zero, which would allow to compare the arrival-time-corrected rise times with each tool individually, with the goal to identify the superior arrival-time monitoring by the shorter rise time. Furthermore, such a study would then yield detailed data about the remaining timing jitter for timing

diagnostics located further away from the actual experiment. Finally, the analysis of the interference pattern in our arrival-time signal additionally allows to derive the thickness of the liquid sheet jet for every laser pulse, potentially allowing for additional correction of a pump-probe signal due to thickness fluctuations in case of sample delivery by a liquid jet. The utilization of a free-flowing liquid jet for in-situ arrival-time measurement promises applicability in liquid chemistry experiments at MHz repetition rate XFEL facilities.

9. Conclusion and Outlook

This chapter summarizes the main results of the thesis and presents an outlook for the future perspective of the newly developed timing-tool detection scheme.

9.1. Conclusion

A new detection scheme to measure the relative arrival-time at hard XFEL facilities was presented in this thesis. The need for a new detection scheme was justified with the new XFEL facilities, delivering intense X-ray pulses at MHz repetition rates. The deposited heat by the X-ray pulses in the timing-tool sample was simulated in Section 6.1. It was shown that commonly used timing-tool samples (Si_3N_4, YAG) could not withstand a fully filled X-ray pulse train at EuXFEL. It was also shown that diamond is the most suitable timing-tool sample. Due to its extraordinary high thermal conductivity and low X-ray absorption cross-section, it can withstand a fully filled pulse train at EuXFEL with moderate X-ray pulse intensities.
A full theoretical description of the expected self-referenced timing-tool signal was discussed in Chapter 6.2. The number of generated free electrons by an X-ray laser pulse was described and simulated for various X-ray photon energies and pulse energies. With the resulting free electron density, the expected transient refractive index change in diamond was calculated with the Drude model and the Maxwell Garnett model of dielectric mixtures. The expected refractive index change using the Drude model is more than one order of magnitude larger than calculated with the Maxwell Garnett model. Using these calculated X-ray-induced refractive index changes, the self-referenced timing-tool signal was simulated by propagating a laser pulse through the entire setup, taking dispersive effects into account up to the 3rd order. Experimental aspects of the alignment of the two BCs, the amount of the X-ray-induced transient refractive index change, and their influence on the emerging self-referenced timing-tool signal were simulated and discussed.
The MHz compatibility of the self-referenced timing-tool was reported in Chapter 7. It was briefly demonstrated that no X-ray arrival-time signal could be measured in diamond using the ordinary spectral encoding detection scheme. Using the same diamond sample, X-ray arrival-times were successfully measured with the self-referenced detection scheme. As expected, it was confirmed that the RFS at EuXFEL results in a larger X-ray arrival-time jitter (161 fs FWHM) compared to the (90 fs FWHM) with the OS. The inter- and intra-train arrival-time jitter of the X-ray pulses were analyzed with the result that the major contribution of the arrival-time jitter at EuXFEL is due to the 10 Hz

inter-train timing jitter. The analysis of the intra-train X-ray pulse arrival-times revealed a nearly identical intra-train timing jitter with both synchronization schemes (RFS 21 fs FWHM and OS 23 fs FWHM). Within the pulse trains, different emerging drift patterns of the X-ray arrival-times were identified for either one of the used synchronization schemes. The arrival-times of subsequent pulses within a pulse train with the RFS are linearly drifting to earlier arrival-times, while the X-ray arrival-times using the OS are drifting to later arrival-times. For the OS the drifting pattern suggests that the arrival-time drift is reversed after the first 75 pulses of the pulse train, potentially limiting the overall intra-train timing drift with the OS compared to the RFS. With the self-referenced detection scheme, it was also possible to determine the scale of the X-ray-induced refractive index change in the diamond sample. The refractive index change was found to be $\Delta n = -5.7 \times 10^{-5}$ and correlates very well with the calculated expected refractive index change of the Maxwell Garnett model. Furthermore, the minimal required X-ray-induced refractive index change to still measure a self-referenced arrival-time signal was estimated to be $\Delta n = -4.3 \times 10^{-6}$.

The results of a beamtime at SACLA are discussed and analyzed in Chapter 8. With this beamtime, it was demonstrated that the self-referenced timing tool could be used as an in-situ timing-tool, using the actual scientifically relevant sample to measure the relative X-ray arrival-times. A flat-sheet liquid jet system was used to deliver the liquid sample. The flow speed of the liquid jet system was analyzed using a high-speed camera. The flow speed of the liquid jet was measured to be fast enough to replenish the sample volume reliably, even at MHz XFEL facilities. In addition to the self-referenced timing-tool, the SACLA standard spatial encoding timing-tool was used to verify and correlate the measurements. The relative arrival-time jitter of the self-referenced timing tool (513 fs FWHM) and spatial encoding timing-tool (510 fs FWHM) correlate almost perfectly with a Pearson correlation of $\rho = 0.98$. A mean residual timing jitter of 39 fs between both timing-tools was observed. This residual timing jitter was attributed to spatial fluctuation of the flat-sheet jet. Thus, the in-situ self-referenced timing-tool approach might be beneficial to increase the achievable time-resolution. In a second measurement, it was shown that an additional intense laser pulse used for the sample excitation in a pump-probe experiment, complicates the analysis of the in-situ arrival-time measurement in the same sample. The additional intense laser pulse induces an additional plateau in the arrival-time signal, which needs to be disentangled from the X-ray-induced arrival-time signal in the same measurement. In addition, we demonstrated that the self-referenced timing-tool signal can even be used to get in-situ information about the thickness of the flat-sheet liquid jet at the moment of the actual experimental measurement.

9.2. Outlook

The reliable operation and accurate detection of X-ray pulse arrival-times using the newly developed self-referenced detection scheme was demonstrated in this thesis. It was shown that with this detection method, diamond can be used for the first time as a reliable interaction sample. Although the operation was demonstrated with diamond and a liquid water jet at 9.1 keV and 5.1 keV, further X-ray photon energies need to be tested. Especially higher photon energies in the region of 20 keV are critical since the X-ray absorption cross-section is smaller at higher energies. Thus, the X-ray-induced refractive index change is smaller at constant pulse energies. Many experiments use an X-ray monochromator to reduce the spectral bandwidth of the X-ray pulses. This reduces the X-ray pulse energy drastically, such that careful studies with an inserted monochromator in front of the self-referenced timing-tool setup are required. The results from Chapter 7.6 suggest that the photon flux behind a 4-bounce monochromator would still be sufficient to be able to operate the timing-tool, but experimental confirmation is needed.
Until now, EuXFEL is only running with a limited number of X-ray pulses in each pulse train. During the time of the measurements, the pulses were limited to 150 X-ray pulses out of the 2700 X-ray pulses possible. Further investigations are needed to really clarify if the diamond sample can withstand the deposited heat load of a fully filled pulse train. With more pulses injected into the pulse trains, further machine studies can be done to evaluate the effect of different injector, accelerator and undulator settings on the observed intra-train arrival-time drift patterns with the RFS and OS.
To establish the self-referenced detection scheme as an in-situ timing diagnostic for liquid chemical experiments, a full experimental demonstration is needed. With the SACLA beamtime and the results reported in Chapter 8, the foundation to realize a true in-situ timing-tool was founded. However, the advantages of the in-situ approach need to be shown in an actual ultrafast experiment where the experimental results of an ultrafast experiment are temporally sorted with the in-situ timing-tool and a remote identical timing-tool. Hopefully, the time-sorted experimental data with the in-situ timing-tool reveal sharper ultrafast features of the sample dynamics than the re-binned data using the identical remote timing-tool.
In this thesis, the relative X-ray arrival-times were spectrally encoded and detected with the self-referenced detection scheme. The self-referenced detection scheme can potentially also be used with other X-ray arrival-time encoding processes, such as spatially encoded X-ray arrival-times. This was recently demonstrated by Droste et al. [198] at LCLS. In addition, Tao et al. explore the possible usage of the self-referenced detection scheme to enhance the time-resolution in time-of-flight positron emission tomography measurements [199, 200].
An open question needing further investigations is the exact interaction of high-energy X-ray photons with the sample material and the subsequent X-ray-induced transient refractive index change. At 9.1 keV and an X-ray fluence of 12 J/cm^2, an X-ray-induced re-

fractive index change of Δn=-5.7$\times 10^{-5}$ could be measured. The calculated X-ray-induced refractive index change using the Maxwell Garnett model predicts a refractive index change of Δn=-5.9$\times 10^{-5}$ and is well within the experimental error bars. However, the error bars are rather large due to the described X-ray pointing and pulse energy fluctuations. To further investigate the predictive power of the Maxwell Garnett model for these calculations, the measurement should be repeated under more stable conditions and using different X-ray photon energies. A precise understanding of X-ray-induced refractive index changes could lead to further insights. Especially the emerging field of non-linear X-ray optics where, e.g., X-ray photons are parametric down-converted into an X-ray / optical photon pulse pair could benefit from these investigations.

Bibliography

[1] Albert Einstein. Zur Quantentheorie der Strahlung. *Phys. Z.*, 18:121–128, 1917.

[2] T. H. Maiman. Stimulated Optical Radiation in Ruby. *Nature*, 187(4736):493–494, August 1960.

[3] B. P. Abbott, R. Abbott, T. D. Abbott, et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *Physical Review Letters*, 116(6), February 2016.

[4] W. C. Röntgen. Ueber eine neue Art von Strahlen. *Nature*, 53(1369):274–276, January 1896.

[5] F. R. Elder, A. M. Gurewitsch, R. V. Langmuir, and H. C. Pollock. Radiation from Electrons in a Synchrotron. *Physical Review*, 71(11):829–830, June 1947.

[6] D. K. Bowen. INSTRUMENTATION AND EXPERIMENTS ON THE DARESBURY u. k. STORAGE RING. 342(1 Ultrasoft X-R):22–34, June 1980.

[7] R. Haensel. The European Synchrotron Radiation Facility ESRF In Grenoble. In *Seventh Conference Real Time '91 on Computer Applications in Nuclear, Particle and Plasma Physics Conference Record*, pages 2–6, 1991.

[8] Ahmed H. Zewail. Femtochemistry: Atomic-Scale Dynamics of the Chemical Bond. 104(24):5660–5694, May 2000.

[9] R. W. Schoenlein, S. Chattopadhyay, H. H. W. Chong, et al. Generation of Femtosecond Pulses of Synchrotron Radiation. *Science*, 287(5461):2237–2240, 2000.

[10] V. Ayvazyan, N. Baboi, J. Bähr, et al. First operation of a free-electron laser generating GW power radiation at 32 nm wavelength. 37(2):297–303, November 2005.

[11] P. Emma, R. Akre, J. Arthur, et al. First lasing and operation of an ångstrom-wavelength free-electron laser. *Nature Photonics*, 4(9):641–647, 2010.

[12] James M. Glownia, J. Cryan, J. Andreasson, et al. Time-resolved pump-probe experiments at the LCLS. *Optics Express*, 18(17):17620, August 2010.

[13] Henry N. Chapman, Petra Fromme, Anton Barty, et al. Femtosecond X-ray protein nanocrystallography. *Nature*, 470(7332):73–77, 2011.

[14] Wenkai Zhang, Roberto Alonso-Mori, Uwe Bergmann, et al. Tracking excited-state charge and spin dynamics in iron coordination complexes. *Nature*, 509(7500):345–348, 2014.

[15] Kyung Hwan Kim, Jong Goo Kim, Shunsuke Nozawa, et al. Direct observation of bond formation in solution with femtosecond X-ray scattering. *Nature*, 518(7539):385–389, February 2015.

[16] Henrik T. Lemke, Kasper S. Kjær, Robert Hartsock, et al. Coherent structural trapping through wave packet dispersion during photoinduced spin state switching. *Nature Communications*, 8(1):15342, 2017.

[17] S Serkez, G Geloni, S Tomin, et al. Overview of options for generating high-brightness attosecond x-ray pulses at free-electron lasers and applications at the European XFEL. *Journal of Optics*, 20(2):024005, jan 2018.

[18] M. Th. Hassan, T. T. Luu, A. Moulet, et al. Optical attosecond pulses and tracking the nonlinear response of bound electrons. *Nature*, 530(7588):66–70, February 2016.

[19] Jungwon Kim, Jonathan A. Cox, Jian Chen, and Franz X. Kärtner. Drift-free femtosecond timing synchronization of remote optical and microwave sources. *Nature Photonics*, 2(12):733–736, 2008.

[20] S. Schulz, I. Grguraš, C. Behrens, et al. Femtosecond all-optical synchronization of an X-ray free-electron laser. *Nature Communications*, 6(1):5938, May 2015.

[21] Ming Xin, Kemal Şafak, and Franz X. Kärtner. Ultra-precise timing and synchronization for large-scale scientific instruments. *Optica*, 5(12):1564, December 2018.

[22] M. Altarelli. The European X-ray free-electron laser facility in Hamburg. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 269(24):2845 – 2849, 2011. Proceedings of the 10th European Conference on Accelerators in Applied Research and Technology (ECAART10).

[23] W. Decking, S. Abeghyan, P. Abramian, et al. A MHz-repetition-rate hard X-ray free-electron laser driven by a superconducting linear accelerator. *Nature Photonics*, 14(6):391–397, June 2020.

[24] R. Abela, A. Aghababyan, M. Altarelli, et al. *XFEL: The European X-Ray Free-Electron Laser - Technical Design Report*. DESY, Hamburg, 2006.

[25] Harald Sinn. Heat load estimates for XFEL beamline optics. *Hasylab Annual Report*, 2007.

[26] Shan Liu, Winfried Decking, Vitali Kocharyan, et al. Preparing for high-repetition rate hard x-ray self-seeding at the European X-ray Free Electron Laser: Challenges and opportunities. *Phys. Rev. Accel. Beams*, 22:060704, Jun 2019.

[27] Tetsuya Ishikawa, Hideki Aoyagi, Takao Asaka, et al. A compact X-ray free-electron laser emitting in the sub-ångström region. *Nature Photonics*, 6(8):540–544, 2012.

[28] Heung-Sik Kang, Chang-Ki Min, Hoon Heo, et al. Hard x-ray free-electron laser with femtosecond-scale timing jitter. *Nature Photonics*, 11(11):708–713, 2017.

[29] Christopher Milne, Thomas Schietinger, Masamitsu Aiba, et al. SwissFEL: The Swiss X-ray Free Electron Laser. *Applied Sciences*, 7(7):720, Jul 2017.

[30] Igor Zagorodnov, Martin Dohlus, and Sergey Tomin. Accelerator beam dynamics at the European X-ray Free Electron Laser. *Physical Review Accelerators and Beams*, 22(2):024401, February 2019.

[31] Brian W. J. McNeil and Neil R. Thompson. X-ray free-electron lasers. *Nature Photonics*, 4(12):814–821, 2010.

[32] E.L. Saldin, E.A. Schneidmiller, and M.V. Yurkov. The physics of free electron lasers. An introduction. *Physics Reports*, 260(4):187 – 327, 1995.

[33] Chao Lu, Tao Jiang, Shengguang Liu, et al. Coulomb-Driven Relativistic Electron Beam Compression. *Phys. Rev. Lett.*, 120:044801, Jan 2018.

[34] E. Harms, H. Edwards, T. Arkan, et al. THIRD HARMONIC SYSTEM AT FERMILAB*/FLASH. 2009.

[35] M. Hamberg et al. Electron Beam Heating with the European XFEL Laser Heater. In *Proc. of International Free Electron Laser Conference (FEL'17), Santa Fe, NM, USA, August 20-25, 2017*, number 38 in International Free Electron Laser Conference, pages 458–459, Geneva, Switzerland, Feb. 2018. JACoW. https://doi.org/10.18429/JACoW-FEL2017-WEP018.

[36] T. Limberg, V. Balandin, R. Brinkmann, et al. Optimized Bunch Compression System for the European XFEL. In *Proceedings of the 2005 Particle Accelerator Conference*, pages 1236–1238, 2005.

[37] B. Aune, R. Bandelmann, D. Bloess, et al. Superconducting TESLA cavities. *Phys. Rev. ST Accel. Beams*, 3:092001, Sep 2000.

[38] E. A. Schneidmiller, M. V. Yurkov, and DESY. Photon beam properties at the European XFEL. *Technical Report*, 2011.

[39] R. Bonifacio, C. Pellegrini, and L.M. Narducci. Collective instabilities and high-gain regime in a free electron laser. *Optics Communications*, 50(6):373 – 378, 1984.

[40] G Geloni, E Saldin, L Samoylova, et al. Coherence properties of the european XFEL. *New Journal of Physics*, 12(3):035021, mar 2010.

[41] Yujong Kim. Start-To-End Simulations for PAL XFEL Project. 2004.

[42] Max J. Lederer, Mikhail Pergament, Martin Kellert, and Cruz Mendez. Pump-probe laser development for the European X-ray Free-Electron Laser facility. In Stefan P. Moeller, Makina Yabashi, and Stefan P. Hau-Riege, editors, *X-Ray Free-Electron Lasers: Beam Diagnostics, Beamline Instrumentation, and Applications*, volume 8504, pages 106 – 111. International Society for Optics and Photonics, SPIE, 2012.

[43] Michael P. Minitti, Joseph S. Robinson, Ryan N. Coffee, et al. Optical laser systems at the Linac Coherent Light Source. *Journal of Synchrotron Radiation*, 22(3):526–531, May 2015.

[44] R. Riedel, M. Schulz, M. J. Prandolini, et al. Long-term stabilization of high power optical parametric chirped-pulse amplifiers. *Opt. Express*, 21(23):28987–28999, Nov 2013.

[45] P. Cinquegrana, S. Cleva, A. Demidovich, et al. Optical beam transport to a remote location for low jitter pump-probe experiments with a free electron laser. *Phys. Rev. ST Accel. Beams*, 17:040702, Apr 2014.

[46] K. Safak, M. Xin, P. T. Callahan, et al. All fiber-coupled, long-term stable timing distribution for free-electron lasers with few-femtosecond jitter. *Structural Dynamics*, 2(4):041715, 2015.

[47] Donna Strickland and Gerard Mourou. Compression of amplified chirped optical pulses. *Optics Communications*, 56(3):219 – 221, 1985.

[48] A. Dubietis, G. Jonušauskas, and A. Piskarskas. Powerful femtosecond pulse generation by chirped and stretched pulse parametric amplification in BBO crystal. *Optics Communications*, 88(4):437 – 440, 1992.

[49] Taisuke Miura, Katsuyuki Kobayashi, Kazuya Takasago, et al. Timing jitter in a kilohertz regenerative amplifier of a femtosecond-pulse Ti:Al2O3 laser. *Opt. Lett.*, 25(24):1795–1797, Dec 2000.

[50] Sandro Klingebiel, Izhar Ahmad, Christoph Wandt, et al. Experimental and theoretical investigation of timing jitter inside a stretcher-compressor setup. *Opt. Express*, 20(4):3443–3455, Feb 2012.

[51] Thorsten Lamb, Marie Czwalinna, Matthias Felber, et al. Large-Scale Optical Synchronization System of the European XFEL with Femtosecond Precision. pages 3835–3838. JACOW Publishing, Geneva, Switzerland, June 2019.

[52] Bartosz Gąsowski, Tomasz Owczarek, Krzysztof Czuba, et al. Real-Time Redundancy for the 1.3 GHz Master Oscillator of the European-XFEL, 2018.

[53] K. Czuba and D. Sikora. Temperature Stability of Coaxial Cables. *Acta Physica Polonica A*, 119(4):553–557, April 2011.

[54] T. R. Schibli, J. Kim, O. Kuzucu, et al. Attosecond active synchronization of passively mode-locked lasers by balanced cross correlation. In *Advanced Solid-State Photonics*, page 108. Optical Society of America, 2003.

[55] T. Lamb, M. Felber, T. Kozak, et al. Femtosecond Laser-to-RF Synchronization and RF Reference Distribution at the European XFEL. In *Proc. FEL'19*, number 39 in Free Electron Laser Conference, pages 343–345. JACoW Publishing, Geneva, Switzerland, nov 2019. https://doi.org/10.18429/JACoW-FEL2019-WEP010.

[56] J. Kim, F. X. Kärtner, and M. H. Perrott. Femtosecond synchronization of radio frequency signals with optical pulse trains. *Opt. Lett.*, 29(17):2076–2078, Sep 2004.

[57] Chris J.H. Wort and Richard S. Balmer. Diamond as an electronic material. *Materials Today*, 11(1):22 – 28, 2008.

[58] Richard P. Mildren, editor. *Optical Engineering of Diamond*. Wiley-VCH, 2013.

[59] Nianjun Yang, editor. *Novel aspects of diamond from growth to applications*. Topics in applied physics volume 121. Springer, second edition edition, 2019.

[60] J M Blocher, Jr and J H Oxley. Chemical vapor deposition opens new horizons in ceramic technology. *American Ceramic Society Bulletin (U.S.)*, Vol: 41, 2 1962.

[61] Christoph E. Nebel. Chapter Four - Nitrogen-vacancy doped CVD diamond for quantum applications: A review. In Christoph E. Nebel, Igor Aharonovich, Norikazu Mizuochi, and Mutsuko Hatano, editors, *Diamond for Quantum Applications Part 1*, volume 103 of *Semiconductors and Semimetals*, pages 73 – 136. Elsevier, 2020.

[62] A. Stacey, T. J. Karle, L. P. McGuinness, et al. Depletion of nitrogen-vacancy color centers in diamond via hydrogen passivation. *Applied Physics Letters*, 100(7):071902, 2012.

[63] J. E. Field. The Properties of Natural and Synthetic Diamond. *Crystal Research and Technology*, 28(5):602–602, 1993.

[64] Stephen John Sque. *A First-Principles Study on Bulk and Transfer Doping of Diamond*. PhD thesis, University of Exeter, England, 2005.

[65] F. Bloch. *Über die Quantenmechanik der Elektronen in Kristallgittern*. Springer, 1928.

[66] M. Dresselhaus, G. Dresselhaus, S.B. Cronin, and A.G.S. Filho. *Solid State Properties: From Bulk to Nano*. Graduate Texts in Physics. Springer Berlin Heidelberg, 2018.

[67] I. Friel, S.L. Clewes, H.K. Dhillon, et al. Control of surface and bulk crystalline quality in single crystal diamond grown by chemical vapour deposition. *Diamond and Related Materials*, 18(5):808–815, 2009. Proceedings of Diamond 2008, the 19th European Conference on Diamond, Diamond-Like Materials, Carbon Nanotubes, Nitrides and Silicon Carbide.

[68] J. Als-Nielsen and D. McMorrow. *Elements of Modern X-ray Physics*. Wiley, 2011.

[69] Gavin B. M. Vaughan, Robert Baker, Raymond Barret, et al. ID15A at the ESRF - a beamline for high speed operando X-ray diffraction, diffraction tomography and total scattering. *Journal of Synchrotron Radiation*, 27(2):515–528, Mar 2020.

[70] M.J. Berger, S.M. Seltzer, J. Chang, et al. XCOM: Photon Cross Sections Database. 2020.

[71] P.A. Tipler and G. Mosca. *Physics for Scientists and Engineers*. Physics for Scientists and Engineers: Standard. W. H. Freeman, 2007.

[72] Olavi Keski-Rahkonen and Manfred O. Krause. Total and partial atomic-level widths. *Atomic Data and Nuclear Data Tables*, 14(2):139 – 146, 1974.

[73] M. O. Krause. Atomic radiative and radiationless yields for K and L shells. *Journal of Physical and Chemical Reference Data*, 8(2):307–327, 1979.

[74] Beata Ziaja, Richard A. London, and Janos Hajdu. Unified model of secondary electron cascades in diamond. *Journal of Applied Physics*, 97(6):064905, March 2005.

[75] A. J. Tuzzolino. Quantum Efficiency of Silicon in the Vacuum Ultraviolet. *Phys. Rev.*, 134:A205–A213, Apr 1964.

[76] R. C. Alig and S. Bloom. Electron-Hole-Pair Creation Energies in Semiconductors. *Phys. Rev. Lett.*, 35:1522–1525, Dec 1975.

[77] A. Kaiser, B. Rethfeld, M. Vicanek, and G. Simon. Microscopic processes in dielectrics under irradiation by subpicosecond laser pulses. *Phys. Rev. B*, 61:11437–11450, May 2000.

[78] N. Medvedev and B. Rethfeld. Effective energy gap of semiconductors under irradiation with an ultrashort VUV laser pulse. *EPL (Europhysics Letters)*, 88(5):55001, nov 2009.

[79] Patrick Lorazo, Laurent J. Lewis, and Michel Meunier. Thermodynamic pathways to melting, ablation, and solidification in absorbing solids under pulsed laser irradiation. *Phys. Rev. B*, 73:134108, Apr 2006.

[80] Nikita A. Medvedev. X-ray-induced electron cascades in dielectrics modeled with XCASCADE code: effect of impact ionization cross sections. In Libor Juha, Saša

Bajt, and Richard London, editors, *Damage to VUV, EUV, and X-ray Optics V*, volume 9511, pages 55 – 62. International Society for Optics and Photonics, SPIE, 2015.

[81] Nikita Medvedev. Femtosecond X-ray induced electron kinetics in dielectrics: application for FEL-pulse-duration monitor. *Applied Physics B*, 118(3):417–429, March 2015.

[82] James Clerk Maxwell. A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155:459–513, 1865.

[83] James Clerk Maxwell. A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155:459–512, 1865.

[84] John David Jackson. *Classical electrodynamics; 2nd ed.* Wiley, New York, NY, 1975.

[85] P. Drude. Zur Elektronentheorie der Metalle. *Annalen der Physik*, 306(3):566–613, 1900.

[86] J. C. Maxwell Garnett and Joseph Larmor. Xii. Colours in metal glasses and in metallic films. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 203(359-371):385–420, 1904.

[87] J. C. Maxwell Garnett and Joseph Larmor. VII. Colours in metal glasses, in metallic films, and in metallic solutions. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 205(387-401):237–288, 1906.

[88] L. D. Landau. Über die Bewegung der Elektronen im Kristallgitter. *Phys. Z. Sowjetunion.*, 3(1):644–645, 1935.

[89] S. I. Pekar. Study on Electron Theory of Crystals. *Usp. Fiz. Nauk*, 48(3):447–451, 1951.

[90] Vadim A. Markel. Introduction to the Maxwell Garnett approximation: tutorial. *Journal of the Optical Society of America A*, 33(7):1244, June 2016.

[91] Max Born and Emil Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th Edition)*. Cambridge University Press, 7th edition, 1999.

[92] W. Nolting. *Grundkurs Theoretische Physik. 3. Elektrodynamik : mit 73 Aufgaben mit vollständigen Lösungen*. Grundkurs: Theoretische Physik. Springer, 2003.

[93] Bahaa E A Saleh and Malvin Carl Teich. *Fundamentals of photonics; 2nd ed.* Wiley series in pure and applied optics. Wiley, New York, NY, 2007.

[94] D. J. Bradley and G. H. C. New. Ultrashort pulse measurements. *Proceedings of the IEEE*, 62(3):313–345, March 1974.

[95] Roy J. Glauber. The Quantum Theory of Optical Coherence. *Phys. Rev.*, 130:2529–2539, Jun 1963.

[96] Willis E. Lamb. Theory of an Optical Maser. *Phys. Rev.*, 134:A1429–A1450, Jun 1964.

[97] L. E. Hargrove, R. L. Fork, and M. A. Pollack. LOCKING OF he–ne LASER MODES INDUCED BY SYNCHRONOUS INTRACAVITY MODULATION. *Applied Physics Letters*, 5(1):4–5, July 1964.

[98] W. Sellmeier. Ueber die durch die Aetherschwingungen erregten Mitschwingungen der Körpertheilchen und deren Rückwirkung auf die ersteren, besonders zur Erklärung der Dispersion und ihrer Anomalien. *Annalen der Physik*, 223(11):386–403, 1872.

[99] X. Yan, A. M. MacLeod, W. A. Gillespie, et al. Subpicosecond Electro-optic Measurement of Relativistic Electron Pulses. *Phys. Rev. Lett.*, 85:3404–3407, Oct 2000.

[100] J. A. Valdmanis, G. Mourou, and C. W. Gabel. Picosecond electro-optic sampling system. *Applied Physics Letters*, 41(3):211–212, 1982.

[101] Q. Wu and X.-C. Zhang. Free-space electro-optic sampling of terahertz beams. *Applied Physics Letters*, 67(24):3523–3525, 1995.

[102] A. L. Cavalieri, D. M. Fritz, S. H. Lee, et al. Clocking Femtosecond X-Rays. *Phys. Rev. Lett.*, 94:114801, Mar 2005.

[103] D.J. Bradley, B. Liddy, and W.E. Sleat. Direct linear measurement of ultrashort light pulses with a picosecond streak camera. *Optics Communications*, 2(8):391–395, 1971.

[104] Ulrike Frühling, Marek Wieland, Michael Gensch, et al. Single-shot terahertz-field-driven X-ray streak camera. *Nature Photonics*, 3(9):523–528, September 2009.

[105] I. Grguraš, A. R. Maier, C. Behrens, et al. Ultrafast X-ray pulse characterization at free-electron lasers. *Nature Photonics*, 6(12):852–857, December 2012.

[106] János Hebling, Ka-Lo Yeh, Matthias C. Hoffmann, et al. Generation of high-power terahertz pulses by tilted-pulse-front excitation and their application possibilities. *Journal of the Optical Society of America B*, 25(7):B6, April 2008.

[107] A. G. Stepanov, S. Henin, Y. Petit, et al. Mobile source of high-energy single-cycle terahertz pulses. *Applied Physics B*, 101(1-2):11–14, August 2010.

[108] P. N. Juranić, A. Stepanov, R. Ischebeck, et al. High-precision x-ray FEL pulse arrival time measurements at SACLA by a THz streak camera with Xe clusters. *Optics Express*, 22(24):30004, December 2014.

[109] Ishkhan Gorgisyan, Rasmus Ischebeck, Christian Erny, et al. THz streak camera method for synchronous arrival time measurement of two-color hard X-ray FEL pulses. *Opt. Express*, 25(3):2080–2091, Feb 2017.

[110] N. Hartmann, G. Hartmann, R. Heider, et al. Attosecond time–energy structure of X-ray free-electron laser pulses. *Nature Photonics*, 12(4), Mar 2018.

[111] R. Heider, M. S. Wagner, N. Hartmann, et al. Megahertz-compatible angular streaking with few-femtosecond resolution at x-ray free-electron lasers. *Phys. Rev. A*, 100:053420, Nov 2019.

[112] Cornelius Gahl, Armin Azima, Martin Beye, et al. A femtosecond X-ray/optical cross-correlator. *Nature Photonics*, 2(3):165–169, March 2008.

