TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Optimising a z-scan experiment with an acousto-optic modulator: Reducing thermal effects and improving beam stability

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Physikalische Energie und Messtechnik

eingereicht von

Tomas Zwettler
Matrikelnummer 01326791

ausgeführt am Institut für Angewandte Physik
der Fakultät für Physik der Technischen Universität Wien
(in Zusammenarbeit mit dem Institut für Werkstoffwissenschaft und Werkstofftechnologie)

Betreuung
Betreuerin: Associate Prof. Dipl.-Ing.in Dr.in techn. Ille Gebeshuber
Mitwirkung: Univ.Prof. Dr.rer.nat. Aleksandr Ovsianikov

Wien, 18.04.2024

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Danksagung

Ich hätte diese Arbeit nicht ohne die Hilfe mehrere Personen vollenden können. Mein erster Dank gilt **Associate Prof. Dipl.-Ing.in Dr.in Ille Gebeshuber**, welche meine Arbeit von Seite der Physik betreut hat. Auch wenn ich mehr Zeit am Institut für Werkstoffwissenschaft und Werkstofftechnologie verbracht habe, war ich immer bei ihr willkommen und habe die Zeit mit ihr und in ihrer Gruppe sehr genossen. Sowohl ihr Wissen zu wissenschaftlichen Arbeiten als auch Gespräche außerhalb der Universität haben mir beim Schreiben dieser Arbeit geholfen und werden mich sicher auch im weiteren Leben immer wieder begleiten.
Als nächstes möchte ich **Univ.Prof. Dr. Aleksandr Ovsianikov** danken, der mich auf den Getreidemarkt geholt und mir die Möglichkeit gegeben hat, Teil seines unglaublich freundlichen Teams *3D Printing and Biofabrication* zu sein. Er hat es mir ermöglicht, intensiv mit Lasern zu arbeiten. Da ich während meines Studiums auf jeden Fall mit Kernkraftwerken und Lasern hantieren wollte, hat es mich sehr gefreut, im Laufe dieser Arbeit auch den zweiten Punkt von meiner Wunschliste zu erfüllen.
Ohne **Projektass.in Dipl.-Ing.in Dr.in Franziska Chalupa-Gantner** hätte es diese Arbeit nicht geben können. Sie hat mir alles im Labor gezeigt, was ich wissen musste. Wenn ich Fragen zum theoretischen Hintergrund hatte, hat sie mir auch immer weiterhelfen können. Sicherlich ist sie die Person, welche diese Arbeit am öftesten gelesen hat und ihre Anmerkungen und Vorschläge waren mir eine große Hilfe.
Weiteres möchte ich allen Mitgliedern aus dem 3D-Druck Team danken, die mir immer wieder bei diversen Problemen weitergeholfen haben. Sei es **Mag.a Dr.in Marica Markovic** die mir gemeinsam mit **Dipl.-Ing. Oliver Kopinski-Grünwald** öfters im Chemie-Labor geholfen hat oder **Projektass. Dipl.-Ing Dr. Stefan Binder** der mich sowohl im Laser-Labor als auch bei diversen Programmier-Problemen unterstützt hat. Ebenfalls möchte ich **Dr. Hamad Syed** danken, der mir eine große Hilfe war, die gewonnen Z-Scan Daten auszuwerten und zu interpretieren.
Mein Dank geht auch an **Dipl.-Ing Dr. Wolfgang Steiger**, dessen Dissertation ich oft gelesen habe. Sein Z-Scan Experiment war das Grundgerüst, auf dem ich aufgebaut habe. Die Kommentare in seinem Python Code haben mir ein sehr sympathisches Bild von ihm vermittelt, welches sich dann bestätigt hat, als ich ihn bei einer Institutsfeier getroffen habe.

Als letztes möchte ich noch einigen Personen danken, die mir zwar nicht inhaltlich bei dieser Arbeit geholfen haben, aber ohne die ich diese Arbeit möglicherweise nicht fertig geschrieben hätte. Allen voran meiner Familie: Meiner Partnerin Fabienne, die mich fast meine gesamte Studienzeit begleitet hat und mir immer zur Seite steht.
Meine Eltern und meine Geschwister Christina und Michi haben mich nicht nur während dieser intensiven Zeit unterstützt, sondern haben mir schon mein gesamtes Leben lang in unzähligen Situationen geholfen.
Zuletzt möchte ich auch meinen Freunden aus dem Studium danken. Sie haben mir über mein ganzes Studium hinweg immer wieder geholfen und sind damit mit verantwortlich, dass diese Arbeit geschrieben wurde.

Abstract

This thesis presents a redesign and enhancement of an existing z-scan setup, focusing on improving beam stability and minimising thermal effects. Z-scan experiments are commonly employed for characterising higher order nonlinearities in thin samples using laser beams. However, the use of pulsed lasers with repetition rates exceeding several tens of kilohertz introduces thermal effects that require careful management, typically through the incorporation of optical choppers.

This study investigates the feasibility of integrating an acousto-optic modulator (AOM) into a setup requiring movable mirrors. Additional optical components were incorporated to facilitate quicker and more precise alignment. Experimental results demonstrate that utilising a movable mirror on a stage yields a sufficiently stable beam path to accommodate an AOM without necessitating readjustment after mirror movements. Moreover, replacing the previously employed optical chopper with an AOM allows for exploring the effects of minor variations in laser beam chopping frequency on the thermal effects during z-scan measurements. This substitution also streamlines pre-experiment calibrations.

It showed, that variations in chopping frequency and sample illumination time exhibit subtle impacts on thermal effects, with noticeable changes observed only when the sample illumination duration surpasses a certain threshold. However, the influence of different chopper settings on thermal effects is found to be minimal compared to fluctuations arising from external factors such as laboratory temperature variations across different days. Nevertheless, the AOM proves to be advantageous compared to the previously utilised mechanical chopper, as it expedites experiment setup and enhances versatility by allowing for measurement of additional variables. Furthermore, the AOM facilitates novel measurement methods without necessitating extensive experiment reconfiguration.

Overall, this thesis demonstrates that the integration of an AOM into a z-scan setup facilitates the straightforward examination of the effects of various chopper settings on thermal characteristics, while also expediting alignment procedures. Additionally, the development of multiple Python programs during this study enables automated analysis of z-scan measurements.

Kurzfassung

In dieser Arbeit wurde ein vorhandenes Z-Scan Experiment umgebaut und optimiert. Z-Scan Experimente sind eine Standard Methode um nicht lineares Verhalten in optischen Medien zu charakterisieren. Bei der Verwendung von gepulsten Lasern mit mehr als einigen zehn Kilohertz Pulsen treten thermische Effekte auf, die minimiert werden müssen. Das kann mit einem optischen Chopper erreicht werden.

Im Laufe dieser Arbeit wurden mehrere optische Elemente in das Z-Scan Experiment eingeführt, beziehungsweise ausgetauscht, um das Experiment schneller einzustellen und den Strahlengang zu stabilisieren. Außerdem wurde getestet, ob der vorhandene optische Chopper mit einem akusto-optischen Modulator (AOM) ersetzt werden kann. Dadurch konnte getestet werden, wie sich Änderungen in der Chopper Frequenz auf die thermischen Effekte in der Probe auswirken. Außerdem erlaubt der AOM ein einfacheres Einstellen des Experiments zu Beginn einer neuen Messreihe.

Die Experimente zeigten, dass der AOM in einem System mit bewegbarem Spiegel gut als Chopper verwendet werden kann. Solange die Belichtungszeit der Probe einen Grenzwert unterschreitet, haben Änderungen in der Chopper Frequenz und Belichtungszeit geringere Auswirkungen auf die thermischen Effekte, als äußere Veränderungen wie die Labortemperatur. Der AOM ist dennoch günstiger als der zuvor verwendete mechanische Chopper, da das Experiment schneller aufgebaut werden kann und vielseitiger wird, da weitere Variablen gemessen werden können. Außerdem ermöglicht der AOM neue Messmethoden ohne das Experiment umbauen zu müssen.
Zusätzlich wurden mehrere Python Programme geschrieben, die die gemessenen Daten von zwei verschiedenen Z-Scan Methoden auswerten und vergleichen.

Abbreviations

2PA	Two-photon absorption
2PP	Two-photon polymerisation
α	Absorption coefficient
β	Two-photon absorption coefficient
σ_2	Two-photon absorption cross section
AOM	Acousto-optic modulator. Sound waves generate a diffraction grating, which modulates the incoming laser light.
Avogadro Number	Number of units in one mole of a substance. $N_A = 6.022\,140\,76 \times 10^{10}\,\mathrm{mol}^{-1}$
PI	Photoinitiator
Beam waist	Diameter of a Gaussian beam in the focal point. w_0
Cuvette	A straight-sided clear container for holding liquid samples.
Deep See module	Part of the utilised laser hardware. Two prism that can be adjusted to minimise the pulse duration of the laser at each wavelength.
Goepert-Mayr	Unit of σ_2, $1\,\mathrm{GM} = 1 \times 10^{-50}\,\mathrm{cm}^4\,\mathrm{s}\,\mathrm{photon}^{-1}\,\mathrm{molecule}^{-1}$
Rayleigh length	The distance along the laser beam from the beam waist until the cross section of the laser is doubled. $z_0 = n\pi w_0^2/\lambda$.