[113] Mina R. Bionta, H. T. Lemke, J. P. Cryan, et al. Spectral encoding of x-ray/optical relative delay. *Optics Express*, 19(22):21855, October 2011.

[114] M. Harmand, R. Coffee, M. R. Bionta, et al. Achieving few-femtosecond time-sorting at hard X-ray free-electron lasers. *Nature Photonics*, 7(3):215–218, March 2013.

[115] M. R. Bionta, N. Hartmann, M. Weaver, et al. Spectral encoding method for measuring the relative arrival time between x-ray/optical pulses. *Review of Scientific Instruments*, 85(8):083116, August 2014.

[116] M. Lorenc, M. Ziolek, R. Naskrecki, et al. Artifacts in femtosecond transient absorption spectroscopy. *Applied Physics B: Lasers and Optics*, 74(1):19–27, January 2002.

[117] James M. Glownia, Karl Gumerlock, Henrik T. Lemke, et al. Pump–probe experimental methodology at the Linac Coherent Light Source. *Journal of Synchrotron Radiation*, 26(3):685–691, May 2019.

[118] Makina Yabashi, Hitoshi Tanaka, and Tetsuya Ishikawa. Overview of the SACLA facility. *Journal of Synchrotron Radiation*, 22(3):477–484, 2015.

[119] S. Schorb, T. Gorkhover, J. P. Cryan, et al. X-ray–optical cross-correlator for gas-phase experiments at the Linac Coherent Light Source free-electron laser. *Applied Physics Letters*, 100(12):121107, March 2012.

[120] Takahiro Sato, Tadashi Togashi, Kanade Ogawa, et al. Highly efficient arrival timing diagnostics for femtosecond X-ray and optical laser pulses. *Applied Physics Express*, 8(1):012702, January 2015.

[121] Tetsuo Katayama, Shigeki Owada, Tadashi Togashi, et al. A beam branching method for timing and spectral characterization of hard X-ray free-electron lasers. *Structural Dynamics*, 3(3):034301, May 2016.

[122] R. Riedel, A. Al-Shemmary, M. Gensch, et al. Single-shot pulse duration monitor for extreme ultraviolet and X-ray free-electron lasers. *Nature Communications*, 4(1):1731, June 2013.

[123] Henry J. Kirkwood, Romain Letrun, Takanori Tanikawa, et al. Initial observations of the femtosecond timing jitter at the European XFEL. *Optics Letters*, 44(7):1650, April 2019.

[124] Tokushi Sato, Romain Letrun, Henry J. Kirkwood, et al. Femtosecond timing synchronization at megahertz repetition rates for an x-ray free-electron laser. *Optica*, 7(6):716, June 2020.

[125] Lawrence Mertz. Spectromètre stellaire multicanal. *Journal de Physique et le Radium*, 19(3):233–236, 1958.

[126] Michael F. A'Hearn, Francis J. Ahern, and David M. Zipoy. Polarization Fourier Spectrometer for Astronomy. *Applied Optics*, 13(5):1147, May 1974.

[127] Daniele Brida, Cristian Manzoni, and Giulio Cerullo. Phase-locked pulses for two-dimensional spectroscopy by a birefringent delay line. *Opt. Lett.*, 37(15):3027–3029, Aug 2012.

[128] Ian Robinson, Joerg Schwenke, Mohammed Yusuf, et al. Towards single particle imaging of human chromosomes at SACLA. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 48(24):244007, nov 2015.

[129] Ichiro Inoue, Yuichi Inubushi, Takahiro Sato, et al. Observation of femtosecond X-ray interactions with matter using an X-ray–X-ray pump–probe scheme. *Proceedings of the National Academy of Sciences*, 113(6):1492–1497, 2016.

[130] S. P. Hau-Riege, R. A. London, R. M. Bionta, et al. Damage threshold of inorganic solids under free-electron-laser irradiation at 32.5nm wavelength. *Applied Physics Letters*, 90(17):173128, 2007.

[131] Stefan P. Hau-Riege, Richard A. London, Richard M. Bionta, et al. Multiple pulse thermal damage thresholds of materials for x-ray free electron laser optics investigated with an ultraviolet laser. *Applied Physics Letters*, 93(20):201105, 2008.

[132] W. Demtröder. *Experimentalphysik 1: Mechanik und Wärme*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2017.

[133] B.L. Henke, E.M. Gullikson, and J.C. Davis. X-Ray Interactions: Photoabsorption, Scattering, Transmission, and Reflection at E = 50-30,000 eV, Z = 1-92. *Atomic Data and Nuclear Data Tables*, 54(2):181 – 342, 1993.

[134] Pol Torres Alvarez. *Phonon Spectrum and Transient Regimes in the KCM*, pages 117–135. Springer International Publishing, Cham, 2018.

[135] H.S. Carslaw and J.C. Jaeger. *Conduction of Heat in Solids*. Oxford science publications. Clarendon Press, 1986.

[136] John Crank. *The mathematics of diffusion*. Clarendon Press, Oxford, [Eng], 2d ed edition, 1975.

[137] L. Euler. *Institutionum calculi integralis*. Number v. 1 in Institutionum calculi integralis. imp. Acad. imp. Saènt., 1768.

[138] Hans Petter Langtangen and Svein Linge. *Finite Difference Computing with PDEs: A Modern Software Approach*, volume 16 of *Texts in Computational Science and Engineering*. Springer International Publishing, Cham, 2017.

[139] P. Jacobson and S. Stoupin. Thermal expansion coefficient of diamond in a wide temperature range. *Diamond and Related Materials*, 97:107469, 2019.

[140] Yasumasa Okada and Yozo Tokumaru. Precise determination of lattice parameter and thermal expansion coefficient of silicon between 300 and 1500 K. *Journal of Applied Physics*, 56(2):314–320, 1984.

[141] W. Paszkowicz, R. Minikayev, P. Piszora, et al. Thermal expansion of spinel-type si_3n_4. *Phys. Rev. B*, 69:052103, Feb 2004.

[142] Xiaodong Xu, Zhiwei Zhao, Honghua Wang, et al. Spectroscopic and thermal properties of Cr,Yb:YAG crystal. *Journal of Crystal Growth*, 262(1):317 – 321, 2004.

[143] Andrew C. Victor. Heat Capacity of Diamond at High Temperatures. *The Journal of Chemical Physics*, 36(7):1903–1911, 1962.

[144] M.W.J. Chase. *NIST-JANAF Thermochemical Tables*. Journal of Physical and Chemical Reference Data Monographs. American Inst. of Physics, 1998.

[145] R.J.M Konings, R.R van der Laan, A.C.G van Genderen, and J.C van Miltenburg. The heat capacity of Y3Al5O12 from 0 to 900K. *Thermochimica Acta*, 313(2):201 – 206, 1998.

[146] C. Howard Shomate. A Method for Evaluating and Correlating Thermodynamic Data. *The Journal of Physical Chemistry*, 58(4):368–372, 1954.

[147] P. Debye. Zur Theorie der spezifischen Wärmen. *Annalen der Physik*, 344(14):789–839, 1912.

[148] D.J. Twitchen, C.S.J. Pickles, S.E. Coe, et al. Thermal conductivity measurements on CVD diamond. *Diamond and Related Materials*, 10(3):731–735, 2001. 11th European Conference on Diamond, Diamond-like Materials, Carbon Nanotubes, Nitrides and Silicon Carbide.

[149] C. J. Glassbrenner and Glen A. Slack. Thermal Conductivity of Silicon and Germanium from 3 K to the Melting Point. *Phys. Rev.*, 134:A1058–A1069, May 1964.

[150] Naoto Hirosaki, Yusuke Okamoto, Motohide Ando, et al. Effect of Grain Growth on the Thermal Conductivity of Silicon Nitride. *Journal of the Ceramic Society of Japan*, 104(1205):49–53, 1996.

[151] Nitin P. Padture and Paul G. Klemens. Low Thermal Conductivity in Garnets. *Journal of the American Ceramic Society*, 80(4):1018–1020, 1997.

[152] The CVD diamond booklet. http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed: 2020-06-30.

[153] M. Gabrysch, E. Marklund, J. Hajdu, et al. Formation of secondary electron cascades in single-crystalline plasma-deposited diamond upon exposure to femtosecond x-ray pulses. *Journal of Applied Physics*, 103(6):064909, 2008.

[154] V V Kononenko, E V Zavedeev, M I Latushko, and V I Konov. Observation of fs laser-induced heat dissipation in diamond bulk. *Laser Physics Letters*, 10(3):036003, March 2013.

[155] B. Rethfeld, O. Brenk, N. Medvedev, et al. Interaction of dielectrics with femtosecond laser pulses: application of kinetic approach and multiple rate equation. *Applied Physics A*, 101(1):19–25, October 2010.

[156] Jan Isberg, Johan Hammersberg, Erik Johansson, et al. High Carrier Mobility in Single-Crystal Plasma-Deposited Diamond. *Science*, 297(5587):1670–1672, 2002.

[157] H. R. Phillip and E. A. Taft. Kramers-Kronig Analysis of Reflectance Data for Diamond. *Phys. Rev.*, 136:A1445–A1448, Nov 1964.

[158] Nobuko Naka, Katsuyuki Fukai, Yushi Handa, and Ikuko Akimoto. Direct measurement via cyclotron resonance of the carrier effective masses in pristine diamond. *Phys. Rev. B*, 88:035205, Jul 2013.

[159] A L Shluger and A M Stoneham. Small polarons in real crystals: concepts and problems. *Journal of Physics: Condensed Matter*, 5(19):3049–3086, may 1993.

[160] A N Brozdnichenko, D M Dolgintsev, and R A Castro. The dielectric properties of the diamond-like films grown by ion-plasma method. *Journal of Physics: Conference Series*, 572:012025, dec 2014.

[161] Toshimaro Sato, Kazutoshi Ohashi, Tomoko Sudoh, et al. Thermal expansion of a high purity synthetic diamond single crystal at low temperatures. *Phys. Rev. B*, 65:092102, Feb 2002.

[162] I. H. Malitson. Interspecimen Comparison of the Refractive Index of Fused Silica. *J. Opt. Soc. Am.*, 55(10):1205–1209, Oct 1965.

[163] D. Eimerl, L. Davis, S. Velsko, et al. Optical, mechanical, and thermal properties of barium borate. *Journal of Applied Physics*, 62(5):1968–1983, 1987.

[164] Giorgio Turri, Scott Webster, Ying Chen, et al. Index of refraction from the near-ultraviolet to the near-infrared from a single crystal microwave-assisted CVD diamond. *Opt. Mater. Express*, 7(3):855–859, Mar 2017.

[165] Ausra Cerkauskaite, Rokas Drevinskas, Alexey O. Rybaltovskii, and Peter G. Kazansky. Ultrafast laser-induced birefringence in various porosity silica glasses: from fused silica to aerogel. *Opt. Express*, 25(7):8011–8021, Apr 2017.

[166] Ulrich Neukirch, Douglas C. Allan, Nicholas F. Borrelli, et al. Laser-induced birefringence in fused silica from polarized lasers. In Bruce W. Smith, editor, *Optical Microlithography XVIII*, volume 5754, pages 638 – 645. International Society for Optics and Photonics, SPIE, 2005.

[167] Shilei Xiao, Bincheng Li, Hao Cui, and Jing Wang. Sensitive measurement of stress birefringence of fused silica substrates with cavity ring-down technique. *Opt. Lett.*, 43(4):843–846, Feb 2018.

[168] R. Clark Jones. A New Calculus for the Treatment of Optical SystemsI. Description and Discussion of the Calculus. *J. Opt. Soc. Am.*, 31(7):488–493, Jul 1941.

[169] H. Crew, C. Huygens, T. Young, et al. *The Wave Theory of Light: Memoirs of Huygens, Young and Fresnel*. Number v. 15 in PCMI collection. American Book Company, 1900.

[170] U. Megerle, I. Pugliesi, C. Schriever, et al. Sub-50 fs broadband absorption spectroscopy with tunable excitation: putting the analysis of ultrafast molecular dynamics on solid ground. *Applied Physics B*, 96(2-3):215–231, August 2009.

[171] Bruno Eugen Schmidt. *White light filamentation*. PhD thesis, Freie Universität Berlin, Freie Universität Berlin, Germany, 2008.

[172] Andreas Galler, Wojciech Gawelda, Mykola Biednov, et al. Scientific instrument Femtosecond X-ray Experiments (FXE): instrumentation and baseline experimental capabilities. *Journal of Synchrotron Radiation*, 26(5):1432–1447, 2019.

[173] Dmitry Khakhulin, Florian Otte, Mykola Biednov, et al. Ultrafast X-ray Photochemistry at European XFEL: Capabilities of the Femtosecond X-ray Experiments (FXE) Instrument. *Applied Sciences*, 10(3):995, 2020.

[174] Thomas Tschentscher, Christian Bressler, Jan Grünert, et al. Photon Beam Transport and Scientific Instruments at the European XFEL. *Applied Sciences*, 7(6), 2017.

[175] M. Pergament, G. Palmer, M. Kellert, et al. Versatile optical laser system for experiments at the European X-ray free-electron laser facility. *Opt. Express*, 24(26):29349–29359, Dec 2016.

[176] A Mozzanica, A Bergamaschi, R Dinapoli, et al. The GOTTHARD charge integrating readout detector: design and characterization. *Journal of Instrumentation*, 7(01):C01019–C01019, jan 2012.

[177] J. Zhang, M. Andrä, R. Barten, et al. Performance evaluation of the analogue front-end and ADC prototypes for the gotthard-II development. *Journal of Instrumentation*, 12(12):C12052–C12052, dec 2017.

[178] L. Rota, M. Caselle, E. Bründermann, et al. KALYPSO: Linear array detector for high-repetition rate and real-time beam diagnostics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 936:10–13, 2019. Frontier Detectors for Frontier Physics: 14th Pisa Meeting on Advanced Detectors.

[179] Matthew Newville, Till Stensitzki, Daniel B. Allen, and Antonino Ingargiola. LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python, September 2014.

[180] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, July 1944.

[181] Donald W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963.

[182] Abraham. Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

[183] Harald Sinn. Stability Task Force. European XFEL PBS Meeting, October 2020.

[184] Anreas Koch. Beam stability measurements with imagers within the Stability Task Force. European XFEL PBS Meeting, November 2020.

[185] A. Mozzanica and J. Zhang. Gotthard-I Documentation, Release 0.3. `https://github.com/slsdetectorgroup/slsDetectorManual/blob/main/GotthardI/GotthardI_50um/GotthardI_50um_Manual.pdf`, 2016.

[186] Turan Erdogan. A New Class of Polarization Optics Designed Specifically for Lasers. *Semrock White Papers*, 2020.

[187] Michael Diez, Andreas Galler, Sebastian Schulz, et al. A self-referenced in-situ arrival time monitor for X-ray free-electron lasers. *Scientific Reports*, 11(1):3562, December 2021.

[188] Jay A. LaVerne, Igor Stefanic, and Simon M. Pimblott. Hydrated Electron Yields in the Heavy Ion Radiolysis of Water. *The Journal of Physical Chemistry A*, 109(42):9393–9401, 2005. PMID: 16866387.

[189] Tetsuo Katayama, Takashi Hirano, Yuki Morioka, et al. X-ray optics for advanced ultrafast pump-probe X-ray experiments at SACLA. *Journal of Synchrotron Radiation*, 26(2):333–338, 2019.

[190] K. Tono, T. Togashi, Y. Inubushi, et al. Beamline, experimental stations and photon beam diagnostics for the hard x-ray free electron laser of SACLA. *New Journal of Physics*, 15(8):083035, 2013.

[191] Sergei A. Trushin, Kyriaki Kosma, Werner Fuß, and Wolfram E. Schmid. Sub-10-fs supercontinuum radiation generated by filamentation of few-cycle 800 nm pulses in argon. *Optics Letters*, 32(16):2432–2434, 2007.

[192] M. Ekimova, W. Quevedo, M. Faubel, et al. A liquid flatjet system for solution phase soft-x-ray spectroscopy. *Structural Dynamics*, 2(5):054301, 2015.

[193] Kyo Nakajima, Yasumasa Joti, Tetsuo Katayama, et al. Software for the data analysis of the arrival-timing monitor at SACLA. *Journal of Synchrotron Radiation*, 25(2):592–603, Mar 2018.

[194] T.H. DiStefano and D.E. Eastman. The band edge of amorphous SiO2 by photoinjection and photoconductivity measurements. *Solid State Communications*, 9(24):2259 – 2261, 1971.

[195] Murray R. Spiegel and Larry J. Stephens. *Statistics*. Schaum's Outlines. McGraw-Hill Education, New York, sixth edition, 2017.

[196] David Freedman, Robert Pisani, and Roger Purves. *Statistics: Fourth International Student Edition*. W. W. Norton & Company, New York, 2007.

[197] R Swanepoel. Determination of the thickness and optical constants of amorphous silicon. *Journal of Physics E: Scientific Instruments*, 16(12):1214–1222, dec 1983.

[198] Stefan Droste, Sioan Zohar, Lingjia Shen, et al. High-sensitivity x-ray/optical cross-correlator for next generation free-electron lasers. *Opt. Express*, 28(16):23545–23553, Aug 2020.

[199] Li Tao, Craig Levin, and Ryan Coffee. Ionizing radiation induces femtosecond time scale modulations of a material's optical properties. In Gary P. Grim, H. Bradford Barber, Lars R. Furenlid, and Jeffrey A. Koch, editors, *Radiation Detectors in Medicine, Industry, and National Security XIX*, page 2, San Diego, United States, September 2018. SPIE.

[200] Li Tao, Ryan N Coffee, Diana Jeong, and Craig S Levin. Ionizing photon interactions modulate the optical properties of crystals with femtosecond scale temporal resolution. *Physics in Medicine & Biology*, 66(4):045032, feb 2021.

[201] P. B. Corkum, Claude Rolland, and T. Srinivasan-Rao. Supercontinuum Generation in Gases. *Physical Review Letters*, 57(18):2268–2271, November 1986.

A. Non Linear Optics

In normal optics Eq. 3.5a describes the response of a medium to an externally applied electric field. In most cases the response is linear. In the case of non linear optics the response is no longer linear, when the field strength of the external applied electric field is sufficiently high, which easily ia the case for an ultrashort laser pulses. In such a case the polarization $\boldsymbol{P}$ in Eq. 3.5a may be described in terms of a Taylor expansion

$$\begin{aligned} \boldsymbol{P}(\boldsymbol{E}) &= \epsilon_0\chi^{(1)}\boldsymbol{E} + \epsilon_0\chi^{(2)}\boldsymbol{E}^2 + \epsilon_0\chi^{(3)}\boldsymbol{E}^3 + \dots && (.1)\\ &= P^{(1)} + P^{(2)} + P^{(3)} + \dots && (.2)\\ &= P^{(1)} + P^{(NL)} && (.3) \end{aligned}$$

where $\chi^{(1)}$ is the dimensionless linear susceptibility. The $\chi^{(n)}$ susceptibilities are the n^{th}-order susceptibility and are a tensor of rank $n+1$ with 3^n components. The non linear susceptibility $\chi^{(2)}$ only occurs in acentric crystals. Media with $\chi^{(2)}$ nonlinearity are mainly used for nonlinear frequency conversion and electro-optic modulators. The $\chi^{(3)}$ nonlinearity term generates the *Kerr effect*, the *Raman effect* and *four wave mixing*.

Wave Mixing

One of the most used effects within non linear optics is frequency conversion. For most homogenous, loss-free materials and assuming the excitation being far away from resonances, the $\chi^{(n)}$ can be reduced to a real-valued scalar, which is true for many gases, liquids and glasses. Concentrating on the effect of $P^{(2)}$ one can write:

$$P^{(2)} = \epsilon_0\chi^{(2)}E^2. \tag{.4}$$

Now consider two incoming fields in the form of Eq. 3.35a with angular frequencies ω_1 and ω_2. Substituting Eq. 3.35a into Eq. .4:

$$\begin{aligned} P^{(2)}(2\omega_1) &= \epsilon_0\chi^{(2)}E_1^2 && \text{SHG, for } \omega_1 = \omega_2\\ P^{(2)}(\omega_1+\omega_2) &= 2\epsilon_0\chi^{(2)}E_1E_2 && \text{SFG}\\ P^{(2)}(\omega_1-\omega_2) &= 2\epsilon_0\chi^{(2)}E_1E_2^* && \text{DFG} \end{aligned} \tag{.5}$$

The first case is a special case where both incoming waves have the same angular frequency $\omega_1 = \omega_2$. In this case, a frequency component with twice the frequency is created. This process is called *second harmonic generation*. The second and third cases are the so called *sum frequency generation* (SFG) and *difference frequency generation* (DFG), where two waves with different frequencies can create a new frequency component with $\omega_{3+} = \omega_1 + \omega_2$ and $\omega_{3-} = \omega_1 - \omega_2$. A combination of these effects are commonly used to frequency double laser light and to create other harmonics by mixing, for instance, the fundamental carrier frequency of the laser with its second harmonic to create its third

harmonic frequency of the laser fundamental.
The described wave mixing effects have to fulfill next to energy conservation also the so called *phase matching* condition:

$$\boldsymbol{k}_{3+,3-} = \boldsymbol{k}_1 \pm \boldsymbol{k}_2 \tag{.6}$$

where $|\boldsymbol{k}_i| = \omega_i \, n(\omega_i) \, c_0^{-1}$.

Supercontinuum Generation

The physics of supercontinuum generation is quite involved and would go beyond the scope of this work. In this section the important main processes leading to an octave spanning supercontinuum are summarized. A comprehensive description can be found in [201] and [171]. The main contributions for supercontinuum generation are *Self-phase modulation* (SPM) and *four-wave mixing* (FWM). SPM and FWM are both consequences of the $\chi^{(3)}$ nonlinearity. All even numbered nonlinearities in Eq. .1 vanish for solids with inversion symmetry as well as for gases and liquids. The $\chi^{(3)}$ susceptibility describes FWM since the basic effect is the coupling of four waves, in analogy to the coupling of three waves already shown for the $\chi^{(2)}$ susceptibility. The third order polarization may be written as:

$$P^{(3)} = \epsilon_0 \chi^{(3)} E^3 \tag{.7}$$

In analogy to Equations .5 waves with now three different frequencies can mix and generate a new frequency component. All possible combinations are given by:

$$\omega_{i,j,j} = \omega_i +, \, \omega_j - \omega_k \qquad \text{for k} \neq i, j \tag{.8}$$

This results in 9 possible solutions. As before, there are again phase matching conditions which needs to be matched to reach high mixing efficiencies.
Another effect caused by the nonlinear $\chi^{(3)}$ susceptibility is the *Kerr effect*:

$$n(I) = n_0 + n_2 I \tag{.9}$$

The Kerr effect describes nonlinearities of the refractive index at high optical intensities. One of these effects ist SPM. As an intense ultrashort pulse passes through a $\chi^{(3)}$ material, its refractive index changes with the pulse intensity, which ultimately leads to a phase change $\Delta\phi$ of the transmitted wave. The effect is a self induced effect of the optical pulse and acts on the cause of this effect itself. Comparing the phase ϕ and ϕ_{NL} easily reveals

the phase shift $\Delta\phi$:

$$\begin{aligned}\Delta\phi &= \phi_{NL} - \phi \\ &= (\omega t - k_{NL} z) - (\omega t - kz)) \\ &= (k_{NL} - k) z \\ &= n_2 k_0 I z \end{aligned} \tag{.10}$$

The instantaneous frequency is then:

$$\begin{aligned}\omega(t) &= \omega_0 - \frac{\mathrm{d}\phi}{\mathrm{d}t} \\ &= \omega_0 - n_2 k_0 z \frac{\mathrm{d}I(t)}{\mathrm{d}t} \end{aligned} \tag{.11}$$

The effect of SPM on a bandwidth-limited pulse is depicted in Fig. .1. For reference the bandwidth-limited pulse in the temporal a) and frequency c) are shown as well. In the time domain the instantaneous frequency $\omega(t)$ is constant. With SPM the fast oscillating electric field of the pulse changed drastically. The early part of the electric field is shifted to lower frequencies wheres the later part of the pulse is shifted to higher frequencies which is shown in b). This leads to a broadening of the pulse in the spectral domain which means that additional frequencies around the carrier frequency ω_0 have been generated.

By a combination of FWM and SPM a broadband spectrum can be generated. At rather high intensities or long interaction region (optical fiber, hollow core fiber), even the additional frequencies generated by FWM can experience SPM and generate a very broadband spectrum together with all other generated frequencies.

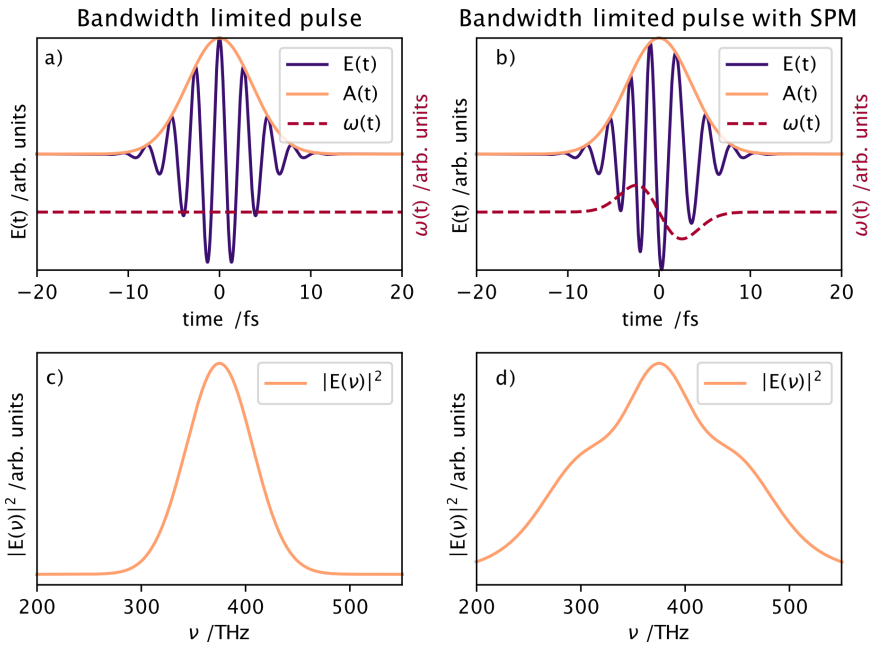

Figure .1.: The effect of SPM on a bandwidth-limited pulse. a) and c) are showing a bandwidth-limited pulse in temporal and spectral domain. b) and d) show the influence of SPM on a bandwidth-limited pulse in the temporal and spectral domain respectively.