Contents

1 Introduction **1**
1.1 Multiphoton absorption . . . 1
1.2 Nonlinear refractive index . . . 4
1.3 Z-scan . . . 4
1.3.1 Open aperture z-scan . . . 6
1.3.2 Closed aperture z-scan . . . 7
1.4 Acousto-optic modulator . . . 10
1.5 Two-photon polymerisation . . . 11
1.6 Scope of this thesis . . . 13

2 Materials and Methods **14**
2.1 Z-scan setup . . . 14
2.2 AOM installation . . . 16
2.3 Z-scan procedure . . . 16
2.3.1 System calibration . . . 17
2.4 Chemical structure of the sample . . . 19

3 Results **20**
3.1 Acoustic optic modulator . . . 20
3.1.1 AOM position . . . 20
3.1.2 AOM stability . . . 22
3.2 Beam parameters . . . 24
3.3 Chopper settings . . . 24
3.4 Open aperture measurements . . . 26
3.5 Closed aperture measurement . . . 29

4 Discussion **31**
4.1 AOM efficiency and stability . . . 31
4.2 AOM as a chopper . . . 32
4.2.1 Chopper frequency . . . 32
4.3 Conclusion and outlook . . . 33

Appendices **35**

A Appendix - Tables **35**

B Appendix - Figures **39**

C Appendix - Figures **41**
C.1 Analyse_OA.py . . . 41
C.2 Analyse_CA.py . . . 46
C.3 AOM_Opening_times.py . . . 50

1 Introduction

The z-scan experiment was developed in 1990 [1, 2] and has become a standard method to characterise higher order nonlinearities [3–9] such as the two-photon absorption (2PA) or the nonlinear refractive index n_2.
One of the uses of the z-scan technique is to analyse the nonlinear absorption of photoinitiators (PIs).

A PI is a molecule that absorbs photons from a light source and creates reactive species out of the excited state [10]. Those reactive species can be cations, anions or free radicals.

The z-scan method is in principle a straightforward technique. A sample is moved through the focal point of a focused laser beam and the intensity of the beam is measured with a photo diode before and after the sample. With the photo diode before the sample any fluctuations of the laser can be detected. When the sample is moved through the focal point of the laser the laser intensity measured by the photo diode after the sample will change. This change in intensity can be used to calculate nonlinearities of the sample.

Although the experiment is simple in design, it is very sensitive to any environmental changes and fluctuations of the laser that can alter the beam path in a small way. Many z-scan experiments use high repetition rate ultra fast lasers [11–17]. These kinds of lasers offer many advantages, such as good pulse to pulse stability and maintainability [18]. However high repetition rate lasers lead to pulse pile up and to cumulative thermal effects [18–20]. These effects can influence the measurement in a way that is not accounted for in the utilised calculation method. This leads to falsification of the calculated data due to thermal effects. To be able to utilise the conventional z-scan equations the thermal effects need to be minimised using a blanking technique like a chopper.
Using a tuneable laser it is possible to perform spectral scans of a sample, without the need to implement additional components to the setup. Spectral scans are very important to characterise new PIs. The knowledge of the 2PA absorption coefficient at various wavelengths allows to optimise two-photon polymerisation (2PP) experiments. By using a light source at a wavelength with a large σ_2 of the PI lower power or shorter illumination time is needed for 2PP processes.

It is also advantageous, that the z-scan technique can be used to characterise photoinitiators with the same laser, that can be used for 2PP 3D printing. This reduces cost since only one laser is needed for two (or more) different experiments. It also guarantees that the sample is measured with the same parameters that are later used for 3D printing.

1.1 Multiphoton absorption

Photon absorption is a transition from a lower to a higher electronic state of a molecule. When an incident photon strikes an electron in a lower energy orbital the electron can absorb the energy of the photon and causes the electron to temporarily spin faster. This transfer of transverse energy to longitudinal energy increases the wave amplitude between the electron and the nucleus, forcing the electron away to the new point of minimal amplitude and it reaches an excited state. This can excite the electron from the ground state S_0 to a higher singlet state. Due to conservation of energy, direct excitation to a triplet state is not possible. If the photon frequency does not match the electrons spin frequency of the wave centre, the photon can not be absorbed by the electron. A single photon is not able to excite an electron to an excited state, if the energy of the photon is

less than the energy gap between the ground state and the excited states. However, it is possible, that multiple photons are absorbed almost simultaneously by the same molecule to excite it to a higher energy state, if their combined energy is equal to the energy gap.

Multiphoton absorption was first described by Maria Goeppert-Mayer in 1931 [21] and proven by Kaiser and Garrett in 1961 [22]. In the case of 2PA a first photon excites a molecule to a short lived virtual state. This virtual state only exists for a few femto seconds. If the molecule absorbs a second photon within the lifetime of this virtual state it can reach its excited state. This process requires high photon densities, which can be achieved by focusing a pulsed laser providing a pulse duration in the fs-range.

One way of visualising the electronic states of a molecule and the transitions between these states is the Jablonski diagram. It was first published by Aleksander Jabłoński in 1933 [23] with contributions by Jean Baptiste Perrin and Francis Perrin [24]. Figure 1 shows the Jablonski diagram [25–27]. In addition to the transitions commonly shown in a Jablonski diagram, 2 photon absorption and the generation of radicals is also pictured[28].

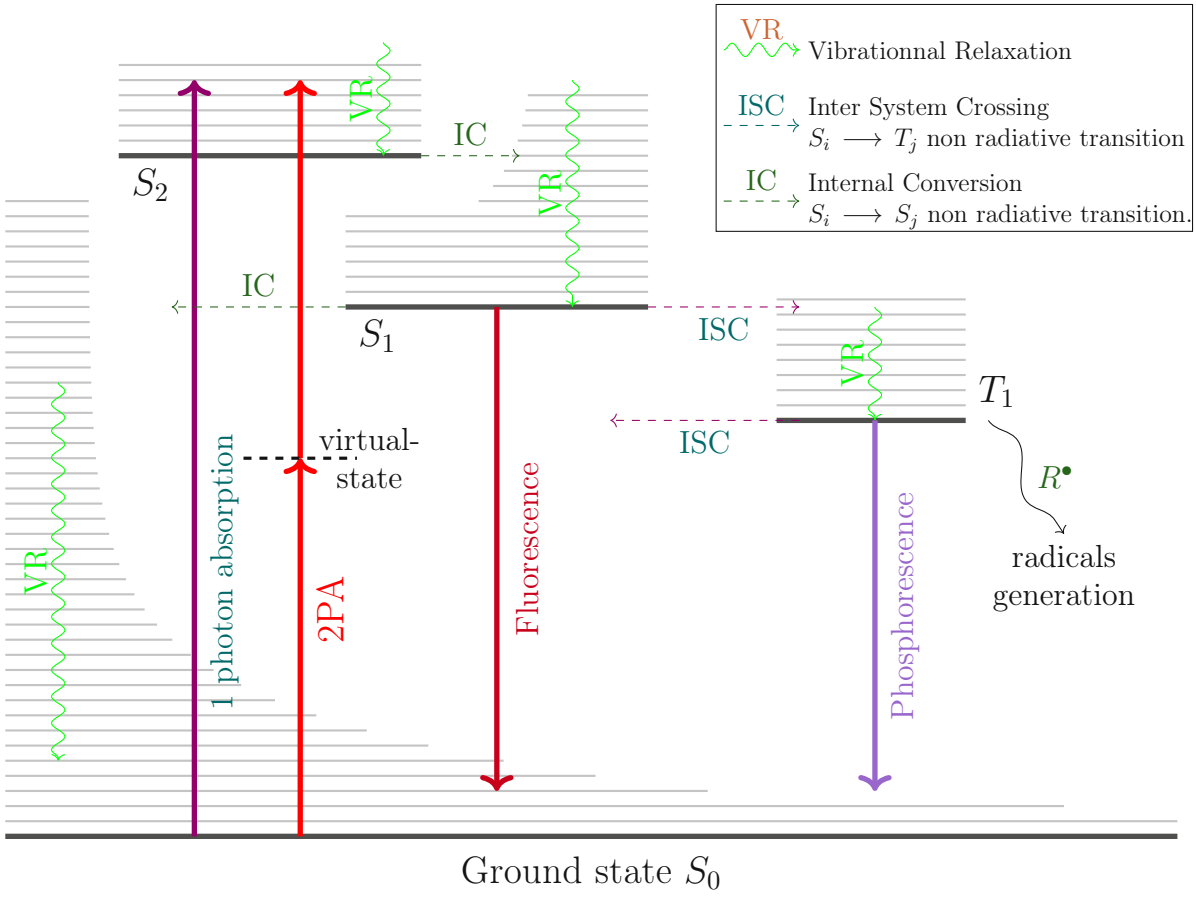

Figure 1: *Adapted Jablonski diagram for organic molecules. It shows multiple energy states and possible transitions between them. The ground state (S_0) is shown on the bottom of the diagram with higher energy states above it. Two excited singlet (S_1, S_2) and one triplet (T_1) state are depicted. Also shown is the generation of radicals. Straight arrows symbolise radiative and wavy arrows non-radiative transitions. The only process to a higher energy state is through absorption. In the case of 2 photon absorption the electron is first excited to a very short lived virtual state from where it is excited by another photon to S_2. Multiple possibilities of crossing to lower energy states are depicted.*

The vertical axis represents energy with the ground state S_0 on the bottom and the higher states above it. Shown are two excited singlet states S_1 and S_2 as well as one triplet state T_1. The transitions between states are given by arrows. Straight arrows represent radiative transitions and wavy arrows non-radiant transitions. Radiative transitions are any transitions that involve photons by either emitting or absorbing them.