B. Python Code

heat_propagation_appendix

March 25, 2022

[ ]:
```
#X-ray Heat Simulation in materials with Heat Diffusion Model
#Code by Michael
#European XFEL GmbH
#2020
```

[ ]:
```
import numpy as np
import heatequation
import matplotlib.pyplot as plt
import matplotlib.cm
from scipy import signal
import scipy.stats as st
from scipy.interpolate import interp1d
from scipy import constants
import lmfit
from lmfit import Model
from lmfit import Parameters
from cycler import cycler
plt.rcParams.update({'font.size': 14})
import matplotlib.colors as colors
import matplotlib.animation as animation

cmap=plt.cm.magma
# build cycler with 5 equally spaced colors from that colormap
c = cycler('color', cmap(np.linspace(.2,.8,5)) )
# supply cycler to the rcParam
plt.rcParams["axes.prop_cycle"] = c
```

[ ]:
```
%matplotlib notebook
```

[ ]:
```
def fit_cp(x,y):
    def func(x, m, c):
        return (m*x**(-1)+c)
    gmodel = Model(func, nan_policy='propagate')
    params = Parameters()
    #params.add('m',value=m)
    #params.add('c',value=20)
    result = gmodel.fit(y, x=x, m = 1000, c=10)
```



```
    return(result.best_fit, result.params)
```

[ ]:
```
plt.rcParams.update({'font.size': 14})
```

[ ]:
```
def photon_number(photon_energy, pulse_energy):
    'Calculates the number of photons in your pulse. Input photon energy in eV and pulse energy in J'
    number=pulse_energy/(photon_energy*1.6021773e-19)
    return int(number)

def electron_density(j = 0.002,ev=10000, d=100e-6,r=500e-6,material='diamond'):
    if material == 'diamond':
        e_pair = 12.2
        A = np.pi*r**2 # m2
        V = A*d #m3
        Eabs = j * x_abs(ev,thickness=d*1e6,material='diamond')
        #print(Eabs)
        dens_e = (Eabs/V)/(e_pair*1.60218e-19)
        return(dens_e)
    if material == 'sio2':
        e_pair = 17
        A = np.pi*r**2 # m2
        V = A*d #m3
        Eabs = j * x_abs(ev,thickness=d*1e6,material='sio2')
        #print(Eabs)
        dens_e = (Eabs/V)/(e_pair*1.60218e-19)
        return(dens_e)
    if material == 'si3n4':
        e_pair = 10.8
        A = np.pi*r**2 # m2
        V = A*d #m3
        Eabs = j * x_abs(ev,thickness=d*1e6,material='si3n4')
        #print(Eabs)
        dens_e = (Eabs/V)/(e_pair*1.60218e-19)
        return(dens_e)
    if material == 'water':
        e_pair = 2.4
        A = np.pi*r**2 # m2
        V = A*d #m3
        Eabs = j * x_abs(ev,thickness=d*1e6,material='water')
        #print(Eabs)
        dens_e = (Eabs/V)/(e_pair*1.60218e-19)
        return(dens_e)
    if material == 'yag':
        e_pair = 13
        A = np.pi*r**2 # m2
        V = A*d #m3
```



```
        Eabs = j * x_abs(ev,thickness=d*1e6,material='yag')
        #print(Eabs)
        dens_e = (Eabs/V)/(e_pair*1.60218e-19)
        return(dens_e)
    if material == 'silicon':
        e_pair = 3.6
        A = np.pi*r**2 # m2
        V = A*d #m3
        Eabs = j * x_abs(ev,thickness=d*1e6,material='si')
        #print(Eabs)
        dens_e = (Eabs/V)/(e_pair*1.60218e-19)
        return(dens_e)

###################scatter times calc with electron mobilities
##Diamond
dia_m_e = 0.25
dia_m_h = 0.25
dia_mu = 2700*1e-4
dia_tau_e = (dia_mu*dia_m_e*constants.m_e)/constants.e
dia_tau_h = (dia_mu*dia_m_h*constants.m_e)/constants.e
##SiO2
sio2_mu = 20*1e-4
sio2_m_e = 0.42
sio2_m_h = 0.58
sio2_tau_e = (sio2_mu*sio2_m_e*constants.m_e)/constants.e
sio2_tau_h = (sio2_mu*sio2_m_h*constants.m_e)/constants.e
##Si3N4
si3n4_m_e = 0.4
si3n4_m_h = 0.4
si3n4_mu = 1*1e-4
si3n4_tau_e = (si3n4_mu*si3n4_m_e*constants.m_e)/constants.e
si3n4_tau_h = (si3n4_mu*si3n4_m_h*constants.m_e)/constants.e
```

[ ]:
```
def gaussian(x, amp, cen, wid):
    #return(1/(wid*np.sqrt(2*np.pi))*np.exp(-(x-cen)**2/(2*wid**2)))
    return(amp*np.exp((-(x-cen)**2)/(2*(wid/2.35482)**2)))

def cp(T, material='diamond'):
    '''returns cp in J/kg/K'''
    from scipy import integrate
    def integrand(x):
        return((np.exp(x)*x**4)/(np.exp(x)-1)**2)
    if material == 'yag':
        Td = 795 #Debye Temp K
        gmol = 593.62 #g/mol
        integral = integrate.quad(integrand,0,Td/T)
        cv = 9*constants.N_A*constants.k*(T/Td)**3*(integral[0]-integral[1])
```



```
        return(cv/gmol*1000*20)
    if material == 'diamond':
        Td = 1860 #Debye Temp K
        gmol = 12.01 #g/mol
        integral = integrate.quad(integrand,0,Td/T)
        cv = 9*constants.N_A*constants.k*(T/Td)**3*(integral[0]-integral[1])
        return(cv/gmol*1000)
    if material == 'si':
        Td = 640 #Debye Temp K
        gmol = 28.0855 #g/mol
        integral = integrate.quad(integrand,0,Td/T)
        cv = 9*constants.N_A*constants.k*(T/Td)**3*(integral[0]-integral[1])
        return(cv/gmol*1000)
    if material == 'si_shomate':
        A = 22.81719
        B = 3.899510
        C = -0.082885
        D = 0.042111
        E = -0.354063
        t = T/1000
        gmol = 28.0855
        return( (A + B*t + C*t**2 + D*t**3 + E/t**2)/gmol*1000)
    if material == 'si3n4_shomate':
        A = 55.63214
        B = 157.7037
        C = -63.61395
        D = 8.610924
        E = 0.213440
        t = T/1000
        gmol = 140.2833
        return((A+B*t+C*t**2+D*t**3+E/t**2)/gmol*1000)
    if material == 'si3n4':
        Td = 1145 #Debye Temp K
        gmol = 26 #g/mol
        integral = integrate.quad(integrand,0,Td/T)
        cv = 9*constants.N_A*constants.k*(T/Td)**3*(integral[0]-integral[1])
        return(cv/gmol*1000)

def plot_fit(par,x):
    a = par['step_amp'].value/2*(1+erf((x-par['step_cen'].value)/par['step_wid'].value))
    b = par['dec_amp1'].value*np.exp(-par['dec_tau1'].value*(x-par['dec_x01'].value))
    c = par['dec_amp2'].value*np.exp(-par['dec_tau2'].value*(x-par['dec_x02'].value))
    return(a*(b+c))
```

```
def plot_fit3(par,x):
    b = par['dec_amp1'].value*np.exp(-par['dec_tau1'].value*(x-par['dec_x01'].
  value))
    c = par['dec_amp2'].value*np.exp(-par['dec_tau2'].value*(x-par['dec_x02'].
  value))
    return((b+c))

def plot_fit2(par,x):
    b = par['dec_amp1'].value*np.exp(-par['dec_tau1'].value*(x-par['dec_x01'].
  value))
    c = par['dec_amp2'].value*np.exp(-par['dec_tau2'].value*(x-par['dec_x02'].
  value))
    return((b+c))

def xray_abs_len(ev = 5000,material='diamond'):
    if material=='yag':
        data = np.genfromtxt('yag_atten.txt',skip_header=2)
        return(np.interp(ev,data[:,0],data[:,1]))
    if material=='diamond':
        data = np.genfromtxt('c_atten.txt',skip_header=2)
        return(np.interp(ev,data[:,0],data[:,1]))
    if material=='si':
        data = np.genfromtxt('si_atten.txt',skip_header=2)
        return(np.interp(ev,data[:,0],data[:,1]))
    if material=='si3n4':
        data = np.genfromtxt('si3n4_atten.txt',skip_header=2)
        return(np.interp(ev,data[:,0],data[:,1]))

def x_abs(ev,thickness=50,material='diamond'):
    import numpy as np
    if material == 'yag':
        c_data = np.genfromtxt('yag_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = 1-np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'diamond':
        c_data = np.genfromtxt('c_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = 1-np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'si3n4':
        c_data = np.genfromtxt('si3n4_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
```



```
        xabs = 1-np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'sio2':
        c_data = np.genfromtxt('sio2_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = 1-np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'si':
        c_data = np.genfromtxt('si_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = 1-np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'water':
        c_data = np.genfromtxt('h2o_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = 1-np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)

def x_trans(ev,thickness=50,material='diamond'):
    import numpy as np
    if material == 'yag':
        c_data = np.genfromtxt('yag_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'diamond':
        c_data = np.genfromtxt('c_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'si3n4':
        c_data = np.genfromtxt('si3n4_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'sio2':
        c_data = np.genfromtxt('sio2_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = np.exp(-thickness*(1/fit_c(ev)))
```



```
        return(xabs)
    if material == 'si':
        c_data = np.genfromtxt('si_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)
    if material == 'water':
        c_data = np.genfromtxt('h2o_atten.txt',skip_header=2)
        fit_c = interp1d(c_data[:,0], c_data[:,1], bounds_error=False,
  fill_value=np.nan)
        xabs = np.exp(-thickness*(1/fit_c(ev)))
        return(xabs)

def gaus2d(x=np.linspace(-10,10,21), y=np.linspace(-10,10,21), A=1, mu=0,
  sig=1):
    xy = np.zeros((len(x),len(y)))
    for i in range(len(x)):
        x_curr = x[i]
        xy[i] = A*np.exp(-((x_curr-mu)**2./(2.*sig**2.)+ (y-mu)**2./(2. *
  sig**2.)))
    return(xy)

def gauss2dnorm(x=np.linspace(-100,100,201), y=np.linspace(-100,100,201), A=1,
  mu=0, sig=1):
    xy = (A/(2*np.pi*sig**2))*np.exp(-((x-mu)**2./(2.*sig**2.)+
                                                    (y-mu)**2./(2. * sig**2.)))
    return(xy)

from scipy.special import erf
def step_func(x, step_amp, step_cen, step_wid):
    return(step_amp/2*(1+erf((x-step_cen)/step_wid)))
def exp_dec(x, dec_amp, dec_x0, dec_tau):
    return(dec_amp*np.exp(-dec_tau*(x-dec_x0)))

def fit_lambda(x,y):
    def step_func(x, step_amp, step_cen, step_wid):
        return(step_amp/2*(1+erf((x-step_cen)/step_wid)))
    def exp_dec1(x, dec_amp1, dec_x01, dec_tau1):
        return(dec_amp1*np.exp(-dec_tau1*(x-dec_x01)))
    def exp_dec2(x, dec_amp2, dec_x02, dec_tau2):
        return(dec_amp2*np.exp(-dec_tau2*(x-dec_x02)))

    gmodel = Model(step_func, nan_policy = 'propagate') * (Model(exp_dec1,
  nan_policy = 'propagate') + Model(exp_dec2, nan_policy = 'propagate'))
    params = Parameters()
```



```
    params.add('step_amp',value=1,vary=False)
    params.add('step_cen',value=150,min=0,max=1000)
    params.add('step_wid',value=40,min=0,max=1000)
    params.add('dec_amp1',value=5500)
    params.add('dec_x01',value=150)
    params.add('dec_tau1',value=0.006)
    params.add('dec_amp2',value=5500)
    params.add('dec_x02',value=150)
    params.add('dec_tau2',value=0.006)
    result = gmodel.fit(y, params, x = x)
    best_fits = result.best_fit
    return(best_fits, result.params)

def fit_lambda2(x,y,m=10000):
    def func(x, m, c):
        return (m*x**(-1)+c)
    gmodel = Model(func, nan_policy='propagate')
    params = Parameters()
    #params.add('m',value=m)
    #params.add('c',value=20)
    result = gmodel.fit(y, x=x, m = 1000, c=10)
    return(result.best_fit, result.params)

def fit_lambda3(x,y):
    def exp_dec1(x, dec_amp1, dec_x01, dec_tau1):
        return(dec_amp1*np.exp(-dec_tau1*(x-dec_x01)))
    def exp_dec2(x, dec_amp2, dec_x02, dec_tau2):
        return(dec_amp2*np.exp(-dec_tau2*(x-dec_x02)))
    gmodel = Model(exp_dec1, nan_policy = 'propagate') + Model(exp_dec2,
  nan_policy = 'propagate')
    params = Parameters()
    params.add('dec_amp1',value=100,min=0,max=10000)
    params.add('dec_x01',value=150,min=0,max=2000)
    params.add('dec_tau1',value=0.006,min=0)
    params.add('dec_amp2',value=5500,min=0)
    params.add('dec_x02',value=150,min=0,max=2000)
    params.add('dec_tau2',value=0.006,min=0)
    result = gmodel.fit(y, params, x = x)
    best_fits = result.best_fit
    return(best_fits, result.params)

def fit_alpha(x,y):
    def step_func(x, step_amp, step_cen, step_wid):
        return(step_amp/2*(1+erf((x-step_cen)/step_wid)))
    gmodel = Model(step_func, nan_policy = 'propagate')
    params = Parameters()
    params.add('step_amp',value=5e-6)
```

```
        params.add('step_cen',value=300,min=0,max=1000)
        params.add('step_wid',value=100,min=0,max=2000)
        result = gmodel.fit(y, params, x=x)
        return(result.best_fit, result.params)

    def p_density(J=.001,r=50e-6,dt=50e-15):
        '''returns power density in TW/cm2 '''
        E_peak = J/dt
        A = np.pi*r**2
        return(.0001/A*E_peak/1e12)

    def d_dens(alpha,T1,T2, material = 'diamond'):
        V = 1
        dT = abs(T2-T1)
        av = 3*alpha(T2)
        dV = av*dT*V
        V_new = dV/V + 1
        if material=='diamond':
            return(3540/V_new)
        if material=='si':
            return(2329/V_new)
        if material=='si3n4':
            return(3170/V_new)
```

[ ]:
```
    dia_lambda = np.load('lambda_dia.npy')
    si_lambda = np.load('lambda_si.npy')
    #si3n4_lambda = np.load('lambda_si3n4.npy')
    si3n4_lambda = np.genfromtxt('si3n4_lam.csv',delimiter=',')
    dia_lambda2 = np.genfromtxt('dia_lam3.csv',delimiter=',')
    yag_lambda = np.genfromtxt('yag_lam.csv',delimiter=',')
```

[ ]:
```
    dia_lambda_x = dia_lambda[:,0][12::]
    dia_lambda_y = dia_lambda[:,1][12::]
    si_lambda_x = si_lambda[:,0][10::]
    si_lambda_y = si_lambda[:,1][10::]
    si3n4_lambda_x = si3n4_lambda[:,0]
    si3n4_lambda_y = si3n4_lambda[:,1]
    yag_lambda_x = yag_lambda[:,0]
    yag_lambda_y = yag_lambda[:,1]
```

[ ]:
```
    dia_lamb_fit = fit_lambda3(dia_lambda_x, dia_lambda_y)
    si3n4_lamb_fit = fit_lambda3(si3n4_lambda_x, si3n4_lambda_y)
    si_lamb_fit = fit_lambda3(si_lambda_x, si_lambda_y)
    yag_lamb_fit = fit_lambda3(yag_lambda_x, yag_lambda_y)

    ax = np.linspace(0,15000,20000)
```



```
    plt.figure()
    plt.semilogy(yag_lambda_x,yag_lambda_y,'C3o')
    plt.semilogy(ax,plot_fit2(yag_lamb_fit[1],ax),'C3')
    plt.semilogy(dia_lambda_x,dia_lambda_y,'C0.')
    plt.semilogy(ax,plot_fit2(dia_lamb_fit[1],ax),'C0')
    plt.semilogy(si_lambda_x,si_lambda_y,'C2.')
    plt.semilogy(ax,plot_fit2(si_lamb_fit[1],ax),'C2')
    plt.semilogy(si3n4_lambda_x,si3n4_lambda_y,'C4.')
    plt.semilogy(ax,plot_fit2(si3n4_lamb_fit[1],ax),'C4')


    plt.xlim(130,1400)
    plt.ylim(1,5000)
```

[ ]:
```
    dia_alpha = np.genfromtxt('dia_alpha.csv',delimiter=',')
    si_alpha = np.genfromtxt('si_alpha.csv',delimiter=',')
    si3n4_alpha = np.genfromtxt('si3n4_alpha.csv',delimiter=',')
    yag_alpha = np.genfromtxt('yag_alpha.csv',delimiter=',')
    yag_alpha[:,1] = yag_alpha[:,1]*1e-6
    import numpy.polynomial.polynomial as poly
```

[ ]:
```
    order = 7
    ax = np.linspace(200,1550,2000)
    axx = np.linspace(100,2500,2000)
    ax_yag = np.linspace(100,2500,2000)
    coefs_dia = poly.polyfit(dia_alpha[:,0], dia_alpha[:,1], order)
    fit_dia_alpha = poly.polyval(ax, coefs_dia)
    coefs_si = poly.polyfit(si_alpha[:,0], si_alpha[:,1], order)
    fit_si_alpha = poly.polyval(ax, coefs_si)
    coefs_si3n4 = poly.polyfit(si3n4_alpha[:,0], si3n4_alpha[:,1], order)
    fit_si3n4_alpha = poly.polyval(ax, coefs_si3n4)
    coefs_yag = poly.polyfit(yag_alpha[:,0], yag_alpha[:,1], 3)
    fit_yag_alpha = poly.polyval(ax_yag, coefs_yag)
    ffit_dia_alpha = interp1d(ax,fit_dia_alpha, fill_value='extrapolate')
    ffit_si_alpha = interp1d(ax,fit_si_alpha, fill_value='extrapolate')
    ffit_si3n4_alpha = interp1d(ax,fit_si3n4_alpha, fill_value='extrapolate')
    ffit_yag_alpha = interp1d(yag_alpha[:,0],yag_alpha[:,1],
      fill_value='extrapolate')
    def ffit_yag_alpha(T):
        #return(-3.96e-6+6.75e-6*T+-1.06e-10*T**2+6.39e-14*T**3)
        return(1.1434e-5*np.exp(-174.59/T))
    plt.figure()
    plt.plot(dia_alpha[:,0],dia_alpha[:,1],'C0.')
    plt.plot(si_alpha[:,0],si_alpha[:,1],'C2.')
    plt.plot(si3n4_alpha[:,0],si3n4_alpha[:,1],'C4.')
    plt.plot(yag_alpha[:,0],yag_alpha[:,1],'C4.')
    plt.plot(ax, fit_dia_alpha,'C0')
```



```
    plt.plot(ax, fit_si_alpha,'C2')
    plt.plot(ax, fit_si3n4_alpha,'C4')
    plt.plot(axx, ffit_dia_alpha(axx),'C0')
    plt.plot(axx, ffit_si_alpha(axx),'C2')
    plt.plot(axx, ffit_si3n4_alpha(axx),'C4')
    plt.plot(axx, ffit_yag_alpha(axx),'C4')
    plt.ylim(0,12e-6)
```

[ ]:
```
    yag_lamb_fit[1]
```

[ ]:
```
    T_prop = np.linspace(0,15000,20000)

    cp_dia = []
    cp_si3n4 = []
    cp_si = []
    cp_yag = []
    for i in T_prop:
        cp_dia.append(cp(i,material='diamond'))
        cp_si3n4.append(cp(i,material='si3n4_shomate'))
        cp_si.append(cp(i,material='si_shomate'))
        cp_yag.append(cp(i,material='yag'))
    cp_dia=np.array(cp_dia)
    cp_si3n4=np.array(cp_si3n4)
    cp_si=np.array(cp_si)
    cp_yag=np.array(cp_yag)
    lamb_dia = plot_fit2(dia_lamb_fit[1],T_prop)
    lamb_si3n4 = plot_fit2(si_lamb_fit[1],T_prop)
    lamb_si = plot_fit2(si3n4_lamb_fit[1],T_prop)
    lamb_yag = plot_fit2(yag_lamb_fit[1],T_prop)
```

[ ]:
```
    ev = np.linspace(200,30000,1000)
    atten_dia = xray_abs_len(ev = ev,material='diamond')
    atten_si = xray_abs_len(ev = ev,material='si')
    atten_si3n4 = xray_abs_len(ev = ev,material='si3n4')
    atten_yag = xray_abs_len(ev = ev,material='yag')

    trans_dia = x_trans(ev = ev,material='diamond')
    trans_si = x_trans(ev = ev,material='si')
    trans_si3n4 = x_trans(ev = ev,material='si3n4')
    trans_yag = x_trans(ev = ev,material='yag')

    props = dict(facecolor='w', alpha=1)
    fig, ax = plt.subplots(2,1, sharex=False, figsize=(6.4,4.8))
    ax[0].semilogy(ev/1000,atten_dia,'C0',label='Dia')
    ax[0].semilogy(ev/1000,atten_si,'C2',label='Si')
    ax[0].semilogy(ev/1000,atten_si3n4,'C4',label='Si$_3$N$_4$')
    ax[0].semilogy(ev/1000,atten_yag,'C3',label='YAG')
```



```
    ax[1].plot(ev/1000,trans_dia,'C0',label='Dia')
    ax[1].plot(ev/1000,trans_si,'C2',label='Si')
    ax[1].plot(ev/1000,trans_si3n4,'C4',label='Si$_3$N$_4$')
    ax[1].plot(ev/1000,trans_yag,'C3',label='YAG')
    #ax[0].set_xlabel('X-ray photon energy /keV')
    #ax[0].xaxis.tick_top()
    #ax[0].xaxis.set_label_position('top')
    ax[0].axvspan(5, 25, alpha=0.3, color='grey')
    ax[1].axvspan(5, 25, alpha=0.3, color='grey')
    ax[1].set_xlabel('X-ray photon energy /keV')
    ax[0].set_ylabel('abs. length /μm')
    ax[0].set_xlim(.2,30)
    ax[1].set_xlim(.2,30)
    ax[1].set_ylabel('transmission')
    ax[0].text(.03, .9, 'a)', transform=ax[0].transAxes,
            verticalalignment='top', bbox=props)
    ax[1].text(.03, .9, 'b)', transform=ax[1].transAxes,
            verticalalignment='top', bbox=props)
    ax[0].legend(loc=(.725,.1),framealpha=1)
    ax[1].legend(loc=(.725,.1),framealpha=1)
    #plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/heat_sim/figures/
      material_xray_properties.pdf', bbox_inches='tight')
```

[ ]:
```
    def ddens(alpha,T1,T2,material='diamond'):
        from scipy import integrate
        def integrand(x):
            return((3*alpha(x)))
        V = 1
        integral = integrate.quad(integrand,T1,T2)[0]
        dV = integral*V
        V_new = dV/V+1
        if material=='diamond':
            return(3540/V_new)
        if material=='si':
            return(2329/V_new)
        if material=='si3n4':
            return(3170/V_new)
```

[ ]:
```
    def d_dens(alpha,T1,T2, material = 'diamond'):
        V = 1
        dT = abs(T2-T1)
        av = 3*alpha(T2)
        dV = av*dT*V
        V_new = dV/V + 1
        if material=='diamond':
            return(3540/V_new)
        if material=='si':
```

```
        return(2329/V_new)
    if material=='si3n4':
        return(3170/V_new)
    if material =='yag':
        return(4600/V_new)
```

```
[ ]: def temp_evolution(T_prop,x,y,T_start=293,sig=100e-6,
                       cryst_d=50e-6,J=.001,x_ray_energy=10000, dt=5e-9,
                       time_steps=40,
      pulse_num=10,material='diamond',d2='no',profile='gaussian'):
        e_dens = 0
        if material == 'diamond':
            dens = 3540 #kg/m3
            cp = cp_dia
            lamb = lamb_dia
            T_start = T_start
            D = np.interp(T_start,T_prop,lamb)/(dens*np.interp(T_start,T_prop,cp))
            e_dens = electron_density(j=J,ev=x_ray_energy,d=cryst_d,r=sig*2.35
      ,material='diamond')*1e-6
            print('electron density: '+str(e_dens)+' e/cm3')
        if material == 'si':
            dens = 2329 #kg/m3
            cp = cp_si
            lamb = lamb_si
            T_start = T_start
            D = np.interp(T_start,T_prop,lamb)/(dens*np.interp(T_start,T_prop,cp))
            e_dens = electron_density(j=J,ev=x_ray_energy,d=cryst_d,r=sig*2.35
      ,material='silicon')*1e-6
            print('electron density: '+str(e_dens)+' e/cm3')
        if material == 'si3n4':
            dens = 3170 #kg/m3
            cp = cp_si3n4
            lamb = lamb_si3n4
            T_start = T_start
            D = np.interp(T_start,T_prop,lamb)/(dens*np.interp(T_start,T_prop,cp))
            e_dens = electron_density(j=J,ev=x_ray_energy,d=cryst_d,r=sig*2.35
      ,material='si3n4')*1e-6
            print('electron density: '+str(e_dens)+' e/cm3')
        if material == 'yag':
            dens = 4600 #kg/m3
            cp = cp_yag
            lamb = lamb_yag
            T_start = T_start
            D = np.interp(T_start,T_prop,lamb)/(dens*np.interp(T_start,T_prop,cp))
            e_dens = electron_density(j=J,ev=x_ray_energy,d=cryst_d,r=sig*2.35
      ,material='yag')*1e-6
            print('electron density: '+str(e_dens)+' e/cm3')
```



```
        if profile == 'gaussian':
            x = x
            y = y
            dx = x[1]-x[0]
            dy = y[1]-y[0]
            print('Material: '+str(material)+'')
            print('Density: '+str(dens)+'')
            print('dx: '+str(dx)+'')
            print('dy: '+str(dy)+'')
            mu = 0
            sig = sig

            cryst_d = cryst_d #um
            print('cryst d: '+str(cryst_d)+'')
            J = J # J
            print('Pulse energy: '+str(J)+'')
            x_ray_energy = x_ray_energy # eV
            print('X-ray energy: '+str(x_ray_energy)+'')
            x_ray_abs = x_abs(ev=x_ray_energy, thickness = cryst_d*1e6, material =
      material)
            print('Absorbed Xray percentage: '+str(x_ray_abs)+'')
            m_dxdy = dx*dy*cryst_d*dens #weidght of one dx dy element in kg
            print('Mass dxdy: '+str(m_dxdy)+'')
            pulse_arr = gaus2d(x=x,y=y,A=1,mu=mu,sig=sig) # test array to scale
      final pulse arr
            j_max = J / pulse_arr.sum()
            pulse_arr = gaus2d(x=x,y=y,A=j_max,mu=mu,sig=sig) * x_ray_abs # array
      with J per dxdy
            print('Max J per dxdy: '+str(pulse_arr.max())+'')
            temp_arr = pulse_arr / (m_dxdy * np.interp(T_start,T_prop,cp)) #array
      with temps for each dxdy due to X-ray, dT
            print('Initial temp increase: '+str(temp_arr.max())+'')
            u0 = np.ones((len(x),len(y)))*293
            u = u0.copy()

            dx2, dy2 = dx*dx, dy*dy
            dt = dt
            F = dt * np.nanmax(lamb)/(dens*np.nanmax(cp))/dx**2
            if F > .5:
                return(print('Stability error: '+str(F)+''))
            print('Stability: '+str(F)+'')

            # Number of timesteps
            nsteps = time_steps #per pulse
            pulse_num = pulse_num
            T_evo = [u0]
            Tmax_evo = [u0.max()]
```



```
            if d2 == 'no':
                for i in range(pulse_num):
                    temp_arr = pulse_arr / (m_dxdy * np.interp(u0,T_prop,cp))
                    u0 = u0 + temp_arr # adding dT to current temperature array
                    for j in range(nsteps): # iterate demperature flux
                        #### calc cp and lambda for current T
                        ####D array for current temperture
                        #D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/(dens*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                        if material=='diamond':
                            #print(d_dens(ffit_dia_alpha,T_start,u0.
      max(),material=material), u0.min())
                            D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_dia_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                        if material=='si':
                            #print(d_dens(ffit_si_alpha,T_start,u0.
      max(),material=material), u0.min())
                            D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_si_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                        if material=='si3n4':
                            #print(d_dens(ffit_si3n4_alpha,T_start,u0.
      max(),material=material), u0.min())
                            D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_si3n4_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                        if material=='yag':
                            #print(d_dens(ffit_si3n4_alpha,T_start,u0.
      max(),material=material), u0.min())
                            D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_yag_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))

                        u[1:-1, 1:-1] = u0[1:-1, 1:-1] + D * dt * (
                          (u0[2:, 1:-1] - 2*u0[1:-1, 1:-1] + u0[:-2, 1:-1])/dx2
                          + (u0[1:-1, 2:] - 2*u0[1:-1, 1:-1] + u0[1:-1, :-2])/dy2 )
                        u0 = u.copy()
                        #T_evo.append(u0)
                        Tmax_evo.append(u0.max())
                        #print(u0.max(),u.max())
                #return(T_evo,Tmax_evo)
                return(Tmax_evo)
            if d2 == 'yes':
                for i in range(pulse_num):
                        temp_arr = pulse_arr / (m_dxdy * np.interp(u0,T_prop,cp))
                        u0 = u0 + temp_arr # adding dT to current temperature array
                        for j in range(nsteps): # iterate demperature flux
                            #### calc cp and lambda for current T
                            ####D array for current temperture
                            #D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/(dens*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                            if material=='diamond':
                                #print(d_dens(ffit_dia_alpha,T_start,u0.
      max(),material=material), u0.min())
                                D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_dia_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                            if material=='si':
                                #print(d_dens(ffit_si_alpha,T_start,u0.
      max(),material=material), u0.min())
                                D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_si_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                            if material=='si3n4':
                                #print(d_dens(ffit_si3n4_alpha,T_start,u0.
      max(),material=material), u0.min())
                                D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_si3n4_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))
                            if material=='yag':
                                #print(d_dens(ffit_si3n4_alpha,T_start,u0.
      max(),material=material), u0.min())
                                D = np.interp(u0[1:-1, 1:-1],T_prop,lamb)/
      (d_dens(ffit_yag_alpha,T_start,u0[1:-1, 1:-1],material=material)*np.
      interp(u0[1:-1,1:-1],T_prop,cp))

                            u[1:-1, 1:-1] = u0[1:-1, 1:-1] + D * dt * (
                              (u0[2:, 1:-1] - 2*u0[1:-1, 1:-1] + u0[:-2, 1:-1])/dx2
                              + (u0[1:-1, 2:] - 2*u0[1:-1, 1:-1] + u0[1:-1, :-2])/
      dy2 )
                            u0 = u.copy()
                            T_evo.append(u0)
                            Tmax_evo.append(u0.max())
                            #print(u0.max(),u.max())
                return(T_evo,Tmax_evo)
                    #return(Tmax_evo)
```



```
[ ]: x = np.linspace(-500e-6,500e-6,101)
     y = np.linspace(-500e-6,500e-6,101)
     pulses_num = 1
```

```
sig1 = 100e-6/2.35
sig2 = 100e-6/2.35
dt = 10e-9
J1 = 20e-6
J2 = 100e-6
time_window = 200e-6
time_steps = int(time_window/dt)
x_ray_energy = 10000


t_evo_dia1 = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig1,
                    cryst_d=50e-6,J=J1,x_ray_energy=x_ray_energy,

  pulse_num=1,dt=dt,time_steps=time_steps,material='diamond',d2='yes')
t_evo_si1 = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig1,
                    cryst_d=50e-6,J=J1,x_ray_energy=x_ray_energy,

  pulse_num=1,dt=dt,time_steps=time_steps,material='si',d2='yes')
t_evo_si3n41 = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig1,
                    cryst_d=50e-6,J=J1,x_ray_energy=x_ray_energy,

  pulse_num=1,dt=dt,time_steps=time_steps,material='si3n4',d2='yes')
t_evo_yag1 = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig1,
                    cryst_d=50e-6,J=J1,x_ray_energy=x_ray_energy,

  pulse_num=1,dt=dt,time_steps=time_steps,material='yag',d2='yes')


t_axis1 = np.linspace(0,dt*len(t_evo_dia1[1]),len(t_evo_dia1[1]))
```

[ ]:
```
plt.figure(figsize=(6.4,2.8))
plt.plot(t_axis1[0::]*1e6,np.array(t_evo_dia1[1][0::]),'C0',label='Dia')
plt.plot(t_axis1[0::]*1e6,np.array(t_evo_si1[1][0::]),'C2',label='Si')
plt.plot(t_axis1[0::]*1e6,np.array(t_evo_si3n41[1][0::
  ]),'C4',label='Si$_3$N$_4$')
plt.plot(t_axis1[0::]*1e6,np.array(t_evo_yag1[1][0::]),'C3',label='YAG')
plt.xlim(0,200)
plt.legend(loc='upper right')
plt.xlabel('time /µs')
plt.ylabel('temperature /K')
plt.tight_layout()
#plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/heat_sim/figures/
  x_ray_single_pulse_heat_100um_20uj.pdf', bbox_inches='tight')
```