The fastest process is absorption occurring on a timescale of 10^{-15} s. This occurs when a photon is absorbed and excites the molecule to a higher state. In the case of two-photon absorption two photons are absorbed almost simultaneously and their combined energy excites the molecule to the higher energy state. The second fastest process is energy loss due to vibrational relaxation. This non radiative transition looses excess vibrational energy to vibrational modes within the same molecule or to surrounding molecules, until the lowest vibrational level of the electronic state is reached. This process takes $10^{-12} - 10^{-10}$ s. Another possible non radiative transition is the internal conversion from a higher level singlet state to a lower level singlet state. The rate of this transition is inversely proportional to the energy gap between the two electronic states. The last of the non radiative transitions is the intersystem crossing from one singlet or triplet state to a triplet or singlet state. For this to happen the spin of the electron needs to change. One way that this can happen, is by a change in orbital configuration. The angular momentum of the different orbials is compensated by the angular momentum change of the spin. This process ist called spin-orbit coupling. The intersystem crossing competes with other S_1 depopulation transitions. In most molecules the most common one is fluorescence. In this radiative transition the electron falls back to the ground state S_0 and emits a photon. The third radiative transition in addition to absorption and fluorescence is phosphorescence. Since this transition from the T_1 to the S_0 state is like the inter system crossing only possible due to spin-orbit coupling it occurs on a much larger timescale than fluorescence. Typical phosphorescence lifetimes are in the range of $10^{-6} - 10$ s, while fluorescence occurs within $10^{-10} - 10^{-7}$ s [29, 30].

Another possible process is the generation of radicals[31]. In some molecules a covalent bond may break, when it is in an excited state due to its increased energy. This can either be homolytic bond cleavage or heterolytic bond cleavage[32, 33]. In the former the covalent bond is broken symmetrically and two radicals are generated. In the latter one of the two atoms involved in the bond retains both electrons and the other one becomes a radical. Depending on the molecule this can occur in either excited singlet or triplet states.

The generation of free radicals due to homolytic bond cleavage, also called homolysis, is a very important process in polymerisation processes which is explained in greater detail in subsection 1.5.

Two-photon absorption can be used in numerous applications such as investigation of biological systems, optical power limiting, 2-Photon microscopy or 3D microfabrication [34–36]

1.2 Nonlinear refractive index

The nonlinear refractive index (n_2) describes the change of the refractive index (Δn) induced by optical intensity (I):

$$\Delta n = n_2 \cdot I. \tag{1}$$

This change is due to the optical Kerr effect [37–39]. This effect is responsible for the nonlinear optical effects of self-(de)focusing, self-phase modulation and modulational instability.
The intense beam of light acts as a modulating electric field and changes the refractive index,

$$|E(t)| = \Re\{E(t) \cdot e^{i(kr-\omega t)}\}. \tag{2}$$

With the wave vector $k = 2\pi/\lambda$, the distance r, the angular velocity $\omega = 2\pi f$ and the time t. The electric field varies slowly over time. For a single frequency input the field can be approximated in the external self-action limit [39, 40]. This is only applicable if the sample length L is thin compared to the Rayleigh length z_0. The polarisation is given by

$$\begin{aligned} P(t) =& \epsilon_0 \int_{-\infty}^{\infty} \chi^{(1)}(t-t_1)E(t_1)dt_1 + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \chi^{(2)}(t-t_1,t-t_2)E(t_2)dt_1dt_2 + \\ &+ \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \chi^{(3)}(t-t_1,t-t_2,t-t_3)E(t_1)E(t_2)E(t_3)dt_1dt_2dt_3 \end{aligned} \tag{3}$$

where ϵ_0 is the vacuum permittivity and $\chi^{(n)}$ describes the n-th order response or susceptibility function [41]. In a linear medium only the first term is relevant. For the Kerr effect even ordered terms can be dropped due to inversion symmetry of the Kerr medium and only the first and third order are significant.

$$P(t) = \epsilon_0 \left[\chi^{(1)} + \chi^{(3)}\frac{|E(t)|^2}{2}\right] \cdot E(t) = \epsilon_0 \chi E(t) \tag{4}$$

with the linear refractive index $n_0 = (1+\chi^{(1)})^{1/2}$, the following formula:

$$n = (1+\chi)^{1/2} = (1+\chi^{(1)}+\chi^{(3)})^{1/2} \approx (1+\frac{\chi^{(3)}}{2n_0^2}) \cdot n_0, \tag{5}$$

and a Taylor expansion of $\chi^{(3)}$ can be used to formulate equation 1:

$$n = n_0 + \frac{\chi^{(3)}}{2n_0}|E_t|^2 = n_0 + n_2 \cdot I \tag{6}$$

Since most materials have a relatively small second-order nonlinear refractive index with n_2 in the order of $10^{-20}\,\mathrm{m^2/W}$, beam intensities in the order of $1\,\mathrm{GW/cm^2}$ are necessary to produce significant variations in the refractive index due to the Kerr effect.

1.3 Z-scan

To perform z-scan measurements, a thin sample (sample length L is smaller or equal to the Rayleigh length $z_0 = n\pi w_0^2/\lambda$ of the focused laser beam) is placed in a holder on a single axis stage (usually called the z-axis, hence the name of the technique) and moved in and

out of the focal plane of a focused Gaussian laser beam (see figure 2). Two photo diodes measure the intensity of the laser beam. Before the focusing lens a beam splitter partly diverts the beam to the first photo diode to detect fluctuations in the beam. The second photo diode measures the intensity after the sample. When the sample is moved through the focus of the laser beam, the intensity measured at the second photo diode changes due to thermal and nonlinear effects within the sample. An aperture before photo diode 2 can be closed to let only a fraction of the intensity through in a so called closed aperture (CA) measurement, or left open for an open aperture measurement (OA). The former allows for measurement of n_2 and thermal effects while the latter is used to measure the two-photon absorption coefficient β. Both are explained in greater detail further down, OA z-scan in subsubsection 1.3.1 and CA z-scan in subsubsection 1.3.2.

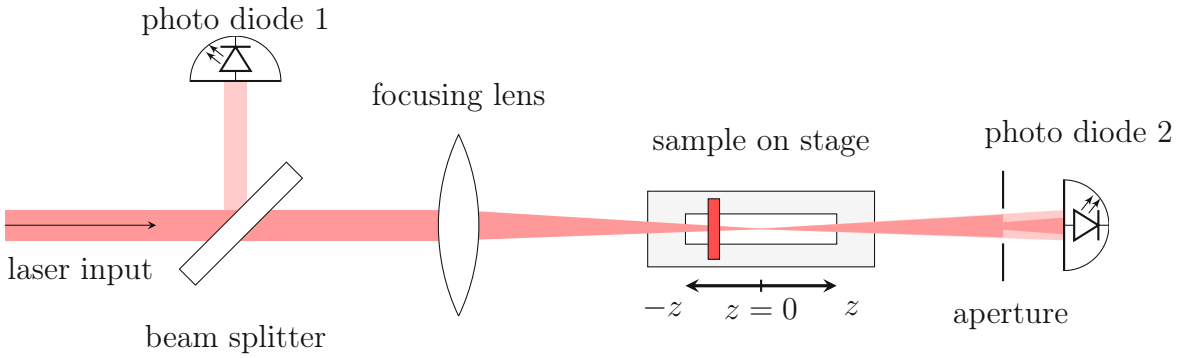

Figure 2: *Illustration of the z-scan setup. The laser light enters the setup from the left. A part of it is diverted at a beam splitter to photo diode 1. The majority of the light is focused by a lens. The sample is placed on a linear stage and moved through the focal point of the laser beam. The intensity of the light is measured by photo diode 2. An aperture can shield some of the light that reaches the second photo diode to perform closed aperture measurements.*

One of the parameters of a sample that can be calculated using measurements of an open aperture z-scan technique is the 2PA cross section σ_2. It is a scale for the probability that 2PA occurs and is measured in Goeppert-Mayr Units ($1\,\mathrm{GM} = 1 \times 10^{-50}\,\mathrm{cm}^4\,\mathrm{s}\,\mathrm{photon}^{-1}\,\mathrm{molecule}^{-1}$). Which is defined as: *With a photon flow of 1 photon per second and square centimetre in a material of 1 molecule per cubic centimetre one out of* 10^{50} *photons will be absorbed when* σ_2 *is one GM.*
Values above 100 GM are considered high [42, 43].

There are multiple methods to perform a z-scan. Most set-ups are based on optical parametric amplifiers operating at a single wavelength and are therefore not practical to measure a 2PA spectrum. A common method to gather spectral data is 2-photon excited fluorescence [44]. This technique is limited by the necessity of using samples with fluorescent behaviour. Additionally the extraction of absolute values depends on the reference standard or requires a complex setup [45]. A spectral z-scan method is white-light continuum z-scan, which is able to determine absolute cross section values. However this system also requires a complicated optical path [46, 47]. In this thesis, a tuneable pulsed fs-laser was utilised. Using a tuneable laser offers the possibility to perform a spectral z-scan. However, due to the high repetition rate thermal effects are likely to occur and have to be eliminated as much as possible [48].

1.3.1 Open aperture z-scan

When the aperture as shown in figure 2 is open completely, all of the transmitted light reaches the second photo diode, which allows for direct measurement of the 2PA coefficient β. The sample is placed in a cuvette on the stage. To be able to neglect linear and nonlinear refraction within the sample, a so called optically thin sample is necessary. A sample is optically thin when the sample thickness L is smaller than the Rayleigh length z_0. When the sample is moved the measured transmission changes. Due to absorption in the sample, the measured transmission is lowest at the focal plane, where the intensity of the Gaussian beam is greatest. When the sample position $|z| >> z_0$ the beam intensity is too weak to induce nonlinear effects. The normalised energy transmittance, depended on the position z of the sample is given by [2]

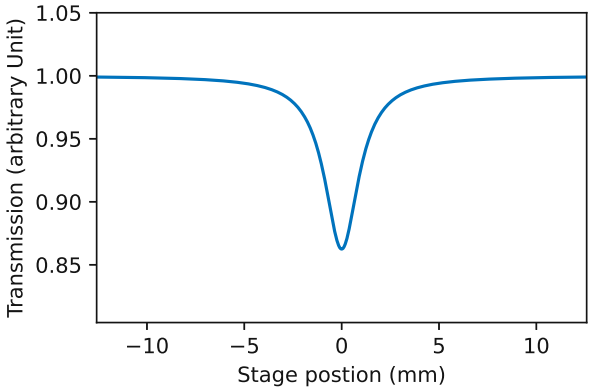

Figure 3: *Example of a transmittance curve of an OA z-scan. The measured trasnmittance decreases when it reaches the focal plane, afterwards it increases again.*

$$T(z) = \frac{1}{\sqrt{\pi} q_0(z,0)} \cdot \int \ln(1 + q_0(z,0) e^{-\tau^2}) \mathrm{dt} \tag{7}$$

with the fit parameter

$$q_0(z) = \frac{\beta I_0(w_0, P) L_{\text{eff}}}{1 + \frac{z^2}{z_0^2}} \tag{8}$$

and the prarameters:

α	linear absorption coefficient,
L	sample length,
$\beta(\sigma_2)$	2PA coefficient,
$I_0(w_0, P)$	intensity distribution cf. equation 9,
P	laser power,
R	repetition rate,
τ	pulse duration,
$L_{\text{eff}} = \frac{1 - e^{-\alpha L}}{\alpha}$	effective cuvette length,
z	sample position,
$z_0 = n\pi w_0^2 / \lambda$	Rayleigh length and
ω_0	beam waist.

$$I_0(\omega_0, P) = 4 \cdot \sqrt{\frac{\ln 2}{\pi}} \cdot \frac{P}{\pi \cdot \omega_0^2 \cdot R \cdot \tau}, \tag{9}$$

For $|q_0| < 1$ equation 7 can be expressed in terms of the peak irradiance in a summation form more suitable for numerical evaluation:

$$T_{\text{OA}}(z) = \sum_{m=0}^{\infty} \frac{-q_0(\sigma_2, z, 0)^m}{(m+1)^{3/2}}. \tag{10}$$

To calculate the 2PA cross section it is necessary to know the exact parameters of the utilised laser beam. Those parameters are the pulse width, the power of the utilised laser and the beam profile. They need to be measured and calibrated before a new measurement is performed. When they are known, and β is measured by fitting equation 10 to the gathered z-scan data, the 2PA absorption cross section can be calculated:

$$\sigma_2 = \sigma_2(\lambda) = \frac{hc}{\lambda} \cdot \frac{\beta(\lambda)}{N_A \cdot \rho}. \tag{11}$$

with the prarameters:

λ	wavelength,
$\frac{hc}{\lambda}$	photon energy,
N_A	Avogadro Number,
ρ	concentration of the solution in mol/litre.

1.3.2 Closed aperture z-scan

Closed aperture (CA) z-scan allows to measure the nonlinear refractive index n_2 as well as the thermo-optic coefficient. To perform CA measurements the aperture before photo diode 2, as shown in figure 2 has to be closed to only allow a small fraction of the transmittance S. Generally, the fraction transmitted by the aperture in the CA z-scan is between 10–50 %.

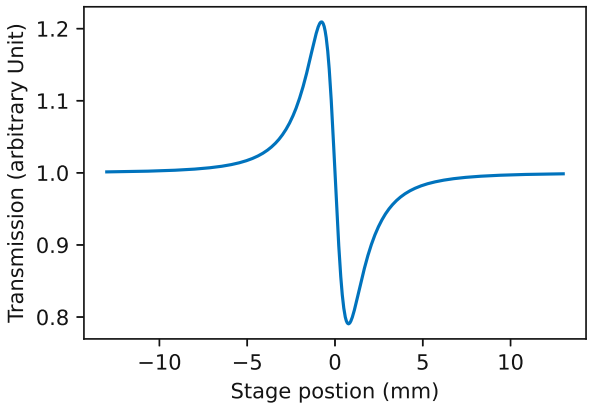

Figure 4: *Example of a transmittance curve of a CA z-scan. The measured transmittance increases before it reaches the focal plane, then it decreases. Farther away from the focal plane it is linear.*

When the scan is started with the sample at $-z$ the transmittance is constant due to the low beam irradiance. The measured intensity increases when the sample approaches the focal plane, leading to negative self-lensing prior to the focus. This collimates the beam and causes a beam narrowing at the aperture which results in an increase in the measured transmittance. When the sample passes the focal plane the beam self defocuses and increases the beam divergence. This leads to a broadening at the aperture and therefore decreases the transmittance. The sample then moves further to $+z$ and the transmittance becomes linear again.

To calculate the nonlinear refraction index n_2 the following approximation of the transmission curve can be used [49]:

$$T_{\mathrm{CAr}}(z) \approx 1 - \frac{4\Delta\varphi_0 \frac{z}{z_0}}{\left(\frac{z^2}{z_0^2} + 1\right)\left(\frac{z^2}{z_0^2} + 9\right)} \tag{12}$$

As in equation 8, z is the position of the sample and z_0 is the Rayleigh length. $\Delta\varphi_0$ is the induced phase distortion of the Gaussian beam of the laser after being passed through the sample. It correlates with the nonlinear refractive index as given by the following

equation:

$$n_2 = \frac{\lambda \Delta\varphi_0}{2\pi I_0 L_{\text{eff}}}. \tag{13}$$

The same CA measurement used to calculate the nonlinear refractive index, can also be used to measure refractive index changes due to density variations induced by thermal lensing [50, 51]. Those effects occur when a pulsed laser with a repetition rate higher than a few tens of kHz are utilised [51]. With CA z-scan the thermal 2PA coefficient β_{therm} can be calculated and compared with the 2PA coefficient β. If both are of the same magnitude it is safe to assume that thermal effects falsify the measured σ_2.

To calculate the thermal optical coefficient and the thermal 2PA coefficient it is necessary to model the temperature profile first. Those calculations have been done in multiple papers, both for linear absorption [52, 53] and for multi photon absorption [54]. The temperature profile is given by [7]:

$$\Delta T(z,r,t) = \eta h \frac{c}{\lambda} N \sigma R \left(\frac{2}{\pi w^2(z)}\right)^{\eta-1} \frac{H(\eta)}{4\pi\kappa\eta} \cdot \left[\text{Ei}\left(\frac{-2\eta r^2}{w^2(z)}\right) - \text{Ei}\left(\frac{-2\eta r^2}{w^2(z)} \cdot \frac{1}{1+2\eta t_d}\right)\right], \tag{14}$$

where η is the order of the multi photon absorption process, N is the density of absorbing centres, w is the spot size of the laser at a given sample position z. $H(\eta)$ is the integration over the pulse duration of the photon power as given in:

$$H(\eta) = \int_0^\tau P(t)^\eta dt = \sqrt{\frac{2}{\pi}} \frac{E_p^2}{\tau}. \tag{15}$$

The temperature profile is also dependent on the thermal conductivity of the sample κ and the exponential-integral function [55],

$$\text{Ei}(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t} dt = \int_{-\infty}^{x} \frac{e^t}{t} dt, \tag{16}$$

as well as the time quotient $t_d = t/t_c$ which depends on the thermal characteristic time $t_c = w^2/4D$ where D is the thermal diffusion coefficient of the sample material.

In equation 15 the photon dependent energy can be expressed as

$$E_p = \frac{E}{h\nu} \quad \text{and} \quad E = \frac{P}{R} \tag{17}$$

with the pulse Energy E. Using $\nu = c/\lambda$ and inserting equation 17 in equation 15 gives H in dependency of η and known laser values:

$$H(\eta) = \sqrt{\frac{2}{\pi}} \cdot \frac{(P\lambda)^2}{(Rhc)^2 \tau}. \tag{18}$$

The thermo-optical coefficient $\mathrm{d}n/\mathrm{d}T$ produces a refractive index profile of the temperature profile in equation 14. The incoming Gaussian beam is multiplied by the thermo-optical phase factor to calculate the propagation of the laser beam in the sample.

$$G_o(z,r,t) = G_i(z,r) \exp\left(-\mathrm{i}k \frac{\mathrm{d}n}{\mathrm{d}T} \Delta T(z,r,t) L\right), \tag{19}$$

with:

G_o	amplitude of the Gaussian beam at the sample exit plane,
G_i	amplitude of the Gaussian beam at the sample entrance plane,
k	wavevector of the light
L	sample length.