[ ]:
```
t_max_off = 0
fig, ax = plt.subplots(2,2,sharex=True, sharey=True)
```



```
im1 = ax[0,0].pcolormesh(x*1e6,y*1e6,t_evo_si1[0][1],vmax=t_evo_si1[0][1].
  max()-t_max_off,cmap='magma', rasterized=True)
ax[0,1].pcolormesh(x*1e6,y*1e6,t_evo_si1[0][500],vmax=t_evo_si1[0][1].
  max()-t_max_off,cmap='magma', rasterized=True)
ax[1,0].pcolormesh(x*1e6,y*1e6,t_evo_si1[0][1000],vmax=t_evo_si1[0][1].
  max()-t_max_off,cmap='magma', rasterized=True)
ax[1,1].pcolormesh(x*1e6,y*1e6,t_evo_si1[0][6000],vmax=t_evo_si1[0][1].
  max()-t_max_off,cmap='magma', rasterized=True)
cbar1 = fig.colorbar(im1, ax=[ax[0,1]],aspect=12,shrink=2.2, anchor=(-.5, 1))
cbar1.set_label('Temperature /K', rotation=90)
fig.subplots_adjust(hspace=0.1)
fig.subplots_adjust(wspace=0.1)
fig.subplots_adjust(left=.15)
fig.subplots_adjust(right=.8)
plt.setp(ax[0,0].get_yticklabels()[0], visible=False)
plt.setp(ax[0,0].get_yticklabels()[3], visible=False)
plt.setp(ax[1,0].get_yticklabels()[0], visible=False)
plt.setp(ax[1,0].get_yticklabels()[3], visible=False)
ax[1,0].set_xticks([-250, 0, 250])
ax[1,1].set_xticks([-250, 0, 250])
ax[0,0].set_xlim(-300,300)
ax[0,0].set_ylim(-300,300)
#plt.setp(ax[1,0].get_xticklabels()[0], visible=False)
#plt.setp(ax[1,0].get_xticklabels()[1], visible=False)
#plt.setp(ax[1,1].get_xticklabels()[0], visible=False)
#plt.setp(ax[1,1].get_xticklabels()[1], visible=False)
ax[0,0].text(.07,.85,'a)',transform=ax[0,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[0,0].text(.4,.15,'0 µs',transform=ax[0,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[0,1].text(.07,.85,'b)',transform=ax[0,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[0,1].text(.4,.15,'5 µs',transform=ax[0,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[1,0].text(.07,.85,'c)',transform=ax[1,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[1,0].text(.4,.15,'10 µs',transform=ax[1,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[1,1].text(.07,.85,'d)',transform=ax[1,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[1,1].text(.4,.15,'60 µs',transform=ax[1,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
fig.text(0.48, 0.01, 'x-axis /µm', ha='center')
fig.text(0.01, 0.5, 'y-axis /µm', va='center', rotation='vertical')
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/heat_sim/figures/
  2d_heat_100um_20uj.pdf', bbox_inches='tight')
```


[ ]:
```
x = np.linspace(-500e-6,500e-6,101)
y = np.linspace(-500e-6,500e-6,101)
pulse_num = 20
sig = 100e-6/2.35
dt = 10e-9
J = 20e-6
time_window = 222e-9
time_steps = int(time_window/dt)
x_ray_energy = 10000
t_evo_dia= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=10e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_evo_si = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=10e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='si')
t_evo_si3n4 = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=10e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='si3n4')
t_evo_yag = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=10e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='yag')
t_axis = np.linspace(0,dt*len(t_evo_dia),len(t_evo_dia))
pulse_axis = np.linspace(0,pulse_num,len(t_evo_dia))




fig = plt.figure(figsize=(6.4,3))
ax1 = fig.add_subplot(111)
ax2 = ax1.twiny()
ax1.plot(t_axis*1e6,t_evo_dia,'C0',label='Dia')
ax1.plot(t_axis*1e6,t_evo_si,'C2',label='Si')
ax1.plot(t_axis*1e6,t_evo_si3n4,'C4',label='Si$_3$N$_4$')
ax1.plot(t_axis*1e6,t_evo_yag,'C3',label='YAG')
ax2.set_xticks([0,4,8,12,16])
ax1.set_xticks([0,1,2,3,4])
ax2.set_xlim(0,20)
ax1.set_xlim(0,4.45)
ax1.legend(loc=(.75,.12))
ax1.set_xlabel(r'time /µs')
ax1.set_ylabel('temperature /K')
ax2.set_xlabel('X-ray pulse number')
plt.tight_layout()
```



```
#plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/heat_sim/figures/
  x_ray_20_pulse_heat_100um_20uj.pdf', bbox_inches='tight')
```

[ ]:
```
x = np.linspace(-250e-6,250e-6,51)
y = np.linspace(-250e-6,250e-6,51)
pulse_num = 1000
sig = 100e-6/2.35
dt = 10e-9
J = 20e-6
time_window = 222e-9
time_steps = int(time_window/dt)
x_ray_energy = 10000
t_evo_dia_max= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_evo_si_max = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='si')
t_evo_si3n4_max = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='si3n4')
t_evo_yag_max = temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=x_ray_energy,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='yag')
t_axis_max = np.linspace(0,dt*len(t_evo_dia_max),len(t_evo_dia_max))
pulse_axis_max = np.linspace(0,pulse_num,len(t_evo_dia_max))
```

[ ]:
```
fig = plt.figure(figsize=(6.4,4.8))
ax1 = fig.add_subplot(211)
ax2 = ax1.twiny()
ax3 = fig.add_subplot(212)
ax1.plot(t_axis_max*1e6,t_evo_dia_max,'C0',label='Dia')
ax1.plot(t_axis_max[0:10209]*1e6,t_evo_si_max[0:10209],'C2',label='Si')
ax1.plot(t_axis_max[0:21671]*1e6,t_evo_si3n4_max[0:
  21671],'C4',label='Si$_3$N$_4$')
ax1.plot(t_axis_max[0:7987]*1e6,t_evo_yag_max[0:7987],'C3',label='YAG')
ax2.set_xticks([0,250,500,750,1000])
#ax1.set_xticks([0,1,2,3,4])
#ax2.set_xlim(0,220)
ax1.set_xlim(0,220)
ax3.set_xlim(0,220)
#ax3.set_ylim(290,358)
```

```
ax3.plot(t_axis_max*1e6,t_evo_dia_max,'C0',label='Dia')
#ax3.plot(t_axis*1e6,t_evo_dia,'C0',label='Dia')
#ax3.plot(t_axis*1e6,t_evo_si,'C2',label='Si')
ax1.legend(loc = (.7,.1))
fig.subplots_adjust(hspace=0)
fig.subplots_adjust(wspace=.1)
plt.setp(ax1.get_xticklabels()[0], visible=False)
ax3.set_xlabel('time /µs')
ax2.set_xlabel('X-ray pulse number')
fig.text(0.01, 0.5, 'temperature /K', va='center', rotation='vertical')
#plt.tight_layout()
ax1.text(.02,.87,'a)',transform=ax1.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
ax3.text(.02,.87,'b)',transform=ax3.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/heat_sim/figures/
  x_ray_200_pulse_heat_100um_20uj.pdf', bbox_inches='tight')
```

[ ]:
```
x = np.linspace(-250e-6,250e-6,51)
y = np.linspace(-250e-6,250e-6,51)
pulse_num = 2700
sig = 500e-6/2.35
dt = 10e-9
J = 2000e-6
time_window = 222e-9
time_steps = int(time_window/dt)
x_ray_energy = 10000
t_evo_dia1= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=10000,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_evo_dia2= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=8000,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_evo_dia3= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=5000,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_axis = np.linspace(0,dt*len(t_evo_dia1),len(t_evo_dia1))
pulse_axis = np.linspace(0,pulse_num,len(t_evo_dia1))
```

[ ]:
```
x = np.linspace(-250e-6,250e-6,51)
y = np.linspace(-250e-6,250e-6,51)
pulse_num = 2700
sig = 30e-6/2.35
```



```
dt = 10e-9
J = 20e-6
time_window = 222e-9
time_steps = int(time_window/dt)
x_ray_energy = 10000
t_evo_dia4= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=10000,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_evo_dia5= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=7000,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_evo_dia6= temp_evolution(T_prop=T_prop,x=x,y=y,T_start=293,sig=sig,
                    cryst_d=50e-6,J=J,x_ray_energy=5000,

  pulse_num=pulse_num,dt=dt,time_steps=time_steps,material='diamond')
t_axis = np.linspace(0,dt*len(t_evo_dia1),len(t_evo_dia1))
pulse_axis = np.linspace(0,pulse_num,len(t_evo_dia1))
```

[ ]:
```
fig = plt.figure(figsize=(6.4,4.8))
ax1 = fig.add_subplot(211)
ax2 = ax1.twiny()
ax3 = fig.add_subplot(212)
ax1.plot(t_axis*1e6,t_evo_dia1,'C0',label='10 keV')
ax1.plot(t_axis[0::]*1e6,t_evo_dia2[0::],'C2',label='7 keV')
ax1.plot(t_axis[0:13245]*1e6,t_evo_dia3[0:13245],'C4',label='5 keV')
ax2.set_xticks(np.linspace(0,2700,5))
#ax1.set_xticks([0,1,2,3,4])
#ax2.set_xlim(0,220)
ax1.set_xlim(0,600)
ax3.set_xlim(0,600)
#ax3.set_ylim(290,358)
ax3.plot(t_axis*1e6,t_evo_dia4,'C0',label='Dia')
ax3.plot(t_axis*1e6,t_evo_dia5,'C2',label='Dia')
ax3.plot(t_axis*1e6,t_evo_dia6,'C4',label='Dia')
#ax3.plot(t_axis*1e6,t_evo_dia,'C0',label='Dia')
#ax3.plot(t_axis*1e6,t_evo_si,'C2',label='Si')
ax1.legend(loc = (.7,.2))
fig.subplots_adjust(hspace=0)
fig.subplots_adjust(wspace=.1)
plt.setp(ax1.get_xticklabels(), visible=False)
ax3.set_xlabel('time /µs')
ax2.set_xlabel('X-ray pulse number')
fig.text(0.01, 0.5, 'temperature /K', va='center', rotation='vertical')
#plt.tight_layout()
```



```
ax1.text(.02,.87,'a)',transform=ax1.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
ax3.text(.02,.87,'b)',transform=ax3.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/heat_sim/figures/
  xray_2700_pulses_diff_conditions.pdf', bbox_inches='tight')
```

[ ]:
```
fig, ax = plt.subplots(2, figsize=(8,4))
plt.subplots_adjust(hspace = .000)
ax[0].plot(t_axis1[0::]*1e6,np.array(t_evo_dia1[1][0::]),'C0',label='Dia')
ax[0].plot(t_axis1[0::]*1e6,np.array(t_evo_si1[1][0::]),'C2',label='Si')
ax[0].plot(t_axis1[0::]*1e6,np.array(t_evo_si3n41[1][0::
  ]),'C4',label='Si$_3$N$_4$')
ax[0].plot(t_axis1[0::]*1e6,np.array(t_evo_yag1[1][0::]),'C3',label='YAG')
ax[0].set_xlim(0,200)
ax[0].legend()
ax[0].axes.get_xaxis().set_ticks([])


ax[1].plot(t_axis_max*1e6,t_evo_dia_max,'C0',label='Dia')
ax[1].plot(t_axis_max[0:10209]*1e6,t_evo_si_max[0:10209],'C2',label='Si')
ax[1].plot(t_axis_max[0:21671]*1e6,t_evo_si3n4_max[0:
  21671],'C4',label='Si$_3$N$_4$')
ax[1].plot(t_axis_max[0:7987]*1e6,t_evo_yag_max[0:7987],'C3',label='YAG')
ax[1].set_xlim(0,200)

inset=ax[1].inset_axes((.58,.2,.4,.6))
inset.plot(t_axis*1e6,t_evo_dia,'C0',label='Dia')
inset.plot(t_axis*1e6,t_evo_si,'C2',label='Si')
inset.plot(t_axis*1e6,t_evo_si3n4,'C4',label='Si$_3$N$_4$')
inset.plot(t_axis*1e6,t_evo_yag,'C3',label='YAG')
inset.set_xlim(-.1,2.5)
inset.set_ylim(290,350)

plt.setp(ax[1].get_yticklabels()[-1], visible=False)
#plt.setp(inset.get_yticklabels(), visible=False)
plt.setp(inset.get_xticklabels(), visible=False)

ax[1].set_xlabel(r'time /µs')
fig.text(0.03, 0.35,'temperature /K', ha='center', rotation='vertical')

ax[0].text(.02,.82,'a)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
ax[1].text(.02,.82,'b)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
```



```
inset.text(.05,.74,'c)',transform=inset.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
plt.savefig('t_evo_comp_paper.pdf',bbox_inches='tight')
```

timing_tool_sim_appendix

March 25, 2022

[ ]:
```
#Self-Referenced Timing-Tool Simulation
#Code by Michael Diez
#2020
```

[ ]:
```
import os
import numpy as np
import matplotlib.pyplot as plt
import os
from scipy.signal import savgol_filter
import lmfit
from lmfit import Model
from scipy.stats import norm
from cycler import cycler
from scipy.stats import gaussian_kde
from matplotlib.patches import Circle
from scipy.special import erf
from copy import deepcopy
from scipy.signal import find_peaks
from matplotlib.widgets import Slider
import numpy.polynomial.polynomial as poly
import pickle
from scipy import constants
from functools import reduce
from scipy import interpolate
from matplotlib.colors import LogNorm
from scipy import integrate
import matplotlib.patches as mpatches
import scipy.integrate as integrate
from decimal import Decimal
import matplotlib
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
from mpl_toolkits.axes_grid1.inset_locator import InsetPosition

cmap=plt.cm.magma
c = cycler('color', cmap(np.linspace(.2,.8,5)) )
plt.rcParams["axes.prop_cycle"] = c
```



```
import sys
sys.path.append('/Users/diezm/Nextcloud/exfel/beamtime2/reduced')
import exfel_ana as ex
sys.path.append('/Users/diezm/Nextcloud/
  Electrons_and_refractive_index_upon_X-ray_illumniation')
import sim_tools as st
```

[ ]:
```
%matplotlib notebook
```

[ ]:
```
plt.rcParams.update({'font.size': 14})
```

[ ]:
```
import lmfit
from lmfit import Model
from lmfit import Parameters

def
  plot_multigauss(x,cen1,amp1,wid1,cen2,amp2,wid2,cen3,amp3,wid3,cen4,amp4,wid4):

    gauss1 = amp1*np.exp((-(x-cen1)**2)/(2*(wid1/2.35482)**2))
    gauss2 = amp2*np.exp((-(x-cen2)**2)/(2*(wid2/2.35482)**2))
    gauss3 = amp3*np.exp((-(x-cen3)**2)/(2*(wid3/2.35482)**2))
    gauss4 = amp4*np.exp((-(x-cen4)**2)/(2*(wid4/2.35482)**2))
    return(gauss1+gauss2+gauss3+gauss4)

def plot_multigauss_fit(x,params):
    gauss1 = params[1]['amp_1'].value*np.exp((-(x-params[1]['cen_1'].value)**2)/
  (2*(params[1]['wid_1'].value/2.35482)**2))
    gauss2 = params[1]['amp_2'].value*np.exp((-(x-params[1]['cen_2'].value)**2)/
  (2*(params[1]['wid_2'].value/2.35482)**2))
    gauss3 = params[1]['amp_3'].value*np.exp((-(x-params[1]['cen_3'].value)**2)/
  (2*(params[1]['wid_3'].value/2.35482)**2))
    gauss4 = params[1]['amp_4'].value*np.exp((-(x-params[1]['cen_4'].value)**2)/
  (2*(params[1]['wid_4'].value/2.35482)**2))
    return(gauss1+gauss2+gauss3+gauss4)

def gaussian(x,cen,amp,wid):
    return(amp*np.exp((-(x-cen)**2)/(2*(wid/2.35482)**2)))

def fit_multigauss(ax,data,
                   cen_1=408e-9, amp_1=1000, wid_1=5e-9,
                   cen_2=408e-9, amp_2=1000, wid_2=5e-9,
                   cen_3=408e-9, amp_3=1000, wid_3=5e-9,
                   cen_4=408e-9, amp_4=1000, wid_4=5e-9):
    def gauss1(x, amp_1, cen_1, wid_1):
        return(amp_1*np.exp((-(x-cen_1)**2)/(2*(wid_1/2.35482)**2)))
    def gauss2(x, amp_2, cen_2, wid_2):
```



```
        return(amp_2*np.exp((-(x-cen_2)**2)/(2*(wid_2/2.35482)**2)))
    def gauss3(x, amp_3, cen_3, wid_3):
        return(amp_3*np.exp((-(x-cen_3)**2)/(2*(wid_3/2.35482)**2)))
    def gauss4(x, amp_4, cen_4, wid_4):
        return(amp_4*np.exp((-(x-cen_4)**2)/(2*(wid_4/2.35482)**2)))
    gmodel = Model(gauss1, nan_policy = 'propagate') + Model(gauss2, nan_policy
  = 'propagate')+ Model(gauss3, nan_policy = 'propagate')+Model(gauss4,
  nan_policy = 'propagate')
    params = Parameters()
    params.add('amp_1',value=amp_1,min=0)
    params.add('cen_1',value=cen_1,min=380e-9,max=450e-9)
    params.add('wid_1',value=wid_1,min=0, max=40e-9)
    params.add('amp_2',value=amp_2,min=0)
    params.add('cen_2',value=cen_2,min=380e-9,max=450e-9)
    params.add('wid_2',value=wid_2,min=0, max=40e-9)
    params.add('amp_3',value=amp_3,min=0)
    params.add('cen_3',value=cen_3,min=380e-9,max=450e-9)
    params.add('wid_3',value=wid_3,min=0,max=40e-9)
    params.add('amp_4',value=amp_4,min=0)
    params.add('cen_4',value=cen_4,min=380e-9,max=450e-9)
    params.add('wid_4',value=wid_4,min=0,max=40e-9)
    result = gmodel.fit(data, params, x = ax)
    fit_params = result.params
    best_fit = result.best_fit
    return(best_fit,fit_params)
```

[ ]:
```
def v_to_l(v):
    return((constants.c/v))

def l_to_v(l):
    return((constants.c/l))

def v_to_w(v):
    return(2*np.pi*v)

def w_to_v(w):
    return(w/(2*np.pi))

def l_to_w(l):
    return(v_to_w(l_to_v(l)))

def w_to_l(w):
    return(v_to_l(w_to_v(w)))

def fft(spec):
    fft=np.fft.fftshift(np.fft.fft(np.fft.fftshift(spec)))
    return(fft)
```



```
def ifft(spec):
    ifft=np.fft.fftshift(np.fft.ifft(np.fft.fftshift(spec)))
    return(ifft)

def nextpow2(x):
    return(math.ceil(np.log2(abs(x))))

def gaussian(x,amp,mu,sig, mode = 'FWHM'):
    if mode == 'FWHM':
        return(amp*np.exp((-(x-mu)**2)/(2*(sig/2.35482)**2)))
    if mode == 'RMS':
        return(amp*np.exp((-(x-mu)**2)/(2*(sig*1)**2)))

def step_func(x, step_amp, step_cen, step_wid):
    return(step_amp/2*(1+erf((x-step_cen)/step_wid)))

def t_to_v(E,t):
    n = len(t)
    #dw = 2.0*np.pi/(t[-1] - t[0])
    #w = np.arange(-n/2,n/2)*dw
    dt = t[1] - t[0]
    sf = 1/dt
    f = np.linspace(-sf/2,sf/2,n)
    F = fft(E)#/dt#/np.sqrt(2.0*np.pi)
    return(F,f)

def v_to_t(F,v):
    n = len(v)
    #dv = v[1] - v[0]
    #t = np.arange(-n/2,n/2)*(1/(v[-1] - v[0]))
    sf = v[1] - v[0]
    dt = 1 / sf
    E = ifft(F)#/dt#sf/np.sqrt(2.0*np.pi)*n
    return(E)

def d_thickness(d,deg):
    phi_rot = np.deg2rad(90 - deg)
    return(d/np.sin(phi_rot))

def sellmaier_sio2(l):
    #l = l*1e6
    n=(1+0.6961663/(1-(0.0684043/l)**2)+0.4079426/(1-(0.1162414/l)**2)+0.
  8974794/
       (1-(9.896161/l)**2))**.5
    return(n)
```

```
def sellmaier_bbo_o(l):
    n=(1+0.90291/(1-0.003926/l**2)+0.83155/(1-0.018786/l**2)+0.76536/(1-60.01/
  l**2))**.5
    return(n)

def sellmaier_bbo_e(l):
    n=(1+1.151075/(1-0.007142/l**2)+0.21803/(1-0.02259/l**2)+0.656/(1-263/
  l**2))**.5
    return(n)

def sellmaier_dia(l):
    n=(1+0.3306/(1-(0.1750/l)**2)+4.3356/(1-(0.1060/l)**2))**.5
    return(n)

def calc_disp(n,l,l0,n0=1):
    from scipy.constants import c as c
    '''
    Input n,l,l0
    n: refractive index, get from sellmaier equations
    l: wavelengths in um -> i.e np.linspace(.3,1,10000)
    l0: central wavelengths for dispersion in um -> i.e .8
    '''
    ### calc k0
    f0 = l_to_v(l0*1000*1e-9)
    k0 = 1/(n0*f0/(constants.c/1e6))
    ### calc d/dn in fs**2 / mm
    diff1 = np.diff(n)/(l[-1]-l[0])*len(l)
    diff2 = np.diff(diff1)/(l[-1]-l[0])*len(l)
    diff3 = np.diff(diff2)/(l[-1]-l[0])*len(l)
    l1 = l[1::]
    l2 = l[2::]
    l3 = l[3::]
    d1n = diff1[abs(l1-l0).argmin()]
    d2n = diff2[abs(l2-l0).argmin()]
    d3n = diff3[abs(l3-l0).argmin()]
    ### calc vg
    lambda0 = l0*1e-3
    vg = (c/n0)/(1-(l0/n0)*d1n)
    ### calc GVD
    lambda0 = l0*1e-3#800e-9 * 1e3
    cmm = c/1e12
    dn2 = d2n * 1e6
    gvd = lambda0**3/(np.pi*2*cmm**2)*dn2
    ### calc TOD
    lambda0 = l0*1e-3#800e-9 * 1e3
    cmm = c/1e12
    dn3 = d2n * 1e6
```



```
    dn3 = d3n * 1e9
    tod = -(lambda0/(2*np.pi*cmm))**2*(1/cmm)*(3*lambda0**2*dn2+lambda0**3*dn3)
    #print(d1n,d2n,d3n,gvd,tod)
    return(d1n,d2n,k0,gvd,tod,vg)

#def n_change(f,amp=50e-15,f0=3.3e14,wid=.3e13):
#    f0_pos = abs(f - f0).argmin()
#    phi = step_func(f,step_amp=amp,step_cen=f0,step_wid=wid)
#    return(phi)

def n_change(f,amp=50e-15,f0=3.3e14,wid=.3e13):
    f0_pos = abs(f - f0).argmin()
    offset = 1/(sellmaier_dia(v_to_l(f0)*1e6)*f0/(constants.c/1e6))
    phi = step_func(f,step_amp=amp,step_cen=f0,step_wid=wid) + offset
    return(phi)

#k = n0 * w / c  -> 1/um

def apply_dispersion(Ef,f,n0=1,phi=0,GVD=0,TOD=0,FOD=0,z=0):
    from scipy import constants
    #GVD = (GVD*1e-15)**2
    #TOD = (TOD*1e-15)**3
    #FOD = (FOD*1e-15)**4
    freq = f
    gvd = (GVD*(1e-15**2))*10000*np.pi
    tod = (TOD*(1e-15**3))*10000*np.pi
    Hw1 = np.exp(-1j*(n0/(constants.c))*2*freq * z)
    Hw2 = np.exp(-1j*freq**2*(gvd/2*z))
    Hw3 = np.exp(-1j*freq**3*(tod/6*z))
    Hw_phi = np.exp(-1j*freq*phi)
    out = Ef*Hw1*Hw2*Hw3*Hw_phi
    #delta = np.exp(-1j*(f*GV*z))
    #delta = np.exp(-1j*(freq*phi+freq**2*GVD/2.+ freq**3*TOD/2./3.+
  freq**4*FOD/2./3./4.)*z)
    #delta = np.exp(-1j*(freq*phi+freq**2*GVD+ freq**3*TOD+ freq**4*FOD)*z)
    return(out)

def d_thickness(d,deg):
    phi_rot = np.deg2rad(90 - deg)
    return(d/np.sin(phi_rot) - d)

def d_phase(nd,f0=l_to_v(400e-9),no=sellmaier_bbo_o(.4),ne=sellmaier_bbo_e(.4)):
    nd = nd / 100
    df0 = 1/f0
    to = (nd*no) / constants.c
    te = (nd*ne) / constants.c
    return((to-te)/df0)
```



```
def d_time(nd,f0=l_to_v(400e-9),no=sellmaier_bbo_o(.4),ne=sellmaier_bbo_e(.4)):
    nd_m = nd / 100
    df0 = 1/f0
    to = (nd_m*no) / constants.c
    te = (nd_m*ne) / constants.c
    return(((to-te) - df0)+df0)


#def d_phi(n0,dn,d,l0=400e-9):
#    f = constants.c/l0
#    t1 = d/(constants.c/n0)
#    t2 = d/(constants.c/(n0+dn))
#    dt = t1-t2
#    phi_t = 1 /f
#    dphi = dt/phi_t
#    return(dphi)

def d_phi(dn,l,d):
    return((2*np.pi*dn)/l*d)

def phi_shift(Et,t,t0,dphi,shift_wid):
    'shift in [rad]'
    shift = step_func(x=t,step_amp=dphi,step_cen=t0,step_wid=shift_wid)
    phi = np.exp(-1j*shift)
    #phi = np.exp(-1j*shift)
    return(Et*phi)

def cen_time(Et1, Et2 = 'none'):
    if Et2 == 'none':
        return(np.roll(Et1,-Et1.argmax()+int(len(Et1)/2)))
    else:
        max1 = Et1.argmax()
        max2 = Et2.argmax()
        offset = abs(max1-max2)/2
        Et1_cen = np.roll(Et1,-Et1.argmax()+int(len(Et1)/2+offset))
        Et2_cen = np.roll(Et2,-Et2.argmax()+int(len(Et2)/2-offset))
        return(Et1_cen,Et2_cen)

def d_abs(Et,t,t0,d_abs,abs_wid):
    shift = step_func(x=t,step_amp=-d_abs,step_cen=t0,step_wid=abs_wid) + 1
    return(Et*shift)

def etalon_spec(d,l,n=1.35,shift=0):
    dr = 2*n*d
    return(np.cos(2*np.pi*dr/l+(l-shift)))
```



```
def fresnel(n=2.42,dn=-.0001):
    n1 = 1
    n2 = n
    r_norm = abs((n1-n2)/(n1+n2))**2
    r_trans = abs((n1-n2+dn)/(n1+n2+dn))**2
    return(r_norm - r_trans)

def delta_d(dphi = .5, l0 = 400e-9, n1 = sellmaier_bbo_o(.4), n2 =
  sellmaier_bbo_e(.4)):
    '''difference thickness needed for .5 phase shift'''
    return((dphi*l0)/(2*np.pi*(n1-n2)))

#def d_rot(d,delta_d):
#    '''rotation of second bbo needed to achieve .5 phase shift'''
#    return(np.rad2deg(np.arccos(d/(d+delta_d))))

def dT_dia(n0,T):
    '''Temperature dependant refractive index of CVD diamond'''
    return(n0+7.47e-6*T+2.09e-8*T**2+2.92e-12*T**3)

def convert_l_plot(f,spec):
    l = v_to_l(f)
    mid = int(len(f)/2)+1
    l = l[mid::]
    spec = spec[mid::]
    return(l,spec)

def gaussian_fraction(x_inf=[-1000,1000],x1=-2,x2=2,amp=1,cen=0,wid=5):
    '''Calculates the fraction between pump and probe beam sizes'''
    amp = amp
    cen = cen
    wid = wid
    def f(x):
        return(amp*np.exp((-(x-cen)**2)/(2*(wid/2.35482)**2)))
    f_inf = integrate.quad(f,x_inf[0],x_inf[1])[0]
    f_x1x2 = integrate.quad(f,x1,x2)[0]
    return(f_x1x2/f_inf)

def gauss2dnorm(x=np.linspace(-100,100,201), y=np.linspace(-100,100,201), A=1,
  mu=0, sig=1):
    xy = (A/(2*np.pi*sig**2))*np.exp(-((x-mu)**2./(2.*sig**2.)+
                                       (y-mu)**2./(2. * sig**2.)))
    return(xy)

def gauss2dnormasym(x=np.linspace(-100,100,201), y=np.linspace(-100,100,201),
  A=1, mu=0, sigx=1, sigy=1):
    xy = (A/(np.pi*sigx**2+np.pi*sigy**2))*np.exp(-((x-mu)**2./(2.*sigx**2.)+
```

```
                                                (y-mu)**2./(2. * sigy**2.)))
    return(xy)

def jm2_to_j(jm2,r,ry=0):
    'Calculates pulse energy for a given j/m2. needs radius_x and radius_y (if
   eliptical beam). Calculates full beam, not FWHM'
    if ry == 0:
        A = np.pi*(2*r)**2 #factor 2 to roughly get90% of full beam
        J = jm2*A
        return(J)
    else:
        A = 2*np.pi*r*ry
        J = jm2*A
        return(J)

def f_inst(Et,t):
    from scipy.signal import hilbert, chirp
    fs = len(t)/t[-1]
    analytic_signal = hilbert(np.real(Et))
    amplitude_envelope = np.abs(np.real(Et))
    instantaneous_phase = np.unwrap(np.angle(analytic_signal))
    instantaneous_frequency = (np.diff(instantaneous_phase) / (2*np.pi)*fs)/
   (2*np.pi)*np.pi
    return(instantaneous_frequency)

def d_rot_phi(d = .005, dphi = .5, l0 = 400e-9, n1 = sellmaier_bbo_o(.4), n2 =
   sellmaier_bbo_e(.4)):
    '''difference thickness needed for .5 phase shift'''
    deg = np.rad2deg(np.arccos(d/(d+abs(constants.c/n1 - constants.c/n2)*(l0/
   constants.c)*dphi)))
    return(deg)

def d_rot(d=.01,phi=0):
    rad = np.deg2rad(phi)
    dd = d/np.cos(rad)-d
    return(dd)

def zero_angle(x,y):
    'Calcs angle between (0,0) and (x,y)'
    return(np.rad2deg(np.arctan2(x,y)))

def zero_dist(x,y):
    'distance from (0,0) to (x,y)'
    return(np.sqrt(x**2+y**2))

def rotate(x,y,angle):
    'rotates a function of x,y values by an angle'
```



```
    angle = np.deg2rad(angle)
    xx = x*np.cos(angle) - y*np.sin(angle)
    yy = x*np.sin(angle) + y*np.cos(angle)
    coef = poly.polyfit(xx,yy,2)
    fit = poly.Polynomial(coef)
    return(xx,yy)

def jones(Ex,Ey,angle):
    'Calculates the change of Ex and Ey polarization components after passing
   through polarizer and returns these two components as well as the final
   transmitted electric field'
    rad = np.deg2rad(angle)
    sp = np.vstack((Ex,Ey))
    m = np.array([[np.cos(rad)**2 , np.cos(rad)*np.sin(rad)],[np.sin(rad)*np.
   cos(rad), np.sin(rad)**2]])
    Exy = np.matmul(m,sp)
    Ex0 = np.where(Exy[0]<0)[0]
    E_new = np.sqrt(Exy[0]**2+Exy[1]**2)
    E_new[Ex0] = E_new[Ex0]*-1
    return(Exy,E_new)

def d_rot_inv(d,dn):
    return(np.rad2deg(np.arccos(d/(dn+d))))
```

[ ]:

```
def sim_tt(dn=.0001,dn_wid=20e-15,d_absorption = 0,
   t0=0,l0=400e-9,dl0=50e-9,d_samp=50e-6,amp=1,
           d_disp=.15, d_phi_bbo=.5, angle_pol = -45, d_bbo = .05, dn_bbo=0,
           num=2**21,f_range = 1e16, shape = 'gaussian', f=1, func = 1,
   mode='polarisation'):
    num = num
    if shape == 'gaussian':
        f = np.linspace(-f_range,f_range,num)
        w = v_to_w(f)
        df = 1/(f[1] - f[0])
        t = np.linspace(-df/2, df/2, num)
        f0 = l_to_v(l0)
        w0 = v_to_w(f0)
        df0 = l_to_v(l0-dl0/2) - l_to_v(l0+dl0/2)
        E0 = amp
        Ef = amp*np.exp(-(f-f0)**2/(2*(df0/2.3548)**2))
        l = np.linspace(.25,1.2,10000)

    if shape == 'custom':
        f = f
        num = len(f)
        w = v_to_w(f)
        df = 1/(f[1] - f[0])
```



```
        t = np.linspace(-df/2, df/2, num)
        f0 = l_to_v(l0)
        w0 = v_to_w(f0)
        df0 = l_to_v(l0-dl0/2) - l_to_v(l0+dl0/2)
        E0 = amp
        Ef = np.sqrt(func)
        l = np.linspace(.25,1.2,10000)

    n_sio2 = sellmaier_sio2(l)
    n_bbo_o = sellmaier_bbo_o(l)
    n_bbo_e = sellmaier_bbo_e(l)
    n0_bbo_o = sellmaier_bbo_o(l0*1e6)
    n0_bbo_e = sellmaier_bbo_e(l0*1e6)
    n0_sio2 = sellmaier_sio2(l0*1e6)
    dn1,dn2,dn3,gvd_sio2,tod_sio2,gv_sio2 =
   calc_disp(n_sio2,l,l0*1e6,n0=n0_sio2)
    dn1,dn2,dn3,gvd_bbo_o,tod_bbo_o,gv_bbo_o =
   calc_disp(n_bbo_o,l,l0*1e6,n0=n0_bbo_o)
    dn1,dn2,dn3,gvd_bbo_e,tod_bbo_e,gv_bbo_e =
   calc_disp(n_bbo_e,l,l0*1e6,n0=n0_bbo_e)
    Ef_disp = apply_dispersion(Ef=Ef, f=f,
   n0=n0_sio2,GVD=gvd_sio2,TOD=tod_sio2,z=d_disp)

    Et = ifft(Ef)
    Et_disp = ifft(Ef_disp)
    Et_disp = cen_time(Et_disp)
    Ef_disp_o = apply_dispersion(Ef=Ef_disp, f=f,
   n0=n0_bbo_o,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=d_bbo)
    Ef_disp_e = apply_dispersion(Ef=Ef_disp, f=f,
   n0=n0_bbo_e,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=d_bbo)
    Et_disp_o = ifft(Ef_disp_o)
    Et_disp_e = ifft(Ef_disp_e)
    Et_disp_o,Et_disp_e = cen_time(Et1=Et_disp_o, Et2=Et_disp_e)
    #t0 = 5*750e-15
    dphi = d_phi(dn=dn,l=l0,d=d_samp)
    #print(dphi)
    Et_disp_o_dn =
   d_abs(Et=Et_disp_o,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
    Et_disp_e_dn =
   d_abs(Et=Et_disp_e,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
    Et_disp_o_dn =
   phi_shift(Et=Et_disp_o_dn,t=t,t0=t0,dphi=dphi,shift_wid=dn_wid)
    Et_disp_e_dn =
   phi_shift(Et=Et_disp_e_dn,t=t,t0=t0,dphi=dphi,shift_wid=dn_wid)
    Ef_disp_o_dn = fft(Et_disp_o_dn)
    Ef_disp_e_dn = fft(Et_disp_e_dn)
```



```
    deg = 1.73
    Ef_disp_o_dn_e = apply_dispersion(Ef=Ef_disp_o_dn, f=f, n0=n0_bbo_e

   ,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=(d_bbo+dn_bbo))
    Ef_disp_e_dn_o = apply_dispersion(Ef=Ef_disp_e_dn, f=f, n0=n0_bbo_o

   ,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=(d_bbo+dn_bbo))
    Et_disp_o_dn_e = (ifft(Ef_disp_o_dn_e))
    Et_disp_e_dn_o = (ifft(Ef_disp_e_dn_o))
    ####################################
    if mode == 'polarisation':
        a, Et_int = jones(Et_disp_o_dn_e, Et_disp_e_dn_o, angle_pol)
        Ef_int = fft(Et_int)
        df = {'Ef':Ef,
          'Ef_disp':Ef_disp,
          'Ef_disp_o':Ef_disp_o,
          'Ef_disp_e':Ef_disp_e,
          'Ef_disp_o_dn':Ef_disp_o_dn,
          'Ef_disp_e_dn':Ef_disp_e_dn,
          'Ef_disp_o_dn_e':Ef_disp_o_dn_e,
          'Ef_disp_e_dn_o':Ef_disp_e_dn_o,
          'Ef_int':Ef_int,
          'Et':Et,
          'Et_disp':Et_disp,
          'Et_disp_o':Et_disp_o,
          'Et_disp_e':Et_disp_e,
          'Et_disp_o_dn':Et_disp_o_dn,
          'Et_disp_e_dn':Et_disp_e_dn,
          'Et_disp_o_dn_e':Et_disp_o_dn_e,
          'Et_disp_e_dn_o':Et_disp_e_dn_o,
          'Et_int':Et_int,
          'f':f,
          't':t}
        return(df)
    if mode == 'interferometric':
        Et_disp_e_dn_o =
   phi_shift(Et=Et_disp_e_dn_o,t=t,t0=t[100],dphi=d_phi_bbo,shift_wid=20e-15)
        Ef_disp_e_dn_o = fft(Et_disp_e_dn_o)
        Ef_int = (1/np.sqrt(2))*Ef_disp_e_dn_o + (1/np.sqrt(2))*Ef_disp_o_dn_e
        Et_int = ifft(Ef_int)
    #Ef_int = 1
    #Et_int = 1
        df = {'Ef':Ef,
              'Ef_disp':Ef_disp,
              'Ef_disp_o':Ef_disp_o,
              'Ef_disp_e':Ef_disp_e,
              'Ef_disp_o_dn':Ef_disp_o_dn,
```

```
                'Ef_disp_e_dn':Ef_disp_e_dn,
                'Ef_disp_o_dn_e':Ef_disp_o_dn_e,
                'Ef_disp_e_dn_o':Ef_disp_e_dn_o,
                'Ef_int':Ef_int,
                'Et':Et,
                'Et_disp':Et_disp,
                'Et_disp_o':Et_disp_o,
                'Et_disp_e':Et_disp_e,
                'Et_disp_o_dn':Et_disp_o_dn,
                'Et_disp_e_dn':Et_disp_e_dn,
                'Et_disp_o_dn_e':Et_disp_o_dn_e,
                'Et_disp_e_dn_o':Et_disp_e_dn_o,
                'Et_int':Et_int,
                'f':f,
                't':t}
        return(df)
```

1 Simulation Chapter

```
[ ]: f = np.linspace(-1e16,1e16,2**21)
     pix = np.arange(1280)
     w = v_to_l(f)
     pp_fraction = 1#1/
       gaussian_fraction(x_inf=[-1000,1000],x1=-30,x2=30,amp=1,cen=0,wid=180)
     pol_abs = 1#1/fit_pol_trans(400)
     nd_filter = 1/.005
     scaling = pp_fraction*pol_abs*nd_filter

     df = sim_tt(dn=0,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=-1200e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')
```



```
[ ]: t_min = np.where(df['t'] < -500e-15)[0][-1]
     t_max = np.where(df['t'] > 500e-15)[0][0]
     fig, ax = plt.subplots(2)
     fig.subplots_adjust(hspace=0)
     fig.subplots_adjust(wspace=0)
     ax[0].
       plot(convert_l_plot(df['f'],abs(df['Ef'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['Ef'])**
     ax[0].
       plot(convert_l_plot(df['f'],abs(df['Ef_disp'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['Ef
     ax[0].set_xlim(300,500)
     ax[0].set_ylabel(r'|E($\lambda$)|$^2$ /a.u.')
     ax[0].set_xlabel('wavelength /nm')
     ax[0].xaxis.set_label_position('top')
     ax[0].xaxis.tick_top()
     ax[1].plot(df['t']*1e15,df['Et']*1e4)
     ax[1].plot(df['t']*1e15,df['Et_disp']*1e4,'C4')
     ax[1].set_xlim(-1000,1000)
     ax[1].set_ylim(-39,39)
     ax[1].set_xlabel('time /fs')
     ax[1].set_ylabel('E(t) /a.u.')
     ax1 = ax[1].twinx()
     ax1.set_ylabel('$\lambda_      $ /nm', color = 'C2')
     ax1.plot(df['t'][t_min:t_max]*1e15,np.ones(len(df['t'][t_min:
       t_max]))*400,'C2--')
     ax1.plot(df['t'][t_min:t_max]*1e15,v_to_l(f_inst(df['Et_disp'],df['t'])[t_min:
       t_max])*1e9,'C2--')
     ax1.tick_params(axis='y')
     ax1.set_ylim(200,500)
     axins = inset_axes(ax[1], width=1.3, height=0.9)
     #plt.setp(axins.get_xticklabels(), visible=False)
     plt.setp(axins.get_yticklabels(), visible=False)
     axins.plot(df['t']*1e15,df['Et']*1e4)
     axins.plot(df['t']*1e15,df['Et_disp']*1e4,'C4')
     axins.set_xlim(-20,20)
     ax[0].text(.03,.85,'a)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     ax[1].text(.03,.85,'b)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
       tt_sim_1.pdf',bbox_inches='tight')
```



```
[ ]: fig, ax = plt.subplots(2)
     fig.subplots_adjust(hspace=0)
     fig.subplots_adjust(wspace=0)
     #ax[0].
       plot(convert_l_plot(df['f'],abs(df['Ef_disp_e'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['
     ax[0].set_xlim(300,500)
     ax[0].set_ylabel(r'|E($\lambda$)|$^2$ /a.u.')
     ax[0].set_xlabel('wavelength /nm')
     ax[0].xaxis.set_label_position('top')
     ax[0].xaxis.tick_top()
     ax[1].plot(df['t']*1e15,df['Et_disp_o']*1e4,'C4')
     ax[1].plot(df['t']*1e15,df['Et_disp_e']*1e4,'C0')
     ax[1].set_xlim(-5000,5000)
     ax[1].set_ylim(-7.9,7.9)
     ax[1].set_xlabel('time /fs')
     ax[1].set_ylabel('E(t) /a.u.')
     ax[0].text(.03,.85,'b)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     ax[1].text(.03,.85,'c)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
       tt_sim_2.pdf',bbox_inches='tight')
```



```
[ ]: fig, ax = plt.subplots(2, figsize=(3.2,2.4))
     fig.subplots_adjust(hspace=0)
     fig.subplots_adjust(wspace=0)
     #ax[0].
       plot(convert_l_plot(df['f'],abs(df['Ef_disp_o'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['
     ax[0].
       plot(convert_l_plot(df['f'],abs(df['Ef_disp_e'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['
     ax[0].set_xlim(300,500)
     #ax[0].set_ylabel('|E($\omega$)|$^2$ /a.u.')
     ax[0].set_xlabel('wavelength /nm')
     ax[0].xaxis.set_label_position('top')
     ax[0].xaxis.tick_top()
     ax[1].plot(df['t']*1e15,df['Et_disp_o']*1e4,'C4')
     ax[1].plot(df['t']*1e15,df['Et_disp_e']*1e4,'C0')
     ax[1].set_xticks([-4000,0,4000])
     ax[1].set_xlim(-5000,5000)
     ax[1].set_ylim(-7.9,7.9)
     ax[1].set_xlabel('time /fs')
     #ax[1].set_ylabel('E(t) /a.u.')
     fig.text(-0.01, 0.75, '|E($\lambda$)|$^2$ /a.u.', ha='center', va='center',
       rotation='vertical')
     fig.text(-0.01, 0.3, 'E(t) /a.u.', ha='center', va='center',
       rotation='vertical')
     ax[0].text(.05,.72,'b)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
```



```
     ax[1].text(.05,.72,'c)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
       tt_sim_2_small.pdf',bbox_inches='tight')

[ ]: offset = 0
     tmin = -5000 + offset
     tmax = 5000 + offset
     t_ax = np.where((df['t']*1e15 > tmin)&(df['t']*1e15<tmax))[0]
     t = df['t'][t_ax]*1e15 - offset

     p2, p1 = cen_time(df['Et_disp_e'],df['Et_disp_o'])
     p1 = p1[t_ax]
     p2 = p2[t_ax]

     fig = plt.figure(figsize=(3.2,2.4))
     ax = fig.add_subplot(111, projection='3d')
     ax.xaxis.labelpad=8
     ax.yaxis.labelpad=-12
     ax.zaxis.labelpad=-12




     ax.plot(t,p1,p2,'C2')




     plt.setp(ax.get_yticklabels(), visible=False)
     plt.setp(ax.get_zticklabels(), visible=False)
     ax.set_xticks([-4000,0,4000])

     ax.set_xlim(-5000,5000)
     ax.set_ylim(-.0008,.0004)
     ax.set_zlim(-.0005,.0005)
     ax.view_init(elev=45, azim=135)
     ax.set_xlabel('time /fs')
     ax.set_ylabel(r'E$_x$(t) /a.u.')
     ax.set_zlabel(r'E$_y$(t) /a.u.')

     ax2 = plt.axes([0,0,1,1])
     ip = InsetPosition(ax, [0.6,0.55,0.3,0.3*1.3333])
     ax2.set_axes_locator(ip)
     ax2.plot(p1,p2,'C2')
     plt.setp(ax2.get_xticklabels(), visible=False)
     plt.setp(ax2.get_yticklabels(), visible=False)
     ax2.set_xlabel(r'E$_x$(t)')
```

```
    ax2.set_ylabel(r'E$_y$(t)')
    #axins.
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[0])
    #axins.
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[1],'C4')
    #axins.set_xlim(-5,5)

    ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    # make the grid lines transparent
    ax.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.invert_xaxis()

    ax.text(0,-.002,.00575,'a)',transform=ax.transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    plt.subplots_adjust(left=0.05, right=.98, top=1, bottom=.15)
    plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_2_3d_small.pdf')

[ ]: fig, ax = plt.subplots(2)
    fig.subplots_adjust(hspace=0)
    fig.subplots_adjust(wspace=0)
    ax[0].
      plot(convert_l_plot(df['f'],abs(df['Ef_disp_o'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['
    ax[0].
      plot(convert_l_plot(df['f'],abs(df['Ef_disp_e'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['
    ax[0].set_xlim(300,500)
    ax[0].set_ylabel('|E($\lambda$)|$^2$ /a.u.')
    ax[0].set_xlabel('wavelength /nm')
    ax[0].xaxis.set_label_position('top')
    ax[0].xaxis.tick_top()
    ax[1].
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[0])
    ax[1].
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[1],'C4--')
    ax[1].set_xlim(-5000,5000)
    ax[1].set_ylim(-7.9,7.9)
    ax[1].set_xlabel('time /fs')
    ax[1].set_ylabel('E(t) /a.u.')
    axins = inset_axes(ax[1], width=1.3, height=0.9)
    #plt.setp(axins.get_xticklabels(), visible=False)
    plt.setp(axins.get_yticklabels(), visible=False)
```



```
    axins.
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[0])
    axins.
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[1],'C4--')
    axins.set_xlim(-5,5)
    ax[0].text(.03,.85,'a)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    ax[1].text(.03,.85,'b)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    #plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_3.pdf',bbox_inches='tight')

[ ]: t_off = 9050
    fig, ax = plt.subplots(2,figsize=(3.2,2.4))
    fig.subplots_adjust(hspace=0)
    fig.subplots_adjust(wspace=0)
    #ax[0].
      plot(convert_l_plot(df['f'],abs(df['Ef_disp_o'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['
    ax[0].
      plot(convert_l_plot(df['f'],abs(df['Ef_disp_e'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['
    ax[0].set_xlim(300,500)
    #ax[0].set_ylabel('|E($\omega$)|$^2$ /a.u.')
    ax[0].set_xlabel('wavelength /nm')
    ax[0].xaxis.set_label_position('top')
    ax[0].xaxis.tick_top()
    ax[1].
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[0])
    ax[1].
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[1],'C4--')
    ax[1].set_xticks([-4000,0,4000])
    ax[1].set_xlim(-5000,5000)
    ax[1].set_ylim(-7.9,7.9)
    ax[1].set_xlabel('time /fs')
    #ax[1].set_ylabel('E(t) /a.u.')
    axins = inset_axes(ax[1], width=1.4/2, height=1/2)
    #plt.setp(axins.get_xticklabels(), visible=False)
    plt.setp(axins.get_yticklabels(), visible=False)
    axins.
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[0])
    axins.
      plot(df['t']*1e15,cen_time(df['Et_disp_o_dn_e']*1e4,df['Et_disp_e_dn_o']*1e4)[1],'C4--')
    axins.set_xlim(-5,5)
    fig.text(-0.01, 0.75, '|E($\lambda$)|$^2$ /a.u.', ha='center', va='center',
      rotation='vertical')
    fig.text(-0.01, 0.3, 'E(t) /a.u.', ha='center', va='center',
      rotation='vertical')
```



```
    ax[0].text(.05,.72,'b)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    ax[1].text(.05,.72,'c)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_3_small.pdf',bbox_inches='tight')

[ ]: tmin = -5000
    tmax = 5000
    t_full = df['t']*1e15
    t_ax = np.where((t_full>tmin)&(t_full<tmax))[0]
    t = t_full[t_ax]
    p1,p2 = cen_time(df['Et_disp_o_dn_e'],df['Et_disp_e_dn_o'])
    p1 = p1[t_ax]
    p2 = p2[t_ax]

    p1pos = p1[np.where(p1>0)[0]]
    tp1pos = t[np.where(p1>0)[0]]
    p1neg = p1[np.where(p1<0)[0]]
    tp1neg = t[np.where(p1<0)[0]]
    p2pos = p2[np.where(p2>0)[0]]
    tp2pos = t[np.where(p2>0)[0]]
    p2neg = p2[np.where(p2<0)[0]]
    tp2neg = t[np.where(p2<0)[0]]

    fig = plt.figure(figsize=(3.2,2.4))
    ax = fig.add_subplot(111, projection='3d')
    ax.xaxis.labelpad=8
    ax.yaxis.labelpad=-12
    ax.zaxis.labelpad=-12

    ax.plot(tp2neg,p1neg,p2neg,'C2')
    ax.plot(tp2neg,p1neg*0,p2neg,'C4', alpha = .5)
    ax.plot(tp1neg,p1neg,0,'C0', alpha = .5)
    ax.plot(tp2pos,p1pos*0,p2pos,'C4', alpha = .5)
    ax.plot(tp1pos,p1pos,0,'C0', alpha = .5)
    ax.plot(tp2pos,p1pos,p2pos,'C2')


    ax2 = plt.axes([0,0,1,1])
    ip = InsetPosition(ax, [0.6,0.5,0.3,0.3*1.333])
    ax2.set_axes_locator(ip)
    ax2.plot(p1,p2*-1,'C2')
    ax2.plot(np.zeros(len(p2)),p2,'C4',alpha=.5)
    ax2.plot(p1,np.zeros(len(p2)),'C0',alpha=.5)
    #ax2.plot(np.linspace(-.0004,.0004,10),np.linspace(-.0004,.0004,10),'--')
    plt.setp(ax2.get_xticklabels(), visible=False)
```



```
    plt.setp(ax2.get_yticklabels(), visible=False)
    ax2.set_xlabel(r'E$_x$(t)')
    ax2.set_ylabel(r'E$_y$(t)')


    plt.setp(ax.get_yticklabels(), visible=False)
    plt.setp(ax.get_zticklabels(), visible=False)
    ax.set_xticks([-4000,0,4000])

    ax.set_xlim(-5000,5000)
    ax.set_ylim(-.0008,.0004)
    ax.set_zlim(-.0005,.0005)
    ax.view_init(elev=45, azim=135)
    ax.set_xlabel('time /fs')
    ax.set_ylabel(r'E$_x$(t) /a.u.')
    ax.set_zlabel(r'E$_y$(t) /a.u.')

    ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    # make the grid lines transparent
    ax.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.invert_xaxis()

    ax.text(0,-.002,.00575,'a)',transform=ax.transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    plt.subplots_adjust(left=0.05, right=.98, top=1, bottom=.15)
    plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_3_3d_small.pdf')

[ ]: dn=0
    dn_wid=20e-15
    d_absorption = 0
    t0=0
    l0=400e-9
    dl0=50e-9
    d_samp=50e-6
    amp=1
    d_disp=.05
    d_phi_bbo=.0
    angle_pol = -45
    d_bbo = .005
    dn_bbo=0
    num=2**21
    f_range = 1e16
```

```
num = (2**16)

f = np.linspace(-f_range,f_range,num)
w = v_to_w(f)
df = 1/(f[1] - f[0])
t = np.linspace(-df/2, df/2, num)
f0 = l_to_v(l0)
w0 = v_to_w(f0)
df0 = l_to_v(l0-dl0/2) - l_to_v(l0+dl0/2)
E0 = amp
Ef = amp*np.exp(-(f-f0)**2/(2*(df0/2.3548)**2))
l = np.linspace(.25,1.2,10000)



n_sio2 = sellmaier_sio2(l)
n_bbo_o = sellmaier_bbo_o(l)
n_bbo_e = sellmaier_bbo_e(l)
n0_bbo_o = sellmaier_bbo_o(l0*1e6)
n0_bbo_e = sellmaier_bbo_e(l0*1e6)
n0_sio2 = sellmaier_sio2(l0*1e6)
dn1,dn2,dn3,gvd_sio2,tod_sio2,gv_sio2 = calc_disp(n_sio2,l,l0*1e6,n0=n0_sio2)
dn1,dn2,dn3,gvd_bbo_o,tod_bbo_o,gv_bbo_o =
  calc_disp(n_bbo_o,l,l0*1e6,n0=n0_bbo_o)
dn1,dn2,dn3,gvd_bbo_e,tod_bbo_e,gv_bbo_e =
  calc_disp(n_bbo_e,l,l0*1e6,n0=n0_bbo_e)
Ef_disp = apply_dispersion(Ef=Ef, f=f,
  n0=n0_sio2,GVD=gvd_sio2,TOD=tod_sio2,z=d_disp)

Et = ifft(Ef)
Et_disp = ifft(Ef_disp)
Et_disp = cen_time(Et_disp)
Ef_disp_o = apply_dispersion(Ef=Ef_disp, f=f,
  n0=n0_bbo_o,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=d_bbo)
Ef_disp_e = apply_dispersion(Ef=Ef_disp, f=f,
  n0=n0_bbo_e,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=d_bbo)
Et_disp_o = ifft(Ef_disp_o)
Et_disp_e = ifft(Ef_disp_e)
Et_disp_o,Et_disp_e = cen_time(Et1=Et_disp_o, Et2=Et_disp_e)
#t0 = 5*750e-15
dphi = d_phi(dn=dn,l=l0,d=d_samp)
#print(dphi)
Et_disp_o_dn = d_abs(Et=Et_disp_o,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
Et_disp_e_dn = d_abs(Et=Et_disp_e,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
Et_disp_o_dn = phi_shift(Et=Et_disp_o_dn,t=t,t0=t0,dphi=dphi,shift_wid=dn_wid)
Et_disp_e_dn = phi_shift(Et=Et_disp_e_dn,t=t,t0=t0,dphi=dphi,shift_wid=dn_wid)
Ef_disp_o_dn = fft(Et_disp_o_dn)
```



```
Ef_disp_e_dn = fft(Et_disp_e_dn)



dn_bbo = d_rot(d_bbo, np.linspace(0,5,2001))
specs = []
for i in dn_bbo:wow
    Ef_disp_o_dn_e = apply_dispersion(Ef=Ef_disp_o_dn, f=f, n0=n0_bbo_e
                              ,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=(d_bbo+i))
    Ef_disp_e_dn_o = apply_dispersion(Ef=Ef_disp_e_dn, f=f, n0=n0_bbo_o
                              ,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=(d_bbo+i))
    Et_disp_o_dn_e = (ifft(Ef_disp_o_dn_e))
    Et_disp_e_dn_o = (ifft(Ef_disp_e_dn_o))
    ####################################

    a, Et_int = jones(Et_disp_o_dn_e, Et_disp_e_dn_o, -45)
    Ef_int = fft(Et_int)
    specs.append(abs(Ef_int)**2)
specs = np.array(specs)
specs_ex = np.vstack((np.flip(specs,axis=0),specs))
angles = np.append(-np.flip(d_rot_inv(d_bbo,dn_bbo)),d_rot_inv(d_bbo,dn_bbo))



plt.figure()
plt.title(specs_ex.max())
for i in range(len(specs_ex)):
    plt.plot(f, specs_ex[i])
plt.xlim(650e12,850e12)
#plt.xlim(748e12,751e12)
plt.axvline((constants.c/400e-9))

f_vis = np.where((f > 650e12) & (f < 900e12))[0]

fig = plt.figure()
ax1 = fig.add_subplot(111)
ax2 = ax1.twinx()
ax1.pcolormesh(v_to_l(f[f_vis])*1e9, angles, specs_ex[:,f_vis], cmap='magma',
  norm=matplotlib.colors.LogNorm(),vmin=1e-2,vmax=2)
ax1.set_xlim(340,460)
#ax2.plot(np.zeros(101),d_rot(.01, np.linspace(0,10,101))/lam2)
#ax2.set_yticks([100,80,50])
#ax2.set_yticklabels(['0','1','2'])
```



```
[ ]: specs400 = specs_ex[:,abs(v_to_l(f)-400e-9).argmin()]
     shift_full = find_peaks(-specs400)[0]
     shift_half = find_peaks(specs400)[0]
     shift_quarter = find_peaks(-abs(specs400-1))[0]

     plt.figure()
     plt.plot(specs400)
     for i in shift_full:
         plt.axvline(i,color='C2')
     for i in shift_half:
         plt.axvline(i,color='C3')
     for i in shift_quarter:
         plt.axvline(i,color='C5')

[ ]: dn=0
     dn_wid=20e-15
     d_absorption = 0
     t0=0
     l0=400e-9
     dl0=50e-9
     d_samp=50e-6
     amp=1
     d_disp=.05
     d_phi_bbo=.0
     angle_pol = -45
     d_bbo = .005
     dn_bbo=0
     num=2**21
     f_range = 1e16
     num = (2**16)

     f = np.linspace(-f_range,f_range,num)
     w = v_to_w(f)
     df = 1/(f[1] - f[0])
     t = np.linspace(-df/2, df/2, num)
     f0 = l_to_v(l0)
     w0 = v_to_w(f0)
     df0 = l_to_v(l0-dl0/2) - l_to_v(l0+dl0/2)
     E0 = amp
     Ef = amp*np.exp(-(f-f0)**2/(2*(df0/2.3548)**2))
     l = np.linspace(.25,1.2,10000)



     n_sio2 = sellmaier_sio2(l)
     n_bbo_o = sellmaier_bbo_o(l)
     n_bbo_e = sellmaier_bbo_e(l)
     n0_bbo_o = sellmaier_bbo_o(l0*1e6)
     n0_bbo_e = sellmaier_bbo_e(l0*1e6)
     n0_sio2 = sellmaier_sio2(l0*1e6)
```



```
dn1,dn2,dn3,gvd_sio2,tod_sio2,gv_sio2 = calc_disp(n_sio2,l,l0*1e6,n0=n0_sio2)
dn1,dn2,dn3,gvd_bbo_o,tod_bbo_o,gv_bbo_o =
  calc_disp(n_bbo_o,l,l0*1e6,n0=n0_bbo_o)
dn1,dn2,dn3,gvd_bbo_e,tod_bbo_e,gv_bbo_e =
  calc_disp(n_bbo_e,l,l0*1e6,n0=n0_bbo_e)
Ef_disp = apply_dispersion(Ef=Ef, f=f,
  n0=n0_sio2,GVD=gvd_sio2,TOD=tod_sio2,z=d_disp)

Et = ifft(Ef)
Et_disp = ifft(Ef_disp)
Et_disp = cen_time(Et_disp)
Ef_disp_o = apply_dispersion(Ef=Ef_disp, f=f,
  n0=n0_bbo_o,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=d_bbo)
Ef_disp_e = apply_dispersion(Ef=Ef_disp, f=f,
  n0=n0_bbo_e,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=d_bbo)
Et_disp_o = ifft(Ef_disp_o)
Et_disp_e = ifft(Ef_disp_e)
Et_disp_o,Et_disp_e = cen_time(Et1=Et_disp_o, Et2=Et_disp_e)
#t0 = 5*750e-15
dphi = d_phi(dn=dn,l=l0,d=d_samp)
#print(dphi)
Et_disp_o_dn = d_abs(Et=Et_disp_o,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
Et_disp_e_dn = d_abs(Et=Et_disp_e,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
Et_disp_o_dn = phi_shift(Et=Et_disp_o_dn,t=t,t0=t0,dphi=dphi,shift_wid=dn_wid)
Et_disp_e_dn = phi_shift(Et=Et_disp_e_dn,t=t,t0=t0,dphi=dphi,shift_wid=dn_wid)
Ef_disp_o_dn = fft(Et_disp_o_dn)
Ef_disp_e_dn = fft(Et_disp_e_dn)

dn_bbo_large = d_rot(d_bbo, np.linspace(0,30,2001))
specs_large = []
for i in dn_bbo_large:
    Ef_disp_o_dn_e = apply_dispersion(Ef=Ef_disp_o_dn, f=f, n0=n0_bbo_e
                              ,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=(d_bbo+i))
    Ef_disp_e_dn_o = apply_dispersion(Ef=Ef_disp_e_dn, f=f, n0=n0_bbo_o
                              ,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=(d_bbo+i))
    Et_disp_o_dn_e = (ifft(Ef_disp_o_dn_e))
    Et_disp_e_dn_o = (ifft(Ef_disp_e_dn_o))
    ####################################

    a, Et_int = jones(Et_disp_o_dn_e, Et_disp_e_dn_o, -45)
    Ef_int = fft(Et_int)
    specs_large.append(abs(Ef_int)**2)
specs_large = np.array(specs_large)
specs_ex_large = np.vstack((np.flip(specs_large,axis=0),specs_large))
angles_large = np.append(-np.
  flip(d_rot_inv(d_bbo,dn_bbo_large)),d_rot_inv(d_bbo,dn_bbo_large))
```

```
#plt.figure()
#plt.title(specs_ex.max())
#for i in range(len(specs_ex_large)):
#    plt.plot(f, specs_ex_large[i])
#plt.xlim(650e12,850e12)
##plt.xlim(748e12,751e12)
#plt.axvline((constants.c/400e-9))

f_vis = np.where((f > 650e12) & (f < 900e12))[0]

fig = plt.figure()
ax1 = fig.add_subplot(111)
ax2 = ax1.twinx()
ax1.pcolormesh(v_to_l(f[f_vis])*1e9, angles_large, specs_ex_large[:,f_vis],
  cmap='magma', norm=matplotlib.colors.LogNorm(),vmin=1e-2,vmax=2)
ax1.set_xlim(340,460)
#ax2.plot(np.zeros(101),d_rot(.01, np.linspace(0,10,101))/lam2)
#ax2.set_yticks([100,80,50])
#ax2.set_yticklabels(['0','1','2'])
```