Next the thermal lens signal can be calculated. Which is the intensity on the field due to the propagation of the field (19) on the axis and normalised to the intensity calculated at $t = 0$ when no thermal lensing occurs. This signal can be calculated by numerical evaluation of the propagation [56]. This integral can be formulated as [57]:

$$E(r = 0, z, t) = \frac{2\pi}{i\lambda(d - z)} J_0(0) \cdot \int_0^\infty r' G_o(z, r', t) \exp\left(\frac{i\pi r'^2}{\lambda(d - z)}\right) \mathrm{d}r'. \tag{20}$$

Where d is the position of the detector plane and J_0 is the Bessel function.
Another method [58] is to use the Fraunhofer region approximation for the propagation integral, and use a first-order series expansion of the exponential in equation 19 to linearise the thermo-optical phase factor. This allows to obtain an analytical form of the signal and utilise a formula, dependent on the position z of the sample without time dependency.

With the temperature profile given by equation 14 the parameter ϑ as defined by Sheldon et al. [58], which represents the thermal lens strength, can be generalised for multiphoton processes of arbitrary order η to [7]:

$$\vartheta(\eta) = L_{\text{eff}} \frac{\eta hcH(\eta) N\sigma_\eta R}{\lambda^2 \kappa} \frac{\mathrm{d}n}{\mathrm{d}T} \left(\frac{2}{\pi w_0^2}\right)^{\eta - 1} \tag{21}$$

and the normalised transmission is given by:

$$T_{\text{CAt}}(z) = 1 + \frac{\vartheta(\eta)}{\eta} \frac{1}{(1 + \frac{z^2}{z_0^2})^{\eta - 1}} \tan^{-1}\left(\frac{2\eta \frac{z}{z_0}}{\left[(2\eta + 1)^2 + \frac{z^2}{z_0^2}\right] \frac{t_c(z)}{2\eta t} + 2\eta + 1 + \frac{z^2}{z_0^2}}\right) \tag{22}$$

Since the thermal characteristic time $t_c = w^2/4D$ is greater than 300 µs for a laser with a repetition rate in the range of MHz, the time dependent part of the equation is negligible when the time for various measurements exceeds 10 ms. Therefore equation 22 can be simplified to:

$$T_{\text{CAt}}(z) = 1 + \frac{\vartheta(\eta)}{\eta} \frac{1}{(1 + \frac{z^2}{z_0^2})^{\eta - 1}} \tan^{-1}\left(\frac{2\eta \frac{z}{z_0}}{2\eta + 1 + \frac{z^2}{z_0^2}}\right) \tag{23}$$

The correlation between the nonlinear refractive index and the thermo-optical coefficient is

$$\frac{\mathrm{d}n}{\mathrm{d}T} = \frac{4n_2\kappa}{\alpha w_0^2}. \tag{24}$$

The CA measurement can be fitted to obtain $\Delta\varphi$ and ϑ using equation 12 and equation 23 respectively. With equation 9, equation 13, equation 18, equation 21, and equation 24 the product of the electron density and the thermal 2PA coefficient $N\sigma_\eta$ can be formulated to:

$$N\sigma_\eta = \sqrt{2\ln(2)} \frac{hc\alpha\vartheta}{\eta P \lambda \Delta\varphi} \left(\frac{2}{\pi w_0^2}\right)^{1 - \eta} \tag{25}$$

The linear absorption coefficient α and the number of photons involved in the absorption η, as well as the power P, the beam waist w_0 and wavelength λ of the laser are measured previous to the z-scan measurement. With the well documented constants for the speed of light c and the Planck's constant h and the values from the two fits, $N\sigma_2$ can be calculated. This can be used to obtain the thermal 2PA coefficient β_{th} with the following equation:

$$\beta_{\text{th}} = 2N\sigma_2 \frac{\lambda}{hc} \tag{26}$$

By comparing β_{th} with the 2PA coefficient measured with an open aperture z-scan one can determine if thermal effects affected the β measured with the open aperture z-scan. If the ratio $\beta_{\text{th}}/\beta > 10$, it confirms that there is only a tiny thermal contribution involved in two-photon absorption.

1.4 Acousto-optic modulator

An acousto-optic modulator (AOM) consists of a piezo-electric transducer bound to a suitable crystal. A radio frequency (RF) signal in the range of 10-10^3 Hz is applied to the transducer and generates an acoustic wave within the crystal. This locally changes the refractive index and the AOM acts as an optical grating [59]. Figure 5 shows the diffraction of two rays of light on two consecutive wavefronts in a crystal. Only a part of the beam is diffracted. The condition for constructive interference of the scattered light is given by [60]

$$n\lambda_L = \Lambda(\sin\theta_i + \sin\theta_d), \tag{27}$$

where n is the refractive index, λ_L is the wavelength of the light, Λ is the acoustic wavelength and θ_i is the angle of the incoming light relative to the acoustic wave front and θ_d is the angle of the diffracted light, scattered on the wave front.

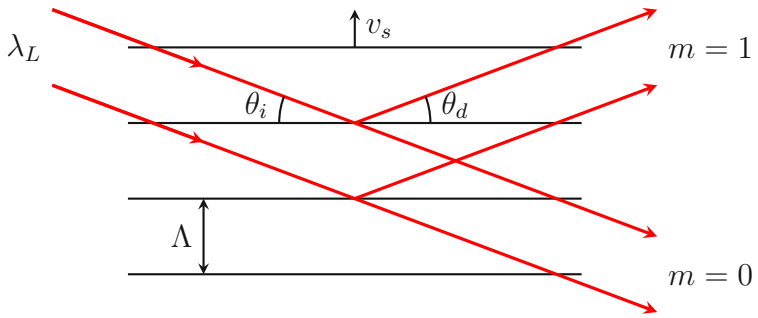

Figure 5: *Illustration of an optical grating. Two rays of light (red) impinging on two consecutive wavefronts in a crystal travelling with the speed of sound v_s (black). The 0^{th}-order is not diffracted, the 1^{st}-order is diffracted under the angle θ_d to the wavefront in the crystal.*

When the acoustic wave has a frequency in the range of MHz the conversation of energy and momentum requires that $\theta_i = \theta_d$ is [61]. Equation 27 can be simplified to the Bragg condition:

$$n\lambda_L = 2\Lambda \sin\theta_d. \tag{28}$$

Often the Bragg angle is calculated outside of the crystal. In that case the external Bragg angle is larger than the internal by a factor of n, where n is the refractive index of the acoustic medium.

A strain transducer that is attached to the AOM crystal is driven by a RF signal. This causes a travelling density wave to form inside the crystal with the frequency Ω of the RF modulation. The wave propagates with the speed of sound v_s with the frequency Ω. This leads to a refractive index, modulated with a wavelength of $\Lambda = 2\pi v_s/\Omega$. The crystal acts like a thick diffraction grating with the rulings travelling with v_s away from the transducer. Equation 28 is only a valid approximation if all phonons of the acoustic wave have the same wave vector and the acoustic wave is describable by a plane wave. This can be achieved when the acoustic diffraction is minimised and with a long strain transducer compared to the acoustic wavelength in direction of the laser beam propagation. AOMs with a efficiency of more than 80 % of the light in a single diffraction order are achievable [62].

The RF signal received by an AOM is controlled by an AOM driver. This driver consists of three components, a voltage controlled oscillator, a voltage variable attenuator and an amplifier. The oscillator provides an RF sine-wave output, with its frequency determined by an applied control voltage. The control voltage and the RF frequency correspond linearly. Some drivers do not support varying control voltages and are fixed to a single RF frequency. The output from the voltage controlled oscillator is attenuated by the voltage variable attenuator. The degree of the attenuation is varied by changing the control voltage applied to the attenuator. The resulting signal is then amplified, such that the RF output is sufficient to drive the AOM. The response of the AOM varies with the frequency and amplitude of the input RF signal. Due to the Bragg condition, equation 28, changes of the frequency alter the angle of the diffraction. This can be used, to keep the first order diffraction on a stable position when the wavelength of a tuneable laser is changed. When the amplitude of the RF is changed the efficiency of the AOM is changed. This can be used to modulate the power of the first order laser beam.

To use the AOM as a chopper or shutter, the 0^{th}-order passes through the AOM and is cut off by a pinhole. The diffracted first order beam passes through the pinhole. The easiest way to achieve a chopper effect is to apply the RF signal in form of a rectangular wave. By alternating the signal between 0 V and the desired voltage one can choose when the first order beam is generated and light reaches the sample, and when only the 0^{th}-order exits the AOM.

In addition to having fine control over the exposure time, by altering the duty and frequency of the RF wave, using an AOM has an additional advantage compared to a conventional chopper. The beam is cut of very fast (less than 300 ns) and the time it takes to cut it off is not dependent on the frequency of the RF signal. It is also cut off omnidirectionally, while a mechanical chopper always cuts it of from one side, takes longer until the entire beam is blocked, or not and those times where only part of the beam is blocked are dependent on the frequency of the chopper.