[ ]:
```
specs400_large = specs_ex_large[:,abs(v_to_l(f)-400e-9).argmin()]
shift_full_large = find_peaks(-specs400_large)[0]
shift_half_large = find_peaks(specs400_large)[0]
shift_quarter_large = find_peaks(-abs(specs400_large-1))[0]

plt.figure()
plt.plot(specs400_large)
for i in shift_full_large:
    plt.axvline(i,color='C2')
#for i in shift_half_large:
#    plt.axvline(i,color='C3')
#for i in shift_quarter_large:
#    plt.axvline(i,color='C5')
```

[ ]:
```
specs_400 = specs_ex[:,abs((v_to_l(f)-400e-9)).argmin()]
specs_400_min = find_peaks(-specs_400)[0]
specs_400_max = find_peaks(specs_400)[0]

fig = plt.figure(figsize=(7,4.2))
ax1 = fig.add_subplot(121)
ax2 = ax1.twinx()
ax3 = fig.add_subplot(122)
#plt.gca().invert_yaxis()
fig.subplots_adjust(hspace=0)
fig.subplots_adjust(wspace=.5)
```



```
im = ax1.pcolormesh(v_to_l(f[f_vis])*1e9, angles, specs_ex[:,f_vis],
  cmap='magma', norm=matplotlib.colors.
  LogNorm(),vmin=1e-4,vmax=2,rasterized=True)
ax_cbar = fig.add_axes([.1, .79, .31, 0.05])
plt.colorbar(im, cax=ax_cbar, orientation='horizontal', ticklocation = 'top')
fig.text(0.26, 0.99, 'spectral amplitude /a.u.', ha='center', va='center',
  rotation='horizontal')
ax1.set_xlim(340,460)
ax2.plot(np.zeros(4001),np.arange(4001))
#ax2.set_yticks([216, 543, 969, 1271, 2000, 2730, 3032, 3458, 3785])
ax2.set_yticks([216,543,969,1271,1484,2000,2517,2730,3032,3458,3785])
#ax2.set_yticks([73, 371, 738, 3263, 3630, 3928], minor=True)
ax2.set_ylim(0,4002)
#ax2.set_yticklabels([3*360,2*360,360,180,0,180,360,2*360,3*360])
ax2.set_yticklabels([1080,720,360,180,90,0,90,180,360,720,1080])
ax1.set_xlabel('wavelength /nm')
ax1.set_ylabel('rotation /deg')
ax2.set_ylabel('phase shift /deg')
ax2.get_yticklabels()[0].set_color('C5')
ax2.get_yticklabels()[1].set_color('C4')
ax2.get_yticklabels()[2].set_color('C3')
ax2.get_yticklabels()[3].set_color('C2')
ax2.get_yticklabels()[4].set_color('C1')
ax2.get_yticklabels()[5].set_color('C0')
ax2.get_yticklabels()[6].set_color('C1')
ax2.get_yticklabels()[7].set_color('C2')
ax2.get_yticklabels()[8].set_color('C3')
ax2.get_yticklabels()[9].set_color('C4')
ax2.get_yticklabels()[10].set_color('C5')


ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_full[3]],'C0')
ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_quarter[7]],'C1')
ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_half[3]],'C2')
ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_full[2]],'C3')
ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_full[1]],'C4')
ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_full[0]],'C5')
#ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][2000],'C0')
#ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][2730],'C1')
#ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][3032],'C3')
#ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][3458],'C4')
#ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][3785],'C5')
ax3.yaxis.tick_right()

ax3.set_xlim(340,460)
ax3.set_ylabel('spectral amp. /a.u')
ax3.yaxis.set_label_position('right')
```



```
ax3.set_xlabel('wavelength /nm')

plt.subplots_adjust(left=0.1, right=.87, top=.78, bottom=.1)
#plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_5_new.pdf',bbox_inches='tight')
```

[ ]:
```
specs_400 = specs_ex[:,abs((v_to_l(f)-400e-9)).argmin()]
specs_400_min = find_peaks(-specs_400)[0]
specs_400_max = find_peaks(specs_400)[0]

fig = plt.figure(figsize=(7,4.2))
ax1 = fig.add_subplot(111)
ax2 = ax1.twinx()


im = ax1.pcolormesh(v_to_l(f[f_vis])*1e9, angles, specs_ex[:,f_vis],
  cmap='magma', norm=matplotlib.colors.
  LogNorm(),vmin=1e-5,vmax=2,rasterized=True)
ax_cbar = fig.add_axes([.1, .79, .77, 0.05])
plt.colorbar(im, cax=ax_cbar, orientation='horizontal', ticklocation = 'top')
fig.text(0.5, 0.95, 'spectral amplitude /a.u.', ha='center', va='center',
  rotation='horizontal')
ax1.set_xlim(340,460)
ax2.plot(np.zeros(4001),np.arange(4001))
#ax2.set_yticks([216, 543, 969, 1271, 2000, 2730, 3032, 3458, 3785])
ax2.set_yticks([216,543,969,1271,1484,2000,2517,2730,3032,3458,3785])
#ax2.set_yticks([73, 371, 738, 3263, 3630, 3928], minor=True)
ax2.set_ylim(0,4002)
#ax2.set_yticklabels([3*360,2*360,360,180,0,180,360,2*360,3*360])
ax2.set_yticklabels([1080,720,360,180,90,0,90,180,360,720,1080])
ax1.set_xlabel('wavelength /nm')
ax1.set_ylabel('crystal rotation /deg')
ax2.set_ylabel('phase shift /deg')
#ax2.get_yticklabels()[0].set_color('C5')
#ax2.get_yticklabels()[1].set_color('C4')
#ax2.get_yticklabels()[2].set_color('C3')
#ax2.get_yticklabels()[3].set_color('C2')
#ax2.get_yticklabels()[4].set_color('C1')
#ax2.get_yticklabels()[5].set_color('C0')
#ax2.get_yticklabels()[6].set_color('C1')
#ax2.get_yticklabels()[7].set_color('C2')
#ax2.get_yticklabels()[8].set_color('C3')
#ax2.get_yticklabels()[9].set_color('C4')
#ax2.get_yticklabels()[10].set_color('C5')


plt.subplots_adjust(left=0.1, right=.87, top=.78, bottom=.1)
```



```
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_5_new.pdf',bbox_inches='tight')
```

[ ]:
```
f = np.linspace(-1e16,1e16,2**21)
pix = np.arange(1280)
w = v_to_l(f)
pp_fraction = 1#1/
  gaussian_fraction(x_inf=[-1000,1000],x1=-30,x2=30,amp=1,cen=0,wid=180)
pol_abs = 1#1/fit_pol_trans(400)
nd_filter = 1/.005
scaling = pp_fraction*pol_abs*nd_filter
df_0 = sim_tt(dn=0,
            d_absorption=.0,
            dn_wid=150e-15,
            t0=-1200e-15,
            l0=400e-9,
            dl0=50e-9,
            d_bbo = .005,
            dn_bbo=0,
            d_samp=50e-6,
            angle_pol = -45,
            amp=1,
            d_disp=.01,
            num=2**22,
            f_range = 1e16,
            shape = 'gaussian',
            f=f,
            mode = 'polarisation')

df_quarter = sim_tt(dn=0,
            d_absorption=.0,
            dn_wid=150e-15,
            t0=-1200e-15,
            l0=400e-9,
            dl0=50e-9,
            d_bbo = .005,
            dn_bbo=1.2675523442425007e-06,
            d_samp=50e-6,
            angle_pol = -45,
            amp=1,
            d_disp=.01,
            num=2**22,
            f_range = 1e16,
            shape = 'gaussian',
            f=f,
            mode = 'polarisation')
```

```
df_half = sim_tt(dn=0,
            d_absorption=.0,
            dn_wid=150e-15,
            t0=-1200e-15,
            l0=400e-9,
            dl0=50e-9,
            d_bbo = .005,
            dn_bbo=2.530538469296649e-06,
            d_samp=50e-6,
            angle_pol = -45,
            amp=1,
            d_disp=.01,
            num=2**22,
            f_range = 1e16,
            shape = 'gaussian',
            f=f,
            mode = 'polarisation')

df_full = sim_tt(dn=0,
            d_absorption=.0,
            dn_wid=150e-15,
            t0=-1200e-15,
            l0=400e-9,
            dl0=50e-9,
            d_bbo = .005,
            dn_bbo=5.063588963035923e-06,
            d_samp=50e-6,
            angle_pol = -45,
            amp=1,
            d_disp=.01,
            num=2**22,
            f_range = 1e16,
            shape = 'gaussian',
            f=f,
            mode = 'polarisation')

fig, ax = plt.subplots(2)
#ax[0].plot(df['f'],abs(df['Ef_disp'])**2)
#ax[0].plot(df['f'],abs(df['Ef'])**2,'C4--')
ax[0].set_title((abs(df_0['Ef_int'])**2).max())
ax[0].plot(df_0['f'],abs(df_0['Ef_int'])**2,'C4--')
ax[0].set_xlim(650e12,850e12)
#ax[0].set_ylim(-1000,.25e6)
ax[1].plot(df_0['t']*1e15,df_0['Et_disp_o'])
ax[1].plot(df_0['t']*1e15,df_0['Et_disp_e'])
ax[1].set_xlim(-5000,5000)
```



```
[ ]: tmin = -5000
     tmax = 5000
     t_0_full = df_0['t']*1e15
     t_0_ax = np.where((t_0_full>tmin)&(t_0_full<tmax))[0]
     t_0 = t_0_full[t_0_ax]
     p1_0,p2_0 = cen_time(df_0['Et_disp_o_dn_e'],df_0['Et_disp_e_dn_o'])
     p1_0 = p1_0[t_0_ax]
     p2_0 = p2_0[t_0_ax]

     t_quarter_full = df_quarter['t']*1e15
     t_quarter_ax = np.where((t_quarter_full>tmin)&(t_quarter_full<tmax))[0]
     t_quarter = t_quarter_full[t_quarter_ax]
     p1_quarter,p2_quarter =
        cen_time(df_quarter['Et_disp_o_dn_e'],df_quarter['Et_disp_e_dn_o'])
     p1_quarter = p1_quarter[t_quarter_ax]
     p2_quarter = p2_quarter[t_quarter_ax]

     t_half_full = df_half['t']*1e15
     t_half_ax = np.where((t_half_full>tmin)&(t_half_full<tmax))[0]
     t_half = t_half_full[t_half_ax]
     p1_half,p2_half = cen_time(df_half['Et_disp_o_dn_e'],df_half['Et_disp_e_dn_o'])
     p1_half = p1_half[t_half_ax]
     p2_half = p2_half[t_half_ax]

     t_full_full = df_full['t']*1e15
     t_full_ax = np.where((t_full_full>tmin)&(t_full_full<tmax))[0]
     t_full = t_full_full[t_full_ax]
     p1_full,p2_full = cen_time(df_full['Et_disp_o_dn_e'],df_full['Et_disp_e_dn_o'])
     p1_full = p1_full[t_full_ax]
     p2_full = p2_full[t_full_ax]

     fig = plt.figure(figsize=(6.4,3.2))
     fig.subplots_adjust(hspace=.05)
     fig.subplots_adjust(wspace=.03)
     ax1 = fig.add_subplot(241, projection='3d')
     ax2 = fig.add_subplot(242, projection='3d')
     ax3 = fig.add_subplot(243, projection='3d')
     ax4 = fig.add_subplot(244, projection='3d')
     ax5 = fig.add_subplot(245, projection='3d')
     ax6 = fig.add_subplot(246, projection='3d')
     ax7 = fig.add_subplot(247, projection='3d')
     ax8 = fig.add_subplot(248, projection='3d')


     for i in [ax1,ax2,ax3,ax4,ax5,ax6,ax7,ax8]:
         i.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
         i.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
```



```
    i.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    i.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    i.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    i.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    plt.setp(i.get_xticklabels(), visible=False)
    plt.setp(i.get_yticklabels(), visible=False)
    plt.setp(i.get_zticklabels(), visible=False)


ax1.plot(t_0,p1_0,p2_0,'C0',rasterized=True)
ax1.invert_xaxis()
ax1.view_init(elev=45, azim=90)
ax1.set_ylim(-.0004,.0004)
ax1.set_zlim(-.0004,.0004)
ax1.set_title('0°')

ax2.plot(t_quarter,p1_quarter,p2_quarter, 'C1',rasterized=True)
ax2.invert_xaxis()
ax2.view_init(elev=45, azim=90)
ax2.set_ylim(-.0004,.0004)
ax2.set_zlim(-.0004,.0004)
ax2.set_title('90°')

ax3.plot(t_half,p1_half, p2_half,'C2',rasterized=True)
ax3.invert_xaxis()
ax3.view_init(elev=45, azim=90)
ax3.set_ylim(-.0004,.0004)
ax3.set_zlim(-.0004,.0004)
ax3.set_title('180°')

ax4.plot(t_full,p1_full,p2_full,'C3',rasterized=True)
ax4.invert_xaxis()
ax4.view_init(elev=45, azim=90)
ax4.set_ylim(-.0004,.0004)
ax4.set_zlim(-.0004,.0004)
ax4.set_title('360°')

ax5.plot(t_0,p1_0,p2_0,'C0',rasterized=True)
ax5.plot(t_full*0,np.linspace(-.0004,.0004,len(t_full)),-np.linspace(-.0004,.
   0004,len(t_full)),'C0--',rasterized=True)
ax5.invert_xaxis()
ax5.view_init(elev=0, azim=180)
ax5.set_ylim(-.0004,.0004)
ax5.set_zlim(-.0004,.0004)

ax6.plot(t_quarter,p1_quarter,p2_quarter,'C1',rasterized=True)
```



```
ax6.plot(t_full*0,np.linspace(-.0004,.0004,len(t_full)),-np.linspace(-.0004,.
   0004,len(t_full)),'C0--',rasterized=True)
ax6.invert_xaxis()
ax6.view_init(elev=0, azim=180)
ax6.set_ylim(-.0004,.0004)
ax6.set_zlim(-.0004,.0004)

ax7.plot(t_half,p1_half,p2_half,'C2',rasterized=True)
ax7.plot(t_full*0,np.linspace(-.0004,.0004,len(t_full)),-np.linspace(-.0004,.
   0004,len(t_full)),'C0--',rasterized=True)
ax7.invert_xaxis()
ax7.view_init(elev=0, azim=180)
ax7.set_ylim(-.0004,.0004)
ax7.set_zlim(-.0004,.0004)

ax8.plot(t_full,p1_full,p2_full,'C3',rasterized=True)
ax8.plot(t_full*0,np.linspace(-.0004,.0004,len(t_full)),-np.linspace(-.0004,.
   0004,len(t_full)),'C0--',rasterized=True)
ax8.invert_xaxis()
ax8.view_init(elev=0, azim=180)
ax8.set_ylim(-.0004,.0004)
ax8.set_zlim(-.0004,.0004)

for i in (ax1,ax2,ax3,ax4):
    i.set_xlabel('time')
    i.set_ylabel(r'E$_x$')
    i.set_zlabel(r'E$_y$')
    i.xaxis.labelpad=-13
    i.yaxis.labelpad=-10
    i.zaxis.labelpad=-6

for i in (ax5,ax6,ax7,ax8):
    #i.set_xlabel('time')
    i.set_ylabel(r'E$_x$')
    i.set_zlabel(r'E$_y$')
    i.xaxis.labelpad=-13
    i.yaxis.labelpad=-10
    i.zaxis.labelpad=-6
#ax9.set_subplots_adjust(left=0.00, right=.98, top=.95, bottom=.01)
plt.subplots_adjust(left=0.00, right=.98, top=.95, bottom=.01)
#plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
   tt_sim_5_2_1_new.pdf',bbox_inches='tight')

f = np.linspace(-f_range,f_range,num)
fig = plt.figure(figsize=(6.4,1.6))
fig.subplots_adjust(wspace=.4)
ax1 = fig.add_subplot(141)
```

```
ax2 = fig.add_subplot(142)
ax3 = fig.add_subplot(143)
ax4 = fig.add_subplot(144)

ax1.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_full[3]],'C0')
ax1.set_ylim(-.1,2.1)
ax1.set_xlim(340,460)
plt.setp(ax1.get_yticklabels(), visible=False)
#ax1.yaxis.tick_right()
ax1.set_yticks([0,1,2])
ax1.set_ylabel(r'|E($\lambda$)|$^2$')
ax1.set_xlabel('$\lambda$ /nm')

ax2.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_quarter[7]],'C1')
ax2.set_ylim(-.1,2.1)
ax2.set_xlim(340,460)
plt.setp(ax2.get_yticklabels(), visible=False)
#ax2.yaxis.tick_right()
ax2.set_yticks([0,1,2])
ax2.set_ylabel(r'|E($\lambda$)|$^2$')
ax2.set_xlabel('$\lambda$ /nm')

ax3.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_half[3]],'C2')
ax3.set_ylim(-.1,2.1)
ax3.set_xlim(340,460)
plt.setp(ax3.get_yticklabels(), visible=False)
#ax1.yaxis.tick_right()
ax3.set_yticks([0,1,2])
ax3.set_ylabel(r'|E($\lambda$)|$^2$')
ax3.set_xlabel('$\lambda$ /nm')

ax4.plot(v_to_l(f[f_vis])*1e9,specs_ex[:,f_vis][shift_full[2]],'C3')
ax4.set_ylim(-.1,2.1)
ax4.set_xlim(340,460)
plt.setp(ax4.get_yticklabels(), visible=False)
#ax1.yaxis.tick_right()
ax4.set_yticks([0,1,2])
ax4.set_ylabel(r'|E($\lambda$)|$^2$')
ax4.set_xlabel('$\lambda$ /nm')

plt.subplots_adjust(left=0.1, right=.9, top=1, bottom=.4)
#plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
   tt_sim_5_2_2_new.pdf',bbox_inches='tight')
```

[ ]:
```
f = np.linspace(-1e16,1e16,2**21)
pix = np.arange(1280)
w = v_to_l(f)
```



```
pp_fraction = 1#1/
   gaussian_fraction(x_inf=[-1000,1000],x1=-30,x2=30,amp=1,cen=0,wid=180)
pol_abs = 1#1/fit_pol_trans(400)
nd_filter = 1/.005
scaling = pp_fraction*pol_abs*nd_filter
#df5 = sim_tt(dn=0,
#            d_absorption=.0,
#            dn_wid=150e-15,
#            t0=-1200e-15,
#            l0=400e-9,
#            dl0=50e-9,
#            d_bbo = .005,
#            dn_bbo=d_rot(.005,5.75),
#            d_samp=50e-6,
#            angle_pol = -45,
#            amp=1,
#            d_disp=.01,
#            num=2**22,
#            f_range = 1e16,
#            shape = 'gaussian',
#            f=f,
#            mode = 'polarisation')

#df10 = sim_tt(dn=0,
#            d_absorption=.0,
#            dn_wid=150e-15,
#            t0=-1200e-15,
#            l0=400e-9,
#            dl0=50e-9,
#            d_bbo = .005,
#            dn_bbo=d_rot(.005,8.116),
#            d_samp=50e-6,
#            angle_pol = -45,
#            amp=1,
#            d_disp=.01,
#            num=2**22,
#            f_range = 1e16,
#            shape = 'gaussian',
#            f=f,
#            mode = 'polarisation')

#df15 = sim_tt(dn=0,
#            d_absorption=.0,
#            dn_wid=150e-15,
#            t0=-1200e-15,
#            l0=400e-9,
#            dl0=50e-9,
```



```
#            d_bbo = .005,
#            dn_bbo=d_rot(.005,9.245),
#            d_samp=50e-6,
#            angle_pol = -45,
#            amp=1,
#            d_disp=.01,
#            num=2**22,
#            f_range = 1e16,
#            shape = 'gaussian',
#            f=f,
#            mode = 'polarisation')

#df20 = sim_tt(dn=0,
#            d_absorption=.0,
#            dn_wid=150e-15,
#            t0=-1200e-15,
#            l0=400e-9,
#            dl0=50e-9,
#            d_bbo = .005,
#            dn_bbo=d_rot(.005,11.432),
#            d_samp=50e-6,
#            angle_pol = -45,
#            amp=1,
#            d_disp=.01,
#            num=2**22,
#            f_range = 1e16,
#            shape = 'gaussian',
#            f=f,
#            mode = 'polarisation')

#df25 = sim_tt(dn=0,
#            d_absorption=.0,
#            dn_wid=150e-15,
#            t0=-1200e-15,
#            l0=400e-9,
#            dl0=50e-9,
#            d_bbo = .005,
#            dn_bbo=d_rot(.005,12.756),
#            d_samp=50e-6,
#            angle_pol = -45,
#            amp=1,
#            d_disp=.01,
#            num=2**22,
#            f_range = 1e16,
#            shape = 'gaussian',
#            f=f,
#            mode = 'polarisation')
```



```
f_vis = np.where((df5['f'] > 650e12) & (df5['f'] < 900e12))[0]
fig = plt.figure(figsize=(6.4,2.4))
ax1 = fig.add_subplot(111)
ax2 = ax1.twinx()
ax1.plot(v_to_l(df5['f'][f_vis])*1e9,abs(df5['Ef_int'][f_vis])**2)
ax1.plot(v_to_l(df5['f'][f_vis])*1e9,abs(df10['Ef_int'][f_vis])**2+2)
ax1.plot(v_to_l(df5['f'][f_vis])*1e9,abs(df15['Ef_int'][f_vis])**2+4)
ax1.plot(v_to_l(df5['f'][f_vis])*1e9,abs(df20['Ef_int'][f_vis])**2+6)
ax1.plot(v_to_l(df5['f'][f_vis])*1e9,abs(df25['Ef_int'][f_vis])**2+8)
ax1.set_yticks([0,2,4,6,8])
ax1.set_xlim(340,460)
ax1.set_ylim(-.5,9.99)
ax1.set_xlabel('wavelength /nm')
ax1.set_ylabel(r'|E($\lambda$)|$^2$ /a.u.')

ax2.plot(np.zeros(5),np.linspace(-.5,1,5))
ax2.set_ylim(-.05,.995)
ax2.set_yticks([0,.2,.4,.6,.8])
ax2.set_yticklabels([5,10,15,20,25])
ax2.set_ylabel('phase shift /$x 360^\circ$')
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
   tt_sim_5_3_new.pdf',bbox_inches='tight')
```

[ ]:
```
f = np.linspace(-1e16,1e16,2**21)
pix = np.arange(1280)
w = v_to_l(f)
pp_fraction = 1#1/
   gaussian_fraction(x_inf=[-1000,1000],x1=-30,x2=30,amp=1,cen=0,wid=180)
pol_abs = 1#1/fit_pol_trans(400)
nd_filter = 1/.005
scaling = pp_fraction*pol_abs*nd_filter

df1 = sim_tt(dn=5e-5,
             d_absorption=.0,
             dn_wid=150e-15,
             t0=900e-15,
             l0=400e-9,
             dl0=50e-9,
             d_bbo = .005,
             d_phi_bbo=.0,
             d_samp=50e-6,
             angle_pol = -45,
             amp=1,
             d_disp=.01,
             num=2**22,
             f_range = 1e16,
```

```
            shape = 'gaussian',
            f=f,
            mode = 'polarisation')

    fig, ax = plt.subplots(2)
    #ax[0].plot(df['f'],abs(df['Ef_disp'])**2)
    #ax[0].plot(df['f'],abs(df['Ef'])**2,'C4--')
    ax[0].plot(df1['f'],abs(df1['Ef_int'])**2,'C4--')
    ax[0].set_xlim(650e12,850e12)
    #ax[0].set_ylim(-1000,.25e6)
    ax[1].plot(df1['t']*1e15,df1['Et_disp_o'])
    ax[1].plot(df1['t']*1e15,df1['Et_disp_e'])
    ax[1].set_xlim(-5000,5000)
```

[ ]:
```
    tmin = -5000
    tmax = 5000
    t_full = df1['t']*1e15
    t_ax = np.where((t_full>tmin)&(t_full<tmax))[0]
    t = t_full[t_ax]
    p1,p2 = cen_time(df1['Et_disp_o_dn_e'],df1['Et_disp_e_dn_o'])
    p1 = p1[t_ax]
    p2 = p2[t_ax]

    p1pos = p1[np.where(p1>0)[0]]
    tp1pos = t[np.where(p1>0)[0]]
    p1neg = p1[np.where(p1<0)[0]]
    tp1neg = t[np.where(p1<0)[0]]
    p2pos = p2[np.where(p2>0)[0]]
    tp2pos = t[np.where(p2>0)[0]]
    p2neg = p2[np.where(p2<0)[0]]
    tp2neg = t[np.where(p2<0)[0]]
    pposind = np.where((p1 > 0)&(p2 > 0))[0]
    pnegind = np.where((p1 < 0) & (p2 < 0))[0]

    fig = plt.figure(figsize=(3.2,2.4))
    ax = fig.add_subplot(111, projection='3d')
    ax.xaxis.labelpad=8
    ax.yaxis.labelpad=-12
    ax.zaxis.labelpad=-12


    ax.plot(tp2neg,p2neg*0,p2neg,'C4')

    ax.plot(tp1pos,p1pos,0,'C0')
    ax.plot(tp2pos,p2pos*0,p2pos,'C4')
    ax.plot(t[pposind],p1[pposind],p2[pposind],'C2')
    ax.plot(t[pnegind],p1[pnegind],p2[pnegind],'C2')
```



```
    ax.plot(tp1neg,p1neg,0,'C0')

    ax2 = plt.axes([0,0,1,1])
    ip = InsetPosition(ax, [0.6,0.5,0.3,0.3*1.333])
    ax2.set_axes_locator(ip)
    ax2.plot(p1,p2*-1,'C2')
    ax2.plot(np.zeros(len(p2)),p2,'C4')
    ax2.plot(p1,np.zeros(len(p2)),'C0')
    ax2.plot(np.linspace(-.0004,.0004,10),np.linspace(-.0004,.0004,10),'--')
    plt.setp(ax2.get_xticklabels(), visible=False)
    plt.setp(ax2.get_yticklabels(), visible=False)
    ax2.set_xlabel(r'E$_x$(t)')
    ax2.set_ylabel(r'E$_y$(t)')


    plt.setp(ax.get_yticklabels(), visible=False)
    plt.setp(ax.get_zticklabels(), visible=False)
    ax.set_xticks([-4000,0,4000])

    ax.set_xlim(-5000,5000)
    ax.set_ylim(-.0008,.0004)
    ax.set_zlim(-.0005,.0005)
    ax.view_init(elev=45, azim=135)
    ax.set_xlabel('time /fs')
    ax.set_ylabel(r'E$_x$(t) /a.u.')
    ax.set_zlabel(r'E$_y$(t) /a.u.')

    ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    # make the grid lines transparent
    ax.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.invert_xaxis()

    ax.text(0,-.002,.00575,'c)',transform=ax.transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    plt.subplots_adjust(left=0.05, right=.98, top=1, bottom=.15)
    #plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_3_3d_small.pdf')
```

[ ]:
```
    tmin = -5000
    tmax = 5000
    t_full = df1['t']*1e15
    t_ax = np.where((t_full>tmin)&(t_full<tmax))[0]
    t = t_full[t_ax]
```



```
    p1,p2 = cen_time(df1['Et_disp_o_dn_e'],df1['Et_disp_e_dn_o'])
    p1 = p1[t_ax]
    p2 = p2[t_ax]

    p1pos = p1[np.where(p1>0)[0]]
    tp1pos = t[np.where(p1>0)[0]]
    p1neg = p1[np.where(p1<0)[0]]
    tp1neg = t[np.where(p1<0)[0]]
    p2pos = p2[np.where(p2>0)[0]]
    tp2pos = t[np.where(p2>0)[0]]
    p2neg = p2[np.where(p2<0)[0]]
    tp2neg = t[np.where(p2<0)[0]]


    fig = plt.figure(figsize=(6.4,3.2))
    ax = fig.add_subplot(111, projection='3d')
    ax.xaxis.labelpad=8
    ax.yaxis.labelpad=-12
    ax.zaxis.labelpad=-12


    ax.plot(tp1pos,p1pos,0,'C0')

    ax.plot(tp1neg,p1neg,0,'C0')
    ax.plot(tp2pos,p2pos*0,p2pos,'C4')
    ax.plot(t,p1,p2,'C2')

    plt.setp(ax.get_yticklabels(), visible=False)
    plt.setp(ax.get_zticklabels(), visible=False)
    ax.set_xticks([-4000,0,4000])

    ax.set_xlim(-5000,5000)
    ax.set_ylim(-.0004,.0004)
    ax.set_zlim(-.0004,.0004)
    ax.view_init(elev=45, azim=90)
    ax.set_xlabel('time /fs')
    ax.set_ylabel(r'E$_x$(t) /a.u.')
    ax.set_zlabel(r'E$_y$(t) /a.u.')

    ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
    # make the grid lines transparent
    ax.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
    ax.invert_xaxis()
```



```
    plt.subplots_adjust(left=0.05, right=.98, top=1, bottom=.15)
    plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_3_3d_small.pdf')
```

[ ]:
```
    df = sim_tt(dn=1e-2,
                d_absorption=.0,
                dn_wid=150e-15,
                t0=-1200e-15,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .005,
                dn_bbo=2.530538469296649e-06,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.01,
                num=2**22,
                f_range = 1e16,
                shape = 'gaussian',
                f=f,
                mode = 'polarisation')
```