1.5 Two-photon polymerisation

Two-photon polymerisation (2PP), also called two-photon lithography is a technology that allows to create three dimensional structures of arbitrary form with high spatial resolution and dimensional accuracy [40, 63–67]. 2PP is applied in various fields such as micromechanical and microfluidic devices [68–70] and medical applications [71].

2PP is based on a process called photopolymerization. In this process a resin of photo-sensitive material is exposed to light. Radicals are created in photo initiators by absorbing

the photons emitted by the light source. Those radicals induce chemical monomers to cross-link together to form polymers [72]. This forms an insoluble solid network. The unpolymerised resin can then be removed by solvents after the polymerisation process. Polymerisation can be divided into step-growth and chain-growth mechanisms. Part of the chain growth mechanism is the free radical polymerisation, which is the base of most 2PP applications [67].

The polymerisation process can be divided in three steps: Chain initiation, chain propagation and chain termination. Those 3 steps are shown in figure 6.

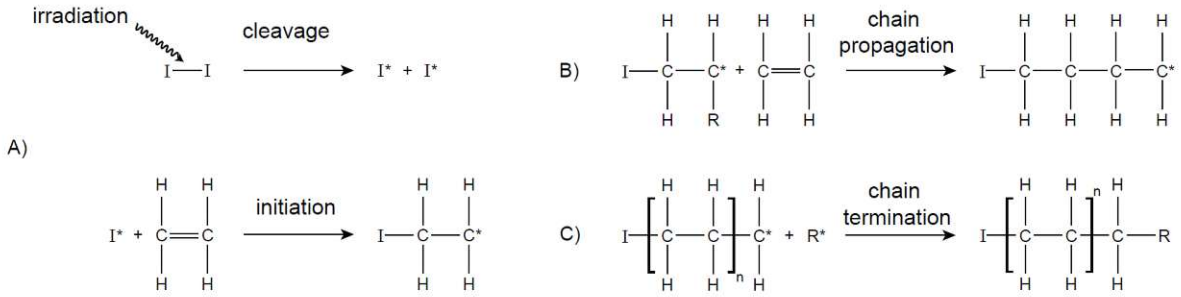

Figure 6: *Steps of reactions during radical polymerisation. Irradiation is absorbed by a photo initiator and it creates a reactive species I*. A) Chain initiation: I* reacts with base untis of the monomer or prepolymer. It breaks the C=C double bond apart and the monomer becomes reactive. B) Chain propagation: additional base units are added and new radicals are created. C) Chain termination: The process in B) is ended by recombination with a radical, chain to chain combination or reactive oxygen species [40, figure 2.3].*

A) Chain initiation When the sample is exposed to light of a suitable intensity and wavelength an initiating molecule, referred to as photo initiator, absorbs it. Upon excitation of the PI, free reactive species are created. In the chain initiation phase these free reactive species break apart the C=C double bonds of the base monomer. The free reactive species bonds with the monomer and the monomer is then itself reactive, since one carbon atom is missing an electron.

B) Chain propagation The molecule with a missing electron then reacts with another base unit and breakes the C=C double bond of it apart and attaches itself to it. The molecule starts to form a chain with the reactive radical always moving to the end of the chain [73].

C) Chain termination This chain propagation continues until a termination reaction inactivates the reactive species. This can happen by reacting with another chain that also has a radical ending or by recombination with another radical [74].

In 2PP 3D printing, the utilised PI needs to be excited by a 2PA process. The absorption process only takes place in the focal volume of the laser, since a threshold of photon dosage is required to excite the PI molecules in a nonlinear absorption process. Outside the focal volume of the utilised laser the photon density is too low for 2PA to occur and the resin will not polymerise. In common stereolithography this would not happen, since they use linear excitation which can occur along the entire beam path. By

moving the focal spot with high spatial and temporal precision arbitrary 3D structures with a high spatial resolution smaller than the diffraction limit of less than 100 nm [35, 75–77] can be created inside a photosensitive resin. In the majority of 2PP setups, the focal point is moved by using galvanometer-based laser scanners, which allow precise positioning of the laser beam in x- and y-direction by two movable mirrors inside a field of view. Structures bigger than a single field of view can be created by stitching multiple field of views together. To manage this the stage, on which the sample has been placed, is moved with linear motors until a new block of the resin is inside the field of view. The stage can move the field of view in x, y and z direction. This allows to fabricate structures in the range of centimetres without compromising the precision of the process [78].

2PP 3D printing is a growing sector and new resins, that are biodegradable, more efficient, or usable in a broader spectrum are constantly developed. When new materials are developed it is important to characterise these synthesised photoinitiators. This is most commonly done with z-scans. Especially spectral z-scans are very important to match the wavelength of utilised lasers with the PI.

1.6 Scope of this thesis

The scope of this thesis was to redesign an existing z-scan setup to optimise the alignment procedures of the setup, allow for closed aperture measurements and minimise thermal effects and testing the viability of an AOM as a chopper in a system with moveable components. Both OA and CA z-scans were performed, see subsubsection 1.3.1 and subsubsection 1.3.2. The following changes in the z-scan system were implemented:

- Installation of additional mirrors and mirror mounts.
- Inclusion of multiple apertures to allow for faster and easier alignment of the system as well as the possibility of closed aperture measurements.
- Installation of an AOM instead of a mechanical chopper.
- Multiple Python programs were written to automatically fit the data both for CA and OA z-scan measurements, as well as compare the two and show the results in graphs and in tables.

It was tested if an acousto-optic modulator is a sensible device in a system with movable mirrors, or if the beam path is too unstable for the usage of it. The AOM was chosen to streamline the measuring process. It also allows for much finer tuning for the amount and frequencies of light that is blocked compared to a mechanical chopper. It was tested if this possibility to fine tune can be used to minimise thermal effects. The installation of an AOM also allows to use the z-scan at a single wavelength with different powers, while the laser is utilised for another experiment simultaneously.

Additionally it allows to perform intensity and time dependent measurements. Intensity dependent measurements can be used to find the 2PA threshold of the sample. One way of performing them is by placing the sample at the focus and using the AOM to only allow a very low power to reaches the sample. By gradually increasing the power until the intensity measured with the second photo diode decreases, the 2PA threshold of the sample is detected.

2 Materials and Methods

2.1 Z-scan setup

The previous z-scan setup is shown in figure 7. The utilised femto second laser (MaiTai DeepSee, Spectra Physics, Santa Clara, United States) has a tuning range from 690–1040 nm. It is utilised for three experimental setups, two of them can be used simultaneously. With the help of a waveplate and a polarising beam splitter the laser is divided between two of the setups. With the waveplate (467-4210, Eksma Optics, Vilnius, Lithuania) the portioning between the p-polarised and s-polarised light can be adjusted by changing its angle. The polarising beam splitter cube (PBS052, Thorlabs, Newton, United States) transmits the p-polarised light to the z-scan and one 2PP printer and reflects the s-polarised component to a second 2PP experiment setup. A mirror on a moveable stage is used to change the beam path between the 2PP printer setup and the z-scan setup. A beam splitter directs a fraction of the beam to the first photo diode (PDA100A-EC, Thorlabs, Newton, United States). Afterwards a dispersive element increases the positive dispersion to compensate the lesser number of optical elements compared to the 2PP setup to which the negative dispersion of the DeepSee module is set. A custom built chopper (142 Hz rotation frequency, 78 µs on-time) reduces the exposure time to minimise thermal effects in the sample. A flip mirror can redirect the beam to an autocorrelator setup. The beam is then expanded in a reflective 4x beam expander (BE04R/M, Thorlabs, Newton, United States). A positive and negative achromatic doublet lens (AC254-200-B, Thorlabs, Newton, United States) with a focal length of 200 mm, with achromatic coating (650–1050 nm) focuses the beam. A motorised stage (LCS16-025-2(4)5, SMAC, Carlsbad, USA) operated by an one-axis controller (LCC-10, SMAC, Carlsbad, US) moves the sample ±12 mm in and out of the focus. A second photo diode of the same model measures the transmitted intensity. The signals of both photo diodes are recorded by an oscilloscope (DS4024, Rigol, Beaverton, USA).

The following additional components were changed or introduced to the setup. To test if being able to finely tune the on-time of the chopper, improves the measurement by further decreasing thermal effects, the custom built mechanical chopper was replaced with an AOM (MCQ110-A1.5-IR, AA Opto-ELectronics, Orsay France). The RF wave is generated by a function generator (AFG21005, RS Components, Corby, Great Britain) and a 110 MHz fixed frequency driver (MODAxx, AA Opto-Electronic, Orsay, France) supplies it to the AOM.

The amplifier is situated close to the function generator, outside of the protective box, that shields the rest of the setup from outside interference and experimenters in the lab from the laser light. This is done to reduce interference due to the heat the amplifier generates. The voltage controlled oscillator however is as close as possible to the AOM to minimise signal loss and noise creation.

For fine tuning of the beam position on the AOM two mirrors had to be introduced to the setup, as well as an aperture to cut off the 0^{th} order of the beam. Since the beam exits the beam expander at an angle, two more mirrors were installed between the expander and mirror 6. Two more apertures were introduced to the setup. One after mirror 6 and the second directly before photo diode 2. Closing the first aperture and reducing the beam width helps with achieving the necessary straight beam path through the lens, sample and to the photo diode. It also allows to perform closed aperture z-scan measurements.