[ ]:
```
    fig = plt.figure()
    fig.subplots_adjust(hspace=0)
    fig.subplots_adjust(wspace=0)
    ax1 = fig.add_subplot(211)
    ax2 = fig.add_subplot(212)
    ax1.plot(df['t']*1e15,df['Et']*1e4,'C0--')
    E_dn = phi_shift(Et=df['Et']*1e4,t=df['t'],t0=0,dphi=np.pi,shift_wid=2e-15)
    E_dn = d_abs(Et=E_dn,t=df['t'],t0=0,d_abs=.2,abs_wid=2e-15)
    ax1.plot(df['t']*1e15,E_dn,'C4')
    ax1.plot(df['t']*1e15,df['Et']*1e4,'C0')


    ax2.plot(df['t']*1e15,step_func(df['t'],1,0,2e-15)*180,'C4--')
    ax2.axvline(-4, linestyle='--', color='gray')
    ax1.axvline(-4, linestyle='--', color='gray')

    ax3 = ax2.twinx()
    ax3.plot(df['t']*1e15,step_func(df['t'],1,0,2e-15)*-20+100,'C0--')

    ax1.set_xlim(-15,15)
    ax1.set_xlabel('time /fs')
    ax1.set_ylabel('E(t) /a.u.')

    ax2.set_xlim(-15,15)
```

```
    ax3.set_xlim(-15,15)

    #ax2.set_ylim(-5,185)
    #ax3.set_ylim(75,105)

    ax1.set_xticks([])

    ax2.set_xlabel('time /fs')
    ax2.set_ylabel(r'$\Delta \phi$ /deg',color = 'C4')
    ax3.set_ylabel('transmission /\%',color = 'C0')

    ax1.text(.03,.82,'a)',transform=ax1.transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=0.99))
    ax2.text(.03,.82,'b)',transform=ax2.transAxes,bbox=dict(boxstyle='square',
      facecolor='white', alpha=1))
    plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_7.pdf',bbox_inches='tight')
```

```
[ ]: f = np.linspace(-1e16,1e16,2**21)
     pix = np.arange(1280)
     w = v_to_l(f)
     pp_fraction = 1#1/
       gaussian_fraction(x_inf=[-1000,1000],x1=-30,x2=30,amp=1,cen=0,wid=180)
     pol_abs = 1#1/fit_pol_trans(400)
     nd_filter = 1/.005
     scaling = pp_fraction*pol_abs*nd_filter

     df = sim_tt(dn=1e-4,
                 d_absorption=.001,
                 dn_wid=100e-15,
                 t0=1200e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')

     fig, ax = plt.subplots(2)
     #ax[0].plot(df['f'],abs(df['Ef_disp'])**2)
```



```
     #ax[0].plot(df['f'],abs(df['Ef'])**2,'C4--')
     ax[0].plot(df['f'],abs(df['Ef_int'])**2,'C4--')
     ax[0].set_xlim(650e12,850e12)
     #ax[0].set_ylim(-1000,.25e6)
     ax[1].plot(df['t']*1e15,df['Et_disp_o'])
     ax[1].plot(df['t']*1e15,df['Et_disp_e'])
     ax[1].set_xlim(-5000,5000)
```

```
[ ]: delta_phi = np.rad2deg(d_phi(dn=1e-4,l=400e-9,d=50e-6))
     fig, ax = plt.subplots(2)
     fig.subplots_adjust(hspace=.4)
     fig.subplots_adjust(wspace=0)
     ax[1].
       plot(convert_l_plot(df['f'],abs(df['Ef_disp_o_dn'])**2)[0]*1e9,convert_l_plot(df['f'],abs(d
     ax[1].
       plot(convert_l_plot(df['f'],abs(df['Ef_disp_e_dn'])**2)[0]*1e9,convert_l_plot(df['f'],abs(d
     b = convert_l_plot(df['f'],abs(df['Ef_disp_o_dn'])**2)[1]
     a = convert_l_plot(df['f'],abs(df['Ef_disp_o'])**2)[1]
     ax[1].
       plot(convert_l_plot(df['f'],abs(df['Ef_disp_e_dn_o'])**2)[0]*1e9,(a-b)*50,'C2',
       label = r'diff $\times$ 100')
     ax[1].set_xlim(300,500)
     ax[1].set_ylim(-.4,1.3)
     ax[1].set_ylabel('|E($\lambda$)|$^2$ /a.u.')
     ax[1].set_xlabel('wavelength /nm')
     #ax[0].xaxis.set_label_position('top')
     #ax[0].xaxis.tick_top()
     ax[1].legend(loc = 'upper right')
     ax[0].plot(df['t']*1e15,df['Et_disp_o_dn']*1e4,'C4')
     ax[0].plot(df['t']*1e15,df['Et_disp_e_dn']*1e4,'C0')
     ax[0].axvline(1000, color='grey', linestyle='--')
     #ax[0].set_xticks([-4000,0,4000])
     ax[0].set_xlim(-5000,5000)
     ax[0].set_ylim(-7.9,7.9)
     ax[0].set_xlabel('time /fs')
     ax[0].set_ylabel('E(t) /a.u.')
     ax[0].text(.03,.85,'a)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     ax[1].text(.03,.85,'b)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     ax1 = ax[0].twinx()
     ax1.set_ylabel('$\Delta \phi$ /deg ', color = 'C2')
     ax1.plot(df['t']*1e15,step_func(df['t'],1,1100e-15,100e-15)*delta_phi,'C2')
     ax1.tick_params(axis='y')
     ax1.set_ylim(-7.9,7.9)
     #ax1.get_yaxis().set_major_formatter(plt.LogFormatter(12,  labelOnlyBase=False))
     plt.subplots_adjust(left=0.1, right=0.86, top=0.9, bottom=0.1)
```



```
     #plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
       tt_sim_6.pdf',bbox_inches='tight')
```

```
[ ]: tmin = -5000
     tmax = 5000
     t_full = df['t']*1e15
     t_ax = np.where((t_full>tmin)&(t_full<tmax))[0]
     t = t_full[t_ax]
     p1,p2 = cen_time(df['Et_disp_o_dn_e'],df['Et_disp_e_dn_o'])
     p1 = p1[t_ax]
     p2 = p2[t_ax]

     p1pos = p1[np.where(p1>0)[0]]
     tp1pos = t[np.where(p1>0)[0]]
     p1neg = p1[np.where(p1<0)[0]]
     tp1neg = t[np.where(p1<0)[0]]
     p2pos = p2[np.where(p2>0)[0]]
     tp2pos = t[np.where(p2>0)[0]]
     p2neg = p2[np.where(p2<0)[0]]
     tp2neg = t[np.where(p2<0)[0]]
     pposind = np.where((p1 > 0)&(p2 > 0))[0]
     pnegind = np.where((p1 < 0) & (p2 < 0))[0]

     fig = plt.figure(figsize=(3.2,2.4))
     ax = fig.add_subplot(111, projection='3d')
     ax.xaxis.labelpad=8
     ax.yaxis.labelpad=-12
     ax.zaxis.labelpad=-12


     ax.plot(tp2neg,p2neg*0,p2neg,'C4')

     ax.plot(tp1pos,p1pos,0,'C0')
     ax.plot(tp2pos,p2pos*0,p2pos,'C4')
     ax.plot(t[pposind],p1[pposind],p2[pposind],'C2')
     ax.plot(t[pnegind],p1[pnegind],p2[pnegind],'C2')
     ax.plot(tp1neg,p1neg,0,'C0')

     ax2 = plt.axes([0,0,1,1])
     ip = InsetPosition(ax, [0.6,0.5,0.3,0.3*1.333])
     ax2.set_axes_locator(ip)
     ax2.plot(p1,p2*-1,'C2')
     ax2.plot(np.zeros(len(p2)),p2,'C4')
     ax2.plot(p1,np.zeros(len(p2)),'C0')
     ax2.plot(np.linspace(-.0004,.0004,10),np.linspace(-.0004,.0004,10),'--')
     plt.setp(ax2.get_xticklabels(), visible=False)
     plt.setp(ax2.get_yticklabels(), visible=False)
```



```
     ax2.set_xlabel(r'E$_x$(t)')
     ax2.set_ylabel(r'E$_y$(t)')


     plt.setp(ax.get_yticklabels(), visible=False)
     plt.setp(ax.get_zticklabels(), visible=False)
     ax.set_xticks([-4000,0,4000])

     ax.set_xlim(-5000,5000)
     ax.set_ylim(-.0008,.0004)
     ax.set_zlim(-.0005,.0005)
     ax.view_init(elev=45, azim=135)
     ax.set_xlabel('time /fs')
     ax.set_ylabel(r'E$_x$(t) /a.u.')
     ax.set_zlabel(r'E$_y$(t) /a.u.')

     ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
     ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
     ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
     # make the grid lines transparent
     ax.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
     ax.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
     ax.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
     ax.invert_xaxis()

     ax.text(0,-.002,.00575,'c)',transform=ax.transAxes,bbox=dict(boxstyle='square',
       facecolor='white', alpha=0.99))
     plt.subplots_adjust(left=0.05, right=.98, top=1, bottom=.15)
```

```
[ ]: tmin = -5000
     tmax = 5000
     t_full = df['t']*1e15
     t_ax = np.where((t_full>tmin)&(t_full<tmax))[0]
     t = t_full[t_ax]
     p1,p2 = cen_time(df['Et_disp_o_dn_e'],df['Et_disp_e_dn_o'])
     p1 = p1[t_ax]
     p2 = p2[t_ax]

     p1pos = p1[np.where(p1>0)[0]]
     tp1pos = t[np.where(p1>0)[0]]
     p1neg = p1[np.where(p1<0)[0]]
     tp1neg = t[np.where(p1<0)[0]]
     p2pos = p2[np.where(p2>0)[0]]
     tp2pos = t[np.where(p2>0)[0]]
     p2neg = p2[np.where(p2<0)[0]]
     tp2neg = t[np.where(p2<0)[0]]
```

```
fig = plt.figure(figsize=(6.4,3.2))
ax1 = fig.add_subplot(111, projection='3d')
ax1.xaxis.labelpad=8
ax1.yaxis.labelpad=-12
ax1.zaxis.labelpad=-12


ax1.plot(tp1pos,p1pos,0,'C0',alpha=.5,rasterized=True)
#ax1.plot(tp1neg,p1neg,0,'C0',alpha=.5)
ax1.plot(tp2pos,p2pos*0,p2pos,'C4',alpha=.5,rasterized=True)
ax1.plot(t,p1,p2,'C2',rasterized=True)

plt.setp(ax1.get_yticklabels(), visible=False)
plt.setp(ax1.get_zticklabels(), visible=False)
ax1.set_xticks([-4000,0,4000])

ax1.set_xlim(-5000,5000)
ax1.set_ylim(-.0004,.0004)
ax1.set_zlim(-.0004,.0004)
ax1.view_init(elev=45, azim=90)
ax1.set_xlabel('time /fs')
ax1.set_ylabel(r'E$_x$(t) /a.u.')
ax1.set_zlabel(r'E$_y$(t) /a.u.')

ax1.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax1.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax1.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
# make the grid lines transparent
ax1.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
ax1.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
ax1.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
ax1.invert_xaxis()

plt.subplots_adjust(left=-0.2, right=1.15, top=1, bottom=.1)
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_7_1.pdf')

f_ax = np.where((v_to_l(df['f'])*1e9<500)&(v_to_l(df['f'])*1e9>300))
fig = plt.figure(figsize=(6.4,2.4))
ax = fig.add_subplot(111)
ax.plot(v_to_l(df['f'][f_ax])*1e9, abs(df['Ef_int'][f_ax])**2*320,'C2', label =
  'x100')
ax.plot(v_to_l(df['f'][f_ax])*1e9, abs(df['Ef_disp'][f_ax])**2,'C2',alpha=.5)
ax.set_xlim(340,460)
ax.legend()
ax.set_xlabel('wavelength /nm')
```



```
ax.set_ylabel('|E($\lambda$)|$^2$ /a.u.')
plt.text(.065,.7,'b)',transform=ax.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_7_2.pdf',bbox_inches='tight')

fig = plt.figure(figsize=(1.5,1.5))
ax = fig.add_subplot(111)
ax.plot(p1,p2*-1,'C2')
ax.plot(np.zeros(len(p2)),p2,'C4')
ax.plot(p1,np.zeros(len(p2)),'C0')
ax.plot(np.linspace(-.0004,.0004,10),np.linspace(-.0004,.0004,10),'--')
plt.setp(ax.get_xticklabels(), visible=False)
plt.setp(ax.get_yticklabels(), visible=False)
ax.set_xlabel(r'E$_x$(t)')
ax.set_ylabel(r'E$_y$(t)')
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_7_3.pdf',bbox_inches='tight')
```

```
[ ]: df1 = sim_tt(dn=5e-5,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=-1500e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')
     df2 = sim_tt(dn=5e-5,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=-1250e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
```



```
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')
df3 = sim_tt(dn=5e-5,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=-1000e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                mode = 'polarisation')
df4 = sim_tt(dn=5e-5,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=-750e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')
df5 = sim_tt(dn=5e-5,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=-500e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
```



```
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')

     df6 = sim_tt(dn=5e-5,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=1500e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')
     df7 = sim_tt(dn=5e-5,
                 d_absorption=.0,
                 dn_wid=150e-15,
                 t0=1250e-15,
                 l0=400e-9,
                 dl0=50e-9,
                 d_bbo = .005,
                 d_phi_bbo=.0,
                 d_samp=50e-6,
                 angle_pol = -45,
                 amp=1,
                 d_disp=.01,
                 num=2**22,
                 f_range = 1e16,
                 shape = 'gaussian',
                 f=f,
                 mode = 'polarisation')
     df8 = sim_tt(dn=5e-5,
                 d_absorption=.0,
```

```
                dn_wid=150e-15,
                t0=1000e-15,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .005,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.01,
                num=2**22,
                f_range = 1e16,
                shape = 'gaussian',
                f=f,
                mode = 'polarisation')
    df9 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=150e-15,
                t0=750e-15,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .005,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.01,
                num=2**22,
                f_range = 1e16,
                shape = 'gaussian',
                f=f,
                mode = 'polarisation')
    df10 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=150e-15,
                t0=500e-15,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .005,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.01,
                num=2**22,
                f_range = 1e16,
                shape = 'gaussian',
```



```
                f=f,
                mode = 'polarisation')


    fig,ax = plt.subplots(5,2,sharex = True, sharey = True)
    fig.subplots_adjust(hspace=0)
    fig.subplots_adjust(wspace=0)
    ax[0,0].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[0,0].
      plot(convert_l_plot(df1['f'],abs(df1['Ef_int'])**2)[0]*1e9,convert_l_plot(df1['f'],abs(df1[
    ax[1,0].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[1,0].
      plot(convert_l_plot(df2['f'],abs(df2['Ef_int'])**2)[0]*1e9,convert_l_plot(df2['f'],abs(df2[
    ax[2,0].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[2,0].
      plot(convert_l_plot(df3['f'],abs(df3['Ef_int'])**2)[0]*1e9,convert_l_plot(df3['f'],abs(df3[
    ax[3,0].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[3,0].
      plot(convert_l_plot(df4['f'],abs(df4['Ef_int'])**2)[0]*1e9,convert_l_plot(df4['f'],abs(df4[
    ax[4,0].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[4,0].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
    ax[0,1].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[0,1].
      plot(convert_l_plot(df6['f'],abs(df1['Ef_int'])**2)[0]*1e9,convert_l_plot(df6['f'],abs(df6[
    ax[1,1].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[1,1].
      plot(convert_l_plot(df7['f'],abs(df2['Ef_int'])**2)[0]*1e9,convert_l_plot(df7['f'],abs(df7[
    ax[2,1].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
```



```
    ax[2,1].
      plot(convert_l_plot(df8['f'],abs(df3['Ef_int'])**2)[0]*1e9,convert_l_plot(df8['f'],abs(df8[
    ax[3,1].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[3,1].
      plot(convert_l_plot(df9['f'],abs(df4['Ef_int'])**2)[0]*1e9,convert_l_plot(df9['f'],abs(df9[
    ax[4,1].
      plot(convert_l_plot(df5['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df5['f'],abs(df5[
      4)
    ax[4,1].
      plot(convert_l_plot(df10['f'],abs(df5['Ef_int'])**2)[0]*1e9,convert_l_plot(df10['f'],abs(df
    ax[0,0].set_xlim(300,500)
    #ax[4,0].set_xlabel('wavelength /nm')
    for i in range(5):
        for j in range(1):
            ax[i,j].set_ylim(-.05,1.4)
    fig.text(0.02, 0.42,'|E($\lambda$)|$^2$ /a.u.', ha='center',
      rotation='vertical')
    plt.subplots_adjust(left=0.15, right=0.9, top=0.9, bottom=0.1)
    ax[0,0].text(.55,.7,'t=-1500 fs',transform=ax[0,0].transAxes)
    ax[1,0].text(.55,.7,'t=-1250 fs',transform=ax[1,0].transAxes)
    ax[2,0].text(.55,.7,'t=-1000 fs',transform=ax[2,0].transAxes)
    ax[3,0].text(.55,.7,'t=-750 fs',transform=ax[3,0].transAxes)
    ax[4,0].text(.55,.7,'t=-500 fs',transform=ax[4,0].transAxes)
    ax[0,1].text(.55,.7,'t=1500 fs',transform=ax[0,1].transAxes)
    ax[1,1].text(.55,.7,'t=1250 fs',transform=ax[1,1].transAxes)
    ax[2,1].text(.55,.7,'t=1000 fs',transform=ax[2,1].transAxes)
    ax[3,1].text(.55,.7,'t=750 fs',transform=ax[3,1].transAxes)
    ax[4,1].text(.55,.7,'t=500 fs',transform=ax[4,1].transAxes)
    fig.text(0.525, 0.01,'wavelength /nm', ha='center')
    plt.setp(ax[4,0].get_xticklabels()[0], visible=False)
    plt.setp(ax[4,0].get_xticklabels()[3], visible=False)
    plt.setp(ax[4,1].get_xticklabels()[0], visible=False)
    plt.setp(ax[4,1].get_xticklabels()[3], visible=False)
    ax[0,1].yaxis.tick_right()
    ax[1,1].yaxis.tick_right()
    ax[2,1].yaxis.tick_right()
    ax[3,1].yaxis.tick_right()
    ax[4,1].yaxis.tick_right()
    plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_10.pdf',bbox_inches='tight')

[ ]: caf2 = np.genfromtxt('caf2spec.csv', delimiter = ',')
    lcaf = caf2[:,0]*1e-9*1.1-12e-9
    icaf = caf2[:,1]
```



```
    icaf[0:2] = 0
    spec_interp = interpolate.interp1d(lcaf,icaf,fill_value='extrapolate')

[ ]: f = np.linspace(-1e16,1e16,2**21)
    l = v_to_l(f)
    spec = spec_interp(l)
    df = sim_tt(dn=1e-4,
                d_absorption=.0,
                dn_wid=100e-15,
                t0=0e-15,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .003,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.002,
                num=2**21,
                f_range = 1e16,
                shape = 'custom',
                func = spec,
                f=f,
                mode = 'polarisation')

[ ]: fig, ax = plt.subplots(3,2)
    fig.subplots_adjust(hspace=0)
    fig.subplots_adjust(wspace=0)

    ax[0,1].
      plot(convert_l_plot(df['f'],abs(df['Ef_disp'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['Ef
      (abs(df['Ef'])**2).max(),'C2')
    ax[0,0].plot(df['t']*1e15,cen_time(df['Et_disp'])*1e4)
    ax[1,1].
      plot(convert_l_plot(df['f'],abs(df['Ef_disp_o_dn'])**2)[0]*1e9,convert_l_plot(df['f'],abs(d
      (abs(df['Ef_disp_o_dn'])**2).max(),'C2')
    ax[1,0].plot(df['t']*1e15,df['Et_disp_o_dn']*1e4,'C4')
    ax[1,0].plot(df['t']*1e15,df['Et_disp_e_dn']*1e4,'C0')
    #ax[1,0].plot(df['t']*1e15,step_func(df['t'],1,1400e-15,100e-15)*7,'C2')
    ax[1,0].axvline(200,color='C2')
    ax[2,1].
      plot(convert_l_plot(df['f'],abs(df['Ef_int'])**2)[0]*1e9,(convert_l_plot(df['f'],abs(df['Ef
      (abs(df['Ef_int'])**2).max())*.94,'C2')
    ax[2,1].
      plot(convert_l_plot(df['f'],abs(df['Ef_int'])**2)[0]*1e9,convert_l_plot(df['f'],abs(df['Ef_
      (abs(df['Ef_disp'])**2).max(),'C2',alpha=.4)
    ax[2,0].plot(df['t']*1e15,np.roll(df['Et_disp_o_dn_e'],-160000)*1e4,'C4')
```

```
    ax[2,0].plot(df['t']*1e15,np.roll(df['Et_disp_e_dn_o'],-160000)*1e4,'C0')


    for i in range(3):
        ax[i,1].yaxis.tick_right()
        #ax[i,0].set_ylim(0,.99)
        ax[i,1].set_xlim(200,900)
        ax[i,0].set_xlim(-5000,5000)
        for j in range(2):
            plt.setp(ax[i,j].get_yticklabels(), visible=False)
    for i in range(2):
        for j in range(2):
            plt.setp(ax[i,j].get_xticklabels(), visible=False)
    plt.setp(ax[2,1].get_xticklabels()[0], visible=False)
    plt.setp(ax[2,1].get_xticklabels()[3], visible=False)
    ax[2,1].set_xlabel('wavelength /nm')
    ax[2,0].set_xlabel('time /fs')
    ax[0,0].text(.04,.8,'a)',transform=ax[0,0].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    ax[0,1].text(.04,.8,'a*)',transform=ax[0,1].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    ax[1,0].text(.04,.8,'b)',transform=ax[1,0].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    ax[1,1].text(.04,.8,'b*)',transform=ax[1,1].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    ax[2,0].text(.04,.8,'c)',transform=ax[2,0].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    ax[2,1].text(.04,.8,'c*)',transform=ax[2,1].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    #ax[3,0].text(.04,.65,'d)',transform=ax[3,0].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    #ax[3,1].text(.04,.65,'d*)',transform=ax[3,1].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    #ax[4,0].text(.04,.65,'e)',transform=ax[4,0].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    #ax[4,1].text(.04,.65,'e*)',transform=ax[4,1].
      transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
    fig.text(0.95, 0.4,'|E($\lambda$)|$^2$ /a.u.', ha='center', rotation='vertical')
    fig.text(0.05, 0.4,'E(t) /a.u.', ha='center', rotation='vertical')
    plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
      tt_sim_caf2_spec_1.pdf',bbox_inches='tight')

[ ]: f = np.linspace(-1e16,1e16,2**21)
    l = v_to_l(f)
    spec = spec_interp(l)
    t0 = 500e-15
```



```
    dt0 = 150e-15
    df1 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=100e-15,
                t0=t0,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .003,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.002,
                num=2**21,
                f_range = 1e16,
                shape = 'gaussian',
                func = spec,
                f=f,
                mode = 'polarisation')

    df11 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=100e-15,
                t0=t0-dt0,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .003,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.002,
                num=2**21,
                f_range = 1e16,
                shape = 'gaussian',
                func = spec,
                f=f,
                mode = 'polarisation')

    df2 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=100e-15,
                t0=t0,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .003,
                d_phi_bbo=.0,
```



```
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.025,
                num=2**21,
                f_range = 1e16,
                shape = 'gaussian',
                func = spec,
                f=f,
                mode = 'polarisation')

    df22 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=100e-15,
                t0=t0-dt0,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .003,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.025,
                num=2**21,
                f_range = 1e16,
                shape = 'gaussian',
                func = spec,
                f=f,
                mode = 'polarisation')

    df3 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=150e-15,
                t0=t0,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .003,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.07,
                num=2**21,
                f_range = 1e16,
                shape = 'gaussian',
                func = spec,
                f=f,
```



```
                mode = 'polarisation')

    df33 = sim_tt(dn=5e-5,
                d_absorption=.0,
                dn_wid=150e-15,
                t0=t0-dt0,
                l0=400e-9,
                dl0=50e-9,
                d_bbo = .003,
                d_phi_bbo=.0,
                d_samp=50e-6,
                angle_pol = -45,
                amp=1,
                d_disp=.07,
                num=2**21,
                f_range = 1e16,
                shape = 'gaussian',
                func = spec,
                f=f,
                mode = 'polarisation')

[ ]: fig, ax = plt.subplots(3,2)
    fig.subplots_adjust(hspace=0)
    fig.subplots_adjust(wspace=0)

    data = [df1, df11, df2, df22, df3, df33]
    steps = [500e-15, 500e-15, 500e-15]

    #for i in range(len(data)):
    #    ax[i,0].plot(data[i]['t']*1e15,data[i]['Et_disp_o_dn']*1e4,'C4')
    #    ax[i,0].plot(data[i]['t']*1e15,data[i]['Et_disp_e_dn']*1e4,'C0')
    #    ax[i,1].
      plot(convert_l_plot(data[i]['f'],abs(data[i]['Ef_int'])**2)[0]*1e9,convert_l_plot(data[i]['
    #    ax[i,1].
      plot(convert_l_plot(data[i]['f'],abs(data[i]['Ef_int'])**2)[0]*1e9,convert_l_plot(data[i]['
      0303,'C2',alpha=.4)

    ax[0,0].plot(df1['t']*1e15,df1['Et_disp_o_dn']*1e4,'C4')
    ax[0,0].plot(df1['t']*1e15,df1['Et_disp_e_dn']*1e4,'C0')
    ax[0,1].
      plot(convert_l_plot(df1['f'],abs(df1['Ef_int'])**2)[0]*1e9,convert_l_plot(df1['f'],abs(df1[
    ax[0,1].
      plot(convert_l_plot(df1['f'],abs(df1['Ef_int'])**2)[0]*1e9,convert_l_plot(df1['f'],abs(df11
    ax[0,1].
      plot(convert_l_plot(df1['f'],abs(df1['Ef_int'])**2)[0]*1e9,convert_l_plot(df1['f'],abs(df1[
      4)
    ax[0,1].plot(406.8,.95/2,'C2o')
```

```
ax[0,1].plot(416.6,.5,'C2o')

ax[1,0].plot(df2['t']*1e15,df2['Et_disp_o_dn']*1e4,'C4')
ax[1,0].plot(df2['t']*1e15,df2['Et_disp_e_dn']*1e4,'C0')
ax[1,1].
  plot(convert_l_plot(df2['f'],abs(df2['Ef_int'])**2)[0]*1e9,convert_l_plot(df2['f'],abs(df2[
ax[1,1].
  plot(convert_l_plot(df2['f'],abs(df2['Ef_int'])**2)[0]*1e9,convert_l_plot(df2['f'],abs(df22
ax[1,1].
  plot(convert_l_plot(df2['f'],abs(df2['Ef_int'])**2)[0]*1e9,convert_l_plot(df2['f'],abs(df2[
  4)
ax[1,1].plot(402,.5,'C2o')
ax[1,1].plot(406,.5,'C2o')

ax[2,0].plot(df3['t']*1e15,df3['Et_disp_o_dn']*1e4,'C4')
ax[2,0].plot(df3['t']*1e15,df3['Et_disp_e_dn']*1e4,'C0')
ax[2,1].
  plot(convert_l_plot(df3['f'],abs(df3['Ef_int'])**2)[0]*1e9,convert_l_plot(df3['f'],abs(df3[
ax[2,1].
  plot(convert_l_plot(df3['f'],abs(df3['Ef_int'])**2)[0]*1e9,convert_l_plot(df3['f'],abs(df33
ax[2,1].
  plot(convert_l_plot(df3['f'],abs(df3['Ef_int'])**2)[0]*1e9,convert_l_plot(df3['f'],abs(df3[
  4)
ax[2,1].plot(400.5,.5,'C2o')
ax[2,1].plot(402.5,.5,'C2o')

ax[0,0].
  plot(data[0]['t']*1e15,step_func(data[0]['t'],1,steps[0],100e-15)*7,'C2')
ax[1,0].
  plot(data[1]['t']*1e15,step_func(data[1]['t'],1,steps[1],100e-15)*7,'C2')
ax[2,0].plot(data[2]['t']*1e15,step_func(data[2]['t'],1,steps[2],100e-15)*3.
  5,'C2')
ax[0,0].
  plot(data[0]['t']*1e15,step_func(data[0]['t'],1,steps[0]-150e-15,100e-15)*7,'C2--')
ax[1,0].
  plot(data[1]['t']*1e15,step_func(data[1]['t'],1,steps[1]-150e-15,100e-15)*7,'C2--')
ax[2,0].
  plot(data[2]['t']*1e15,step_func(data[2]['t'],1,steps[2]-150e-15,100e-15)*3.
  5,'C2--')
#ax[3,0].plot(data[3]['t']*1e15,step_func(data[3]['t'],1,steps[3],100e-15)*3.
  5,'C2')


for i in range(3):
    ax[i,1].yaxis.tick_right()
    ax[i,0].set_ylim(-9.9,9.9)
```



```
    ax[i,0].set_xlim(-5000,5000)
    ax[i,1].set_xlim(340,460)
    for j in range(2):
        plt.setp(ax[i,j].get_yticklabels(), visible=False)
for i in range(2):
    for j in range(2):
        plt.setp(ax[i,j].get_xticklabels(), visible=False)
plt.setp(ax[2,0].get_xticklabels()[0], visible=False)
plt.setp(ax[2,0].get_xticklabels()[3], visible=False)
plt.setp(ax[2,1].get_xticklabels()[0], visible=False)
#plt.setp(ax[3,1].get_xticklabels()[0], visible=False)
ax[2,0].set_xlabel('time /fs')
ax[2,1].set_xlabel('wavelength /nm')
fig.text(0.95, 0.4,'|E($\lambda$)|$^2$ /a.u.', ha='center', rotation='vertical')
fig.text(0.05, 0.4,'E(t) /a.u.', ha='center', rotation='vertical')
ax[0,0].text(.04,.74,'a)',transform=ax[0,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[0,1].text(.04,.74,'a*)',transform=ax[0,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[1,0].text(.04,.74,'b)',transform=ax[1,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[1,1].text(.04,.74,'b*)',transform=ax[1,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[2,0].text(.04,.74,'c)',transform=ax[2,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
ax[2,1].text(.04,.74,'c*)',transform=ax[2,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
#ax[3,0].text(.04,.74,'d)',transform=ax[3,0].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
#ax[3,1].text(.04,.74,'d*)',transform=ax[3,1].
  transAxes,bbox=dict(boxstyle='square', facecolor='white', alpha=0.99))
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_caf2_bbo_chirp.pdf',bbox_inches='tight')
```



```
[ ]: dn=0
     dn_wid=100e-15
     d_absorption = 0
     t0=600e-15
     l0=400e-9
     dl0=50e-9
     d_samp=50e-6
     amp=1
     d_disp=.002
     d_phi_bbo=.0
     angle_pol = -45
     d_bbo = .003
```

```
dn_bbo=0
num=2**21
f_range = 1e16
num = (2**20)



f = np.linspace(-f_range,f_range,num)
w = v_to_w(f)
df = 1/(f[1] - f[0])
t = np.linspace(-df/2, df/2, num)
f0 = l_to_v(l0)
w0 = v_to_w(f0)
df0 = l_to_v(l0-dl0/2) - l_to_v(l0+dl0/2)
E0 = amp
Ef = amp*np.exp(-(f-f0)**2/(2*(df0/2.3548)**2))
l = np.linspace(.25,1.2,10000)

l_ax = np.where((v_to_l(f)*1e9<500)&(v_to_l(f)*1e9>300))

n_sio2 = sellmaier_sio2(l)
n_bbo_o = sellmaier_bbo_o(l)
n_bbo_e = sellmaier_bbo_e(l)
n0_bbo_o = sellmaier_bbo_o(l0*1e6)
n0_bbo_e = sellmaier_bbo_e(l0*1e6)
n0_sio2 = sellmaier_sio2(l0*1e6)
dn1,dn2,dn3,gvd_sio2,tod_sio2,gv_sio2 = calc_disp(n_sio2,l,l0*1e6,n0=n0_sio2)
dn1,dn2,dn3,gvd_bbo_o,tod_bbo_o,gv_bbo_o =
  calc_disp(n_bbo_o,l,l0*1e6,n0=n0_bbo_o)
dn1,dn2,dn3,gvd_bbo_e,tod_bbo_e,gv_bbo_e =
  calc_disp(n_bbo_e,l,l0*1e6,n0=n0_bbo_e)
Ef_disp = apply_dispersion(Ef=Ef, f=f,
  n0=n0_sio2,GVD=gvd_sio2,TOD=tod_sio2,z=d_disp)

Et = ifft(Ef)
Et_disp = ifft(Ef_disp)
Et_disp = cen_time(Et_disp)
Ef_disp_o = apply_dispersion(Ef=Ef_disp, f=f,
  n0=n0_bbo_o,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=d_bbo)
Ef_disp_e = apply_dispersion(Ef=Ef_disp, f=f,
  n0=n0_bbo_e,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=d_bbo)
Et_disp_o = ifft(Ef_disp_o)
Et_disp_e = ifft(Ef_disp_e)
Et_disp_o,Et_disp_e = cen_time(Et1=Et_disp_o, Et2=Et_disp_e)
#t0 = 5*750e-15
#dphi = np.append(np.arange(0,20,1),np.arange(20,100,5))
#dphi = np.append(dphi,np.arange(100,1240,20))
```



```
dphi = np.arange(0,11)
dphi = np.append(dphi,np.arange(20,1280,10))
dphi = np.deg2rad(dphi)
specs = []
for i in range(len(dphi)):
#print(dphi)
    print(int(i/len(dphi)*100), end='  ')
    Et_disp_o_dn =
  d_abs(Et=Et_disp_o,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
    Et_disp_e_dn =
  d_abs(Et=Et_disp_e,t=t,t0=t0,d_abs=d_absorption,abs_wid=dn_wid)
    Et_disp_o_dn =
  phi_shift(Et=Et_disp_o_dn,t=t,t0=t0,dphi=dphi[i],shift_wid=dn_wid)
    Et_disp_e_dn =
  phi_shift(Et=Et_disp_e_dn,t=t,t0=t0,dphi=dphi[i],shift_wid=dn_wid)
    Ef_disp_o_dn = fft(Et_disp_o_dn)
    Ef_disp_e_dn = fft(Et_disp_e_dn)
    Ef_disp_o_dn_e = apply_dispersion(Ef=Ef_disp_o_dn, f=f, n0=n0_bbo_e

  ,GVD=gvd_bbo_e,TOD=tod_bbo_e,z=(d_bbo+dn_bbo))
    Ef_disp_e_dn_o = apply_dispersion(Ef=Ef_disp_e_dn, f=f, n0=n0_bbo_o

  ,GVD=gvd_bbo_o,TOD=tod_bbo_o,z=(d_bbo+dn_bbo))
    Et_disp_o_dn_e = (ifft(Ef_disp_o_dn_e))
    Et_disp_e_dn_o = (ifft(Ef_disp_e_dn_o))
    ####################################

    a, Et_int = jones(Et_disp_o_dn_e, Et_disp_e_dn_o, -45)
    Ef_int = fft(Et_int)
    specs.append(abs(Ef_int[l_ax])**2)
specs=np.array(specs)
```

```
[ ]: x = v_to_l(f[l_ax])*1e9
     y = np.rad2deg(dphi)
     X, Y = np.meshgrid(x, y)


     from matplotlib.patches import Rectangle
     fig = plt.figure(figsize=(9,4.8))
     fig.subplots_adjust(hspace=0)
     fig.subplots_adjust(wspace=0)


     ax1 = plt.subplot2grid((3,6), (0,0), colspan=3, rowspan=3)
     ax2 = plt.subplot2grid((3,6), (0,3), colspan=3, rowspan=1)
     ax3 = plt.subplot2grid((3,6), (1,3), colspan=3, rowspan=1)
```

```
ax4 = plt.subplot2grid((3,6), (2,3), colspan=3, rowspan=1)


im = ax1.pcolormesh(x,y,specs,norm=matplotlib.colors.LogNorm(),vmin=1e-4,vmax=1.
  4,cmap='magma', rasterized = True)
#plt.gca().invert_yaxis()
ax1.set_ylim(ax1.get_ylim()[::-1])
ax_cbar = fig.add_axes([.126, .83, .386, 0.05])
plt.colorbar(im, cax=ax_cbar, orientation='horizontal', ticklocation = 'top')
fig.text(0.3, 0.98, '|E($\lambda$)|$^2$ /a.u.', ha='center', va='center',
  rotation='horizontal')


ax4.plot(v_to_l(f[l_ax])*1e9,specs[71], label=''+str(int(np.
  rad2deg(dphi[71])))+'°')
ax4.plot(v_to_l(f[l_ax])*1e9,specs[91], label=''+str(int(np.
  rad2deg(dphi[91])))+'°')
ax4.plot(v_to_l(f[l_ax])*1e9,specs[111], label=''+str(int(np.
  rad2deg(dphi[111])))+'°')
ax4.plot(v_to_l(f[l_ax])*1e9,specs[131], label=''+str(int(np.
  rad2deg(dphi[131])))+'°')
ax4.plot(v_to_l(f[l_ax])[np.diff(specs[71]).argmax()-50]*1e9,specs[71][np.
  diff(specs[71]).argmax()-50],'C0o')
ax4.plot(v_to_l(f[l_ax])[np.diff(specs[91]).argmax()-40]*1e9,specs[91][np.
  diff(specs[91]).argmax()-40],'C1o')
ax4.plot(v_to_l(f[l_ax])[np.diff(specs[111]).argmax()-30]*1e9,specs[111][np.
  diff(specs[111]).argmax()-30],'C2o')
ax4.plot(v_to_l(f[l_ax])[np.diff(specs[131]).argmax()-30]*1e9,specs[131][np.
  diff(specs[131]).argmax()-20],'C3o')
ax4.axvline(399, color = 'gray')
ax4.legend(frameon = False, loc = (.65,.0))
ax4.set_xlim(365,450)
ax4.set_ylim(-.05,1.999)
ax4.yaxis.tick_right()

ax3.plot(v_to_l(f[l_ax])*1e9,specs[17], label=''+str(int(np.
  rad2deg(dphi[17])))+'°')
ax3.plot(v_to_l(f[l_ax])*1e9,specs[23], label=''+str(int(np.
  rad2deg(dphi[23])))+'°')
ax3.plot(v_to_l(f[l_ax])*1e9,specs[27], label=''+str(int(np.
  rad2deg(dphi[27])))+'°')
ax3.plot(v_to_l(f[l_ax])*1e9,specs[31], label=''+str(int(np.
  rad2deg(dphi[31])))+'°')
ax3.plot(v_to_l(f[l_ax])[np.diff(specs[17]).argmax()]*1e9,specs[17][np.
  diff(specs[17]).argmax()],'C0o')
```



```
ax3.plot(v_to_l(f[l_ax])[np.diff(specs[23]).argmax()]*1e9,specs[23][np.
  diff(specs[23]).argmax()],'C1o')
ax3.plot(v_to_l(f[l_ax])[np.diff(specs[27]).argmax()]*1e9,specs[27][np.
  diff(specs[27]).argmax()],'C2o')
ax3.plot(v_to_l(f[l_ax])[np.diff(specs[31]).argmax()]*1e9,specs[31][np.
  diff(specs[31]).argmax()],'C3o')
ax3.axvline(399, color = 'gray')
ax3.legend(frameon = False, loc = (.65,.0))
ax3.set_xlim(365,450)
ax3.yaxis.tick_right()

ax2.plot(v_to_l(f[l_ax])*1e9,specs[1], label=''+str(int(np.
  rad2deg(dphi[1])))+'°')
ax2.plot(v_to_l(f[l_ax])*1e9,specs[2], label=''+str(int(np.
  rad2deg(dphi[2])))+'°')
ax2.plot(v_to_l(f[l_ax])*1e9,specs[3], label=''+str(int(np.
  rad2deg(dphi[3])))+'°')
ax2.plot(v_to_l(f[l_ax])*1e9,specs[4], label=''+str(int(np.
  rad2deg(dphi[4])))+'°')
ax2.plot(v_to_l(f[l_ax])[np.diff(specs[1]).argmax()]*1e9,specs[1][np.
  diff(specs[1]).argmax()],'C0o')
ax2.plot(v_to_l(f[l_ax])[np.diff(specs[2]).argmax()]*1e9,specs[2][np.
  diff(specs[2]).argmax()],'C1o')
ax2.plot(v_to_l(f[l_ax])[np.diff(specs[3]).argmax()]*1e9,specs[3][np.
  diff(specs[3]).argmax()],'C2o')
ax2.plot(v_to_l(f[l_ax])[np.diff(specs[4]).argmax()]*1e9,specs[4][np.
  diff(specs[4]).argmax()],'C3o')
#ax2.plot(v_to_l(f[l_ax])[7910]*1e9,specs[1][7906],'C0o')
#ax2.plot(v_to_l(f[l_ax])[7910]*1e9,specs[2][7906],'C1o')
#ax2.plot(v_to_l(f[l_ax])[7910]*1e9,specs[3][7906],'C2o')
#ax2.plot(v_to_l(f[l_ax])[7910]*1e9,specs[4][7906],'C3o')
ax2.axvline(399, color = 'gray')
ax2.legend(frameon = False, loc = (.65,.0))
ax2.set_xlim(365,450)
ax2.yaxis.tick_right()
#plt.setp(ax2.get_yticklabels()[1], visible=False)
fig.text(0.99, 0.5, '|E($\lambda$)|$^2$ /a.u.', ha='center', va='center',
  rotation='vertical')
fig.text(0.04, 0.5, 'phase shift /deg', ha='center', va='center',
  rotation='vertical')
fig.text(0.5, 0.02, 'wavelength /nm', ha='center', va='center',
  rotation='horizontal')
plt.subplots_adjust(left = .125, bottom = .1, right = .9, top = .825)
ax1.text(.04,.92,'a)',transform=ax1.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
```



```
ax2.text(.04,.76,'b)',transform=ax2.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
ax3.text(.04,.76,'c)',transform=ax3.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
ax4.text(.04,.76,'d)',transform=ax4.transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_phase_shift_dependence.pdf',bbox_inches='tight')
```

```
[ ]: specs_diff_max = []
     for i in range(1,len(specs)):
         specs_diff_max.append(np.diff(specs[i]).argmax())
     specs_diff_max = np.array(specs_diff_max)
     specs_diff_max_n = []
     for i in range(len(specs_diff_max)):
         specs_diff_max_n.append(specs_diff_max[i] - specs_diff_max.min())
     specs_diff_max_n = np.array(specs_diff_max_n)
```

```
[ ]: l = (constants.c/f[l_ax])*1e9

     fig, ax = plt.subplots(2,figsize=(6.4,4.8))
     fig.subplots_adjust(hspace=0)
     fig.subplots_adjust(wspace=0)

     for i in range(1,11):
         ax[0].plot(v_to_l(f[l_ax])*1e9,specs[i]/specs[i].max(),'C0')
     ax[0].plot(v_to_l(f[l_ax])*1e9,specs[2]/specs[2].max(),'C0',label = r'$\Delta
       \phi$: 1° - 20°')
     ax[0].plot(v_to_l(f[l_ax])*1e9,specs[27]/specs[27].max(),'C2',label = r'$\Delta
       \phi$: 180°')
     ax[0].plot(v_to_l(f[l_ax])*1e9,specs[33]/specs[33].max(),'C4',label = r'$\Delta
       \phi$: 240°')
     ax[0].plot(l[specs_diff_max][13],.5,'o', color = 'C0')
     ax[0].plot(l[specs_diff_max][27],.5,'o', color = 'C2')
     ax[0].plot(l[specs_diff_max][33],.5,'o', color = 'C4')
     ax[0].set_xlim(340,460)
     ax[0].xaxis.set_label_position('top')
     ax[0].xaxis.tick_top()
     ax[0].legend()
     ax[0].set_xlabel('wavelength /nm')
     ax[0].set_ylabel('|E($\lambda$)|$^2$ /norm.')

     ax[1].plot(np.rad2deg(dphi[1::]),l[specs_diff_max]-l[specs_diff_max[1]],'.')
     ax[1].set_xlim(0,420)
     ax[1].set_ylim(0,9)
     ax[1].set_xlabel('phase shift /deg')
```



```
ax[1].set_ylabel('edge pos. shift /nm')

axins1 = ax[1].inset_axes([0.58, 0.15, 0.4, 0.4])
axins1.plot(np.rad2deg(dphi[1::]),l[specs_diff_max]-l[specs_diff_max[1]],'.')
axins1.set_xlim(0, 60)
axins1.set_ylim(-.1, 2)
ax[1].indicate_inset_zoom(axins1)


ax[0].text(.03,.86,'a)',transform=ax[0].transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
ax[1].text(.03,.86,'b)',transform=ax[1].transAxes,bbox=dict(boxstyle='square',
  facecolor='white', alpha=0.99))
plt.savefig('/Users/diezm/Nextcloud/thesis/numerical_results/tt_sim/figures/
  tt_sim_phase_signal_pos.pdf',bbox_inches='tight')
```