To increase the stability of the beam path the 0.5 inch poles with adjustable height of various optical instruments were replaced with 1 inch poles of a fixed height before and close after the newly installed AOM.

Before any new equipment was bought and installed, the new setup was designed in Autodesk Inventor to create an accurate 3D-model. The current version of this design is shown in the appendix in figure 19.

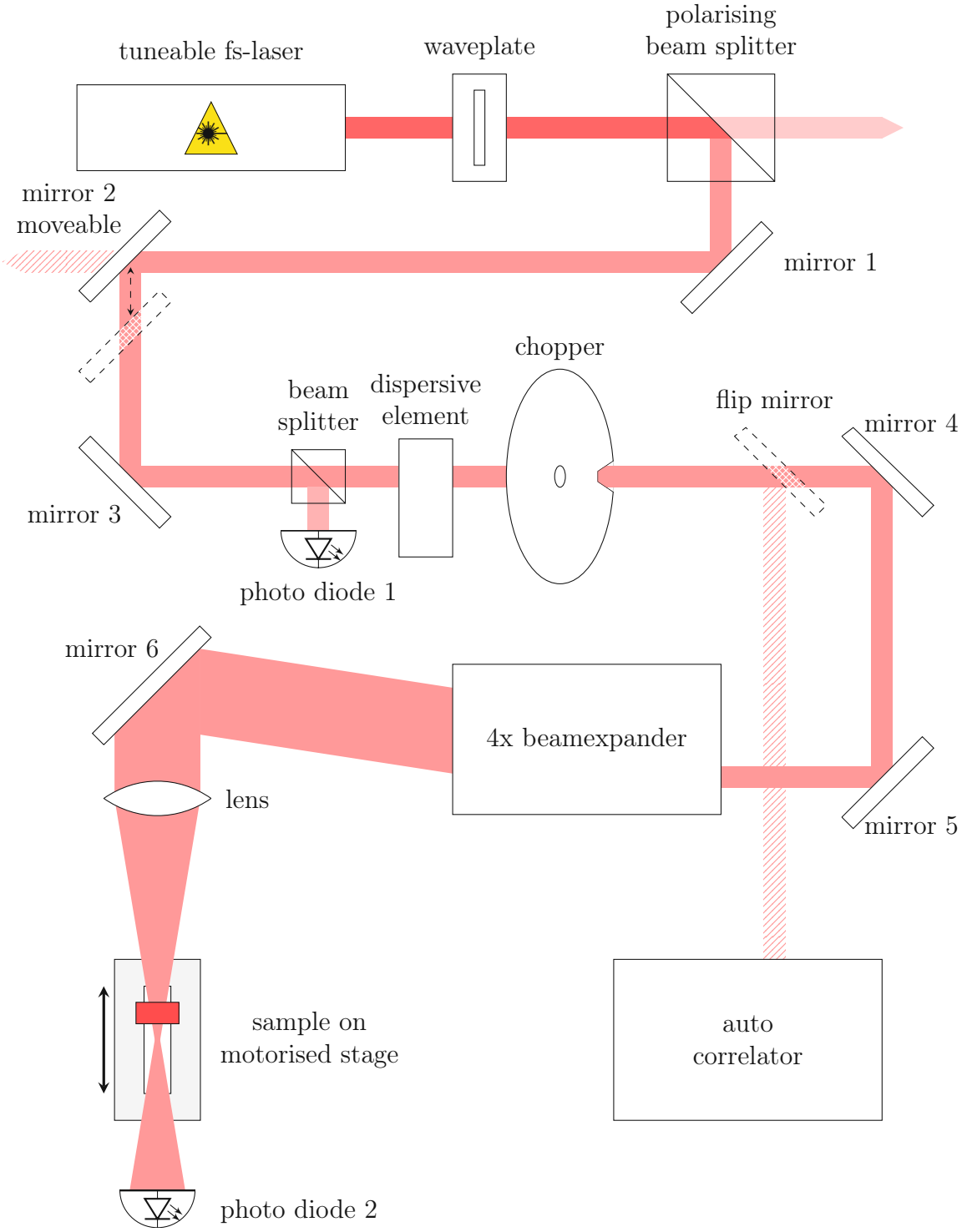

Figure 7: *Schematic of the utilised z-scan setup, before new components were installed.*

2.2 AOM installation

To install the AOM it is necessary that the crystal is positioned in a way to maximise its efficiency. Therefore the AOM is mounted on a stage which is fixed to a height adjustable pole. While it is switched on, it is positioned by hand in a way, that the power of the laser is split approximately 50:50 between the 0^{th} and the 1^{st} order. When this is achieved the AOM is fixed in place and a sensitive photo diode (S121C, Thorlabs, Newton, United States) that is attached to a power meter (PM100D, Thorlabs, Newton, United States) is installed after an aperture. The aperture is positioned on a manual stage to easier align it with the first order beam. Then it is closed until no power from the 0^{th} order reaches the photo diode. This can be checked by turning the AOM off and closing and opening the shutter of the laser. If the aperture is positioned correctly, the photo diode will measure the same power in either case. When only the first order is measured, the two mirrors in-front of the laser are used to fine tune the position of the laser beam. This is done until the measured power reaches a maximum. When the measured power increases rapidly, the AOM should be turned off, to check if the aperture still blocks the zeroth order. If the measured power is in the mW range while the AOM is switched off, the beam shifted enough, that the 0^{th} order is no longer completely blocked by the aperture. If that is the case the 0^{th} order has to be blocked again.

2.3 Z-scan procedure

Before any measurements are performed, it is necessary to check if the alignment of the beam at the AOM is still correct. With a photo diode (S121C, Thorlabs, Newton, United States) attached to a power meter (PM100D, Thorlabs, Newton, United States) the intensity of the beam is measured directly after the AOM. It is then moved behind the aperture and measures only the intensity of the 1^{st} order. If the 1^{st} order has less than 75 % of the total power the alignment of the AOM has to be improved. Those measurements are done with 500 mW or less, to avoid damaging the photo diode. Afterwards a second power meter (Fieldmax II, Coherent Inc, Santa Clara, USA) is installed after the focusing lens and spot checks of the power are performed. If the power set in the user interface is within 3 mW of the power measured by the power meter the z-scan can be performed, if there are greater deviations the input power has to be recalibrated. During the calibration the AOM has to be supplied with a continuous voltage for all power measurements and must not be used as a chopper.

To perform measurements the liquid samples of the photo initiators are filled into a cuvette of 1 mm thickness, a volume of 120 µL, and a 17.5 mm × 6.5 mm aperture (170-000-1-40, Hellma-Analytics, Müllheim, Germany). At the start and end of each measurement the cuvette is thoroughly cleaned with 1-Propanol (≥ 99.9%, Sigma- Aldrich) and lens tissue with ethanol is used for the aperture.
To ensure that there are no residues from previous scans in the cuvette, blank scans are performed before measuring the PI. They are performed with 1-Propanol in the cuvette and the laser tuned to 800 nm at 600 mW, 800 mW, and 1000 mW. All scans are repeated 3 times at each input power and the cuvette is considered clean if all scans result in average noise at each stage position. Before filling the cuvette with the utilised photo initiator it is again cleaned with pressurised air.

The measurements are performed using a custom built Python software that Dr. Wolfgang Steiger programmed as part of his dissertation [40]. Each experiment set is measured

three times each at different power settings. After nine scans 100 µL of fresh sample material was pumped into the cuvette.

The raw data of the OA z-scans is automatically fitted with a custom Python program and the CA data with another program. In both cases the fits and raw data points are plotted and are manually checked for accuracy. A third custom Python program is used to analyse the data and compare the OA and CA data.

2.3.1 System calibration

If there was a long period between z-scan measurements or if the set up was changed in a significant way the parameters of the laser need to be measured to achieve accurate calculations.

Input power The input power is changed by changing the waveplate angle θ using a motorised rotation stage (PRM1Z8, Thorlabs, Newton, United States). The correlation between the laser power and θ is given by [79]

$$P = P_0 + P_{\text{out}} \cdot \cos^2(\theta \cdot a + \phi_0). \tag{29}$$

To do this P is measured by a power meter (Fieldmax II, Coherent Inc, Santa Clara, USA) which is installed in the beam path after the focusing lens. For all utilised wavelengths the waveplates are rotated from 0–120° in 3° increments and a database is created so that it is possible to simply enter the desired power in the user interface and the software sets the waveplate at the correct position.

Auto correlator The Mai Tai eHP Deep See (Spectra Physics, Santa Clara, USA) includes two adjustable prisms, which can compensate the group velocity dispersion and minimise the pulse duration at the sample. Since it is always necessary to have a measurement device that is at least as fast as the measured event, the pulse of an ultra short laser is not easily measured [80]. Instead of an external device, an auto correlator is used which measures the laser pulse width by comparing one laser pulse with another one from the same laser with a slight time delay between them. As shown in figure 7 a flip mirror can redirect the laser beam to the utilised auto correlator.

Figure 8 shows the setup of the utilised auto correlator. The principle is based on the recording of the second order correlation function using a Michelson Interferometer.