r290_full_mp_fitting

April 19, 2022

[1]:
```
import os
import numpy as np
import matplotlib.pyplot as plt
import os
from scipy.signal import savgol_filter
import lmfit
from lmfit import Model
from scipy.stats import norm
from cycler import cycler
from scipy.stats import gaussian_kde
from matplotlib.patches import Circle
from scipy.special import erf
from copy import deepcopy
from scipy.signal import find_peaks
from matplotlib.widgets import Slider
import numpy.polynomial.polynomial as poly
from IPython.parallel import Client
import time
import tqdm
import pickle

cmap=plt.cm.magma
# build cycler with 5 equally spaced colors from that colormap
c = cycler('color', cmap(np.linspace(.2,.8,5)) )
# supply cycler to the rcParam
plt.rcParams["axes.prop_cycle"] = c

import sys
sys.path.append('/home/diezm/python/timing_tool/exfel/beamtime2')
import exfel_ana as ex


from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))

import warnings
warnings.filterwarnings('ignore')
```



```
/home/diezm/.conda/envs/diezm2/lib/python3.7/site-
packages/IPython/parallel.py:13: ShimWarning: The `IPython.parallel` package has
been deprecated since IPython 4.0. You should import from ipyparallel instead.
  "You should import from ipyparallel instead.", ShimWarning)
```

```
<IPython.core.display.HTML object>
```

[2]:
```
#############Separated data from HDF5 File
specs = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/specs_sav.
  npy')
specs_avg = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  specs_avg.npy')
delays = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/delays.
  npy', allow_pickle = True)
single_ipm = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  ipm_single.npy', allow_pickle = True)
single_ipm_avg = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  ipm_single_avg.npy')
beam_pos_x = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  beam_pos_x.npy')
beam_pos_y = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  beam_pos_y.npy')
beam_pos_x_avg = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  beam_pos_x_avg.npy')
beam_pos_y_avg = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  beam_pos_y_avg.npy')
phaseshifter = np.load('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/
  phaseshifter.npy')
```

[1]:
```
##### Multicore fitting
```

[3]:
```
cluster = Client()
```

[4]:
```
lb_view = cluster.load_balanced_view()
```

[ ]:

[2]:
```
#Import for all cluster CPUs
```

[7]:
```
%px import os
%px import numpy as np
%px import matplotlib.pyplot as plt
%px import os
%px from scipy.signal import savgol_filter
%px import lmfit
%px from lmfit import Model
```



```
%px from scipy.stats import norm
%px from cycler import cycler
%px from scipy.stats import gaussian_kde
%px from matplotlib.patches import Circle
%px from scipy.special import erf
%px from copy import deepcopy
%px from scipy.signal import find_peaks
%px from matplotlib.widgets import Slider
%px import numpy.polynomial.polynomial as poly
%px import exfel_ana as ex
%px import time
%px import tqdm
```

[ ]:

[8]:
```
df_full = {}
delays_all = [0,1,2,3,13,14,15,16]
for i in delays_all:
    print(i, end = '  ')
    df_full['delay'+str(i)+''] = {}
    for j in range(len(delays[i])):
        df_full['delay'+str(i)+'']['train'+str(j)+''] = {'specs':np.
  zeros((len(specs[delays[i][j]]),1280))*np.nan,
                                                         'beam_pos_x':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'beam_pos_y':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'beam_intensity':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_cen':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_amp':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_wid':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_cen_std':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_amp_std':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_wid_std':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_intersect':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_slope':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
```



```
                                                         'step_intersect_std':np.
  zeros(len(specs[delays[i][j]]))*np.nan,
                                                         'step_slope_std':np.
  zeros(len(specs[delays[i][j]]))*np.nan}
```

```
16
```

[9]:
```
#delays_all = [0,1,2,3,13,14,15,16]
delays_all = [15,16]
for i in delays_all:
    print(i,end='  ')
    for j in range(len(delays[i])):
        df_full['delay'+str(i)+'']['train'+str(j)+'']['specs'] =
  specs[delays[i]][j]
        df_full['delay'+str(i)+'']['train'+str(j)+'']['beam_pos_x'] =
  beam_pos_x[delays[i]][j]
        df_full['delay'+str(i)+'']['train'+str(j)+'']['beam_pos_y'] =
  beam_pos_y[delays[i]][j]
        df_full['delay'+str(i)+'']['train'+str(j)+'']['beam_intensity'] =
  single_ipm[delays[i]][j]
    with open('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/df_full_all.
  p', 'wb') as f:
        pickle.dump(df_full, f, pickle.HIGHEST_PROTOCOL)
```

```
16
```

[8]:
```
with open('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/df_full_all.p',
  'rb') as f:
    df_full = pickle.load(f)
```

[10]:
```
delays_r = [0,1,2,3]
#delays_r = [2]
for k in delays_r:
    data = np.concatenate(specs[delays[k]])
    print('Current delay: '+str(k)+'')
    start = time.time()
    b = lb_view.map(ex.fit_step_delay_r,data,window = 100, edge = 'r',
  step_amp_min = -1e5, block=True, chunksize = 50)
    end = time.time()
    print('Time for '+str(len(data))+' fits: '+str(end-start)+' seconds')
    delay = k
    list_step = np.linspace(48,len(b)-1,int(len(b)/49),dtype = int)
    j = 0
    c = 0
    for i in range(len(b)):
        df_full['delay'+str(delay)+'']['train'+str(j)+'']['specs'][c] = data[i]
```

```
            if b[i] is not None:
                df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_cen'][c] =
        b[i][1]['step_cen'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_cen_std'][c] =
        b[i][1]['step_cen'].stderr
                df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_amp'][c] =
        b[i][1]['step_amp'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_amp_std'][c] =
        b[i][1]['step_amp'].stderr
                df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_wid'][c] =
        b[i][1]['step_wid'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_wid_std'][c] =
        b[i][1]['step_wid'].stderr

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_intersect'][c] =
        b[i][1]['step_intersect'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_intersect_std'][c] =
        b[i][1]['step_intersect'].stderr
                df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_slope'][c]
        = b[i][1]['step_slope'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_slope_std'][c] =
        b[i][1]['step_slope'].stderr
                c += 1
                if i in list_step:
                    j += 1
                    c = 0
            else:
                c += 1
                if i in list_step:
                    j += 1
                    c = 0
```

```
Current delay: 0
Time for 14259 fits: 4.192219495773315 seconds
Current delay: 1
Time for 19600 fits: 8.931103229522705 seconds
Current delay: 2
Time for 19404 fits: 18.16642475128174 seconds
Current delay: 3
Time for 14553 fits: 15.49581003189087 seconds
```



```
[11]: delays_f = [13,14,15,16]
      #delays_f = [14]
      for k in delays_f:
          data = np.concatenate(specs[delays[k]])
          print('Current delay: '+str(k)+'')
          start = time.time()
          b = lb_view.map(ex.fit_step_delay_f,data, block=True, chunksize = 50)
          end = time.time()
          print('Time for '+str(len(data))+' fits: '+str(end-start)+' seconds')
          delay = k
          list_step = np.linspace(48,len(b)-1,int(len(b)/49),dtype = int)
          j = 0
          c = 0
          for i in range(len(b)):
              if b[i] is not None:
                  df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_cen'][c] =
        b[i][1]['step_cen'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_cen_std'][c] =
        b[i][1]['step_cen'].stderr
                df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_amp'][c] =
        b[i][1]['step_amp'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_amp_std'][c] =
        b[i][1]['step_amp'].stderr
                 df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_wid'][c] =
        b[i][1]['step_wid'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_wid_std'][c] =
        b[i][1]['step_wid'].stderr

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_intersect'][c] =
        b[i][1]['step_intersect'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_intersect_std'][c] =
        b[i][1]['step_intersect'].stderr
                df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_slope'][c]
        = b[i][1]['step_slope'].value

        df_full['delay'+str(delay)+'']['train'+str(j)+'']['step_slope_std'][c] =
        b[i][1]['step_slope'].stderr
                c += 1
                if i in list_step:
                    j += 1
                    c = 0
            else:
```



```
                c += 1
                if i in list_step:
                    j += 1
                    c = 0
```

```
Current delay: 13
Time for 15827 fits: 15.953448295593262 seconds
Current delay: 14
Time for 23226 fits: 20.64716410636902 seconds
Current delay: 15
Time for 16807 fits: 13.536579132080078 seconds
Current delay: 16
Time for 18375 fits: 8.896865129470825 seconds
```

```
[ ]:
```

```
[12]: with open('/gpfs/exfel/data/scratch/diezm/exfel_reduced/run290/df_full_all.p',
          'wb') as f:
              pickle.dump(df_full, f, pickle.HIGHEST_PROTOCOL)
```

C. List of Publications

2022

Michael Diez, Andreas Galler, Henning Kirchberg, Michael Thorwart, ... and Christian Bressler. *A sensitive high repetition rate arrival time monitor for X-Ray Free Electron Lasers* (to be published).

2021

Michael Diez, Andreas Galler, Sebastian Schulz, Christina Boemer, Ryan N. Coffee, Nick Hartmann, Rupert Heider, Martin S. Wagner, Wolfram Helml, Tetsuo Katayama, Tokushi Sato, Takahiro Sato, Makina Yabashi, and Christian Bressler. *A self-referenced in-situ arrival time monitor for X-ray free-electron lasers*. Scientific Reports 11, 3562 (2021)

2020

Christina Boemer, Dietrich Krebs, **Michael Diez**, Nina Rohringer, Andreas Galler, and Christian Bressler. *X-ray parametric down-conversion: Challenging previous findings on the basis of improved experimental methods*. arXiv preprint arXiv:2002.12822.

Dmitry Khakhulin, Florian Otte, Mykola Biednov, Christina Bömer, Tae-Kyu Choi, **Michael Diez**, Andreas Galler, Yifeng Jiang, Katharina Kubicek, Frederico Alves Lima, Angel Rodriguez-Fernandez, Peter Zalden, Wojciech Gawelda, and Christian Bressler. *Ultrafast x-ray photochemistry at European XFEL: Capabilities of the femtosecond X-ray Experiments (FXE) instrument*. Applied Sciences, 10(3), 2020.

2019

Andreas Galler, Wojciech Gawelda, Mykola Biednov, Christina Bomer, Alexander Britz, Sandor Brockhauser, Tae-Kyu Choi, **Michael Diez**, Paul Frankenberger, Marcus French, Dennis Görries, Matthiew Hart, Steffen Hauf, Dmitry Khakhulin, Martin Knoll, Timo Korsch, Katharina Kubicek, Markus Kuster, Philipp Lang, Frederico Alves Lima, Florian Otte, Sebastian Schulz, Peter Zalden, and Christian Bressler. *Scientific instrument Femtosecond X-ray Experiments (FXE): instrumentation and baseline experimental capabilities*. Journal of Synchrotron Radiation, 26(5):1432–1447, Sep 2019.

Alexander Britz, Wojciech Gawelda, Tadesse A. Assefa, Lindsey L. Jamula, Jonathan T. Yarranton, Andreas Galler, Dmitry Khakhulin, **Michael Diez**, Manuel Harder, Gilles Doumy, Anne Marie March, Éva Bajnóczi, Zoltán Németh, Mátyás Pápai, Emese Rozsályi, Dorottya Sárosiné Szemes, Hana Cho, Sriparna Mukherjee, Chang Liu, Tae Kyu Kim, Robert W. Schoenlein, Stephen H. Southworth, Linda Young, Elena Jakubikova, Nils Huse, György Vankó, Christian Bressler, and James K. McCusker. *Using ultrafast x-ray spectroscopy to address questions in ligand-field theory: The excited state spin and structure of [fe(dcpp)2]2+*. Inorganic Chemistry, 58(14):9341–9350, 2019. PMID: 31241335.

2017

Anne Marie March, Tadesse A. Assefa, Christina Boemer, Christian Bressler, Alexander Britz, **Michael Diez**, Gilles Doumy, Andreas Galler, Manuel Harder, Dmitry Khakhulin, Zoltán Németh, Mátyás Pápai, Sebastian Schulz, Stephen H. Southworth, Hasan Yavaş, Linda Young, Wojciech Gawelda, and György Vankó. *Probing transient valence orbital changes with picosecond valence-to-core x-ray emission spectroscopy*. The Journal of Physical Chemistry C, 121(5):2620–2626, 2017.

2016

Alexander Britz, Tadesse A. Assefa, Andreas Galler, Wojciech Gawelda, **Michael Diez**, Peter Zalden, Dmitry Khakhulin, Bruno Fernandes, Patrick Gessler, Hamed Sotoudi Namin, Andreas Beckmann, Manuel Harder, Hasan Yavaş, and Christian Bressler. *A multi-MHz single-shot data acquisition scheme with high dynamic range: pump–probe X-ray experiments at synchrotrons*. Journal of Synchrotron Radiation, 23(6):1409–1423, Nov 2016.

2015

Stephanie Bley, **Michael Diez**, Friederike Albrecht, Sebastian Resch, Siegfried R. Waldvogel, Andreas Menzel, Margit Zacharias, Jürgen Gutowski, and Tobias Voss. *Electron tunneling from colloidal CdSe quantum dots to ZnO nanowires studied by time-resolved luminescence and photoconductivity experiments*. The Journal of Physical Chemistry C, 119(27):15627–15635, 2015.

Acknowledgements

At first, I would like to thank my supervisor Christian Bressler for giving me the opportunity to become a member of the FXE group at European XFEL and working in this international and exciting company. I appreciate the freedom and trust he gave me, even to change the initial topic of my PhD project. Even during the stressful times at XFEL with an opening beamline and the last 1 1/2 years with COVID19, he always reached out to me and was interested in my progression of the thesis. During endless meetings and discussions about the results, he provided crucial feedback and new perspectives on the ongoing analysis and interpretation. I would like to thank my second supervisor Michael Rübhausen for the fruitful discussions and insightful recommendations to improve my thesis. I also express my gratitude to Daniela Pfannkuche and Franz Kärtner for agreeing to be part of the Promotionskomitee to evaluate my defence.
The thesis would not have been possible without Andreas Galler. He introduced me to the FXE laser lab and taught me many valuable and handy tricks. He also invited me to the first timing tool beamtime at LCLS where the original setup used in this thesis was tested, and, therefore he majorly influenced the trajectory of this thesis. I also greatly enjoyed his company during many beamtimes and nightshifts around the world. The same holds true for Sebastian Schulz whom I had the privilege to work with closely for two years in the FXE laser lab. These were fun times! Without you, the SACLA beamtime wouldn't have happened.
Many thanks go to my co-suffering PhD fellows and friends Christina Bömer, Florian Otte, Tae Kyu Choi, Manuel Harder, Alexander Britz and Tadesse Abebaw Assefa. You guys made the past years a lot easier and more enjoyable! The beamtimes we had together were a lot of work and stress, but the fun we had together outweighed these downsides by a lot. I am happy that there were the SQS students Jones Rafipoor, Alexander Achner and Rene Wagner, who were also working in the laser lab. It was a lot easier to sit the whole day in darkness with you guys and watch some Tour de France mountain stages together. I would also like to thank Frederico Lima for the fun times at work and outside and for his awesome cheese bread. A special thanks is going to the fun office of the FXE Technicians and Engineers Paul Frankenberger, Martin Knoll and Timo Korsch, where always a NERF gun battle was waiting for the innocent office invaders. In addition, I would like to thank Peter Zalden, Dmitry Khakhulin, Wojciech Gawelda, Mykola Biednov, Katharina Kubicek and Patrik Grychtol for the great discussions, fun times and always answering my stupid questions with great endurance. I would also like to thank the numerous people who contributed to this thesis and are not mentioned here, but were

involved in one of the many beamtimes and the technical and organizational staff from DESY and European XFEL who ensured a fast solution to all problems.
A very special thanks goes to my family for always supporting and believing in me what ever I chose to do. Thank you Moni and Fred!
At last, I want to thank the person who is the most important person in my life, Helen Bieker. Without you, I would have never finished this thesis. Thank you for always being there, helping me, cheering me up, giving good advise, despite all your own problems and suffering through the same PhD adventure as I do. I love you.