The incoming laser pulse with the electric field $E(t)$ is separated into two beams via a beam splitter. By varying one of the beam paths a time delay, τ, between the two fields is generated. The length of the beam path can be varied by having a pair of mirrors on a stage (MF A-CC: B15 0105) that is able to move in one direction (cf. figure 8). The two beams, $E(t)$ and $E(t+\tau)$, are focused with a parabolic mirror and recombined in a nonlinear barium borate crystal generating a second harmonic signal. The total intensity of the second harmonic I_{SH} is proportional to [81]

$$I_{SH}(t+\tau) \propto [E(t) + E(t+\tau)]^2. \tag{30}$$

When the square in equation 30 is unfolded to

$$I_{SH}(t+\tau) \propto E(t)^2 + 2E(t)E(t+\tau) + E(t+\tau)^2, \tag{31}$$

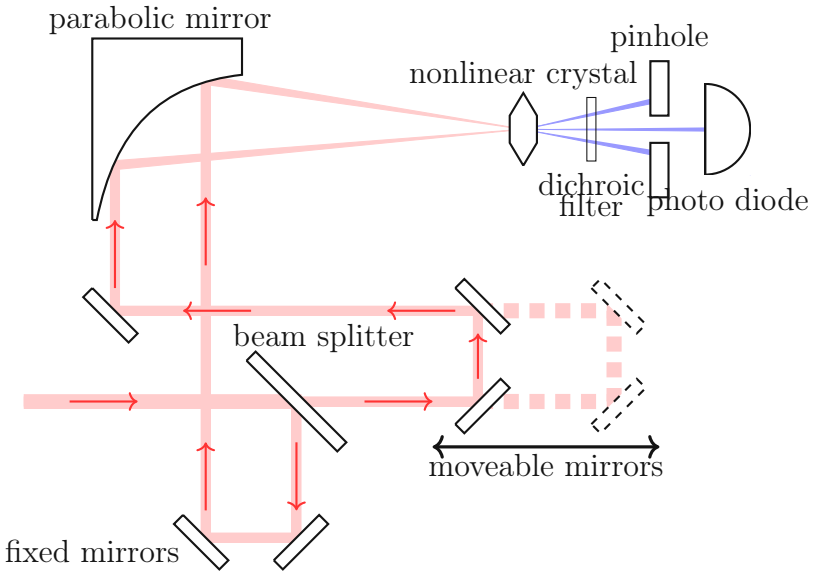

Figure 8: *Schematic of the auto correlator setup*

it illustrates that the I_{SH} is composed of three components. Due to the noncollinear geometry as shown in figure 8 the two components $E(t)^2$ and $E(t+\tau)^2$ can be blocked by a pinhole and only the component $2E(t)E(t+\tau)$ is measured by the installed photo diode (DET100A2, Thorlabs, Newton, United States).
Therefore the measured intensity will change depending on the overlap of the two pulses, as is illustrated in figure 9. A maximum is measured when the two pulses overlap completely. If there is not sufficient overlap of the two pulses their combined intensity is not sufficient to stimulate second harmonics in the nonlinear crystal.

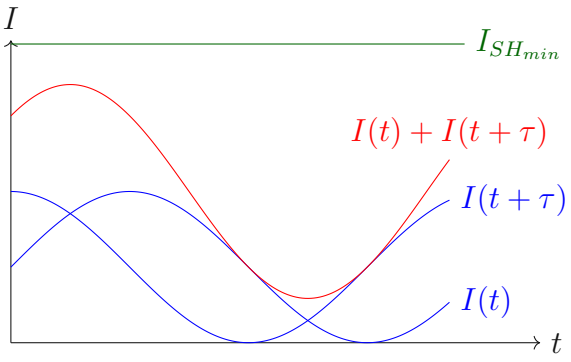

Figure 9: *In blue an illustration of the intensity of the two beam paths, in red their combination and in green the minimum intensity for second harmonics*

The amplitude measured by the photo diode I_{AC} is proportional to

$$I_{AC}(\tau) \propto \int [2E(t)E(t+\tau)]^2 dt. \tag{32}$$

and therefore the intensities of the two pulses

$$I_{AC}(\tau) \propto \int 2I(t)I(t+\tau)dt. \tag{33}$$

The auto correlator doesn't measure the pulse width directly, but if the two beams are identical, equation 33 can be solved analytically and the pulse width can be calculated by dividing the auto correlation signal width by a constant factor that depends on the profile of the pulse [81].

The time delay can be changed by moving the mirrors on the stage in small increments via a stepper motor (SMC100CC, Newport, California, United States).

The measurement is then executed multiple times, changing the motor position of the Deep See module, which changes the position of the two crystals to find the minimum pulse width.

Beam-profiling To measure the characteristic parameters of the focused laser beam a CMOS camera (UI-1492LE, IDS, Obersulm, Germany) is installed on the stage and pictures are taken at multiple stage positions (0–24 mm in 1 mm steps). The camera chip has a resolution of 1.67 µm per pixel. Before mounting the camera in the beam path filters are installed to protect the camera chip. The resulting images were reviewed manually and pictures with artefacts were removed or repeated. A custom made Python software was then used to calculate the characteristic beam parameters by integrating the 2D-Gaussian beam recorded by the camera chip.

By repeating those steps on multiple stage positions it is possible to fit the beam waist $\omega(z)$, extract the Rayleigh length z_R and determine the focal position z_0. The beam quality factor can then also be calculated using $M^2 = z_r/z_{\text{ideal}}$.

For a real laser the beam quality factor M^2 is always greater than 1. A value below 1.2 indicates a sufficiently good beam quality. If the factors were higher the laser beam would deviate too much from a Gaussian beam and the calculations and assumptions utilised to calculate σ_2 from the measured transmission changes would no longer be correct.

As an additional method to calculate w_0 and z_R, photos were taken in front of the beam expander. Those were analysed with a different custom Python software using the same method of integrating the Gaussian beam to calculate the beam waist at the position of the picture. The average beam width from multiple pictures at the same position were used to calculate the beam width and Rayleigh length at the focal area after the beam passed through the beam expander and the focusing lens.

2.4 Chemical structure of the sample

All measurements in this thesis were performed with Rhodamine B (≥95% HPLC, Sigma-Aldrich). Rhodamine B (cf figure 10) is a common dye with well documented σ_2 at different wavelengths [82].

All scans were performed with a 10 mM solution of Rhodamine B in methanol (Uvasol©).

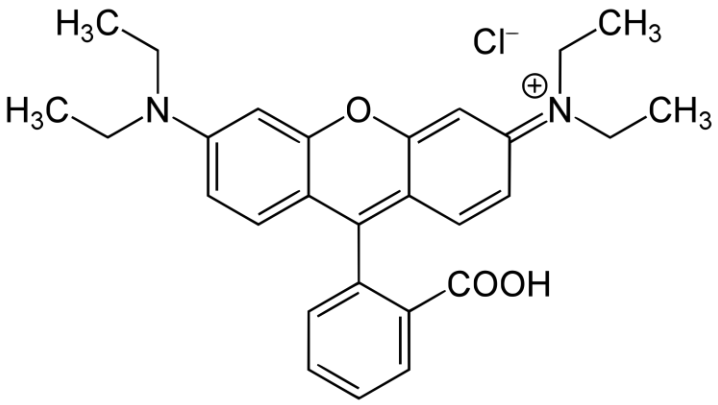

Figure 10: *Chemical Structure of Rhodamine B*

3 Results

3.1 Acoustic optic modulator

An AOM was installed in the setup to replace the mechanical chopper. This was done since an AOM allows for fine tuning of the chopping frequency, allows for power changes of the laser without moving the waveplate and does not need to be removed for power calibrations. It also chops the incoming light much cleaner than a mechanical chopper, since it is orders of magnitude faster to apply or remove the driving voltage to the AOM than it takes for the rotating disk with gaps of a mechanical chopper to completely cut off or open to a laser beam.

3.1.1 AOM position

The utilised AOM (MCQ110-A1.5-IR, AA Opto-Electronic, Orsay, France) was initially installed where the old chopper used to be in the beam path (see figure 7) with two additional mirrors in front of it to align it correctly.

In the first setup redesign the 0^{th}- and 1^{st} order of the beam were too close together, at the planned aperture position to cut off one without reducing power of the other. To completely cut off the 0^{th}-order without cutting into the 1^{st}-order the distance between the AOM and the pinhole needs to exceed 10 cm to allow for a greater distance between the 0^{th}-order and its diffracted beam.

Next the efficiency of the installed AOM was measured. It depends on the angle between the laser beam and the crystal inside the AOM. It was maximised by changing the angle of the laser beam passing through the device. The efficiency $E_{\text{ff}} = I_1/I_{0+1}$ is calculated by measuring the intensity of the first I_1 order after the aperture 1 (see figure 11) and dividing it by the intensity of 0^{th}- and 1^{st}-order I_{0+1} measured directly after the AOM.

The highest achieved efficiency with the AOM at the position of the old chopper was 45 %. According to the technical data sheet of the AOM it should have a possible efficiency of 85 % when operated with a laser beam with a diameter between 0.4–1.2 mm. A measurement in front of the AOM with the camera utilised in calculating the beam parameters measured a beam diameter of more than 1.8 mm.

Since the laser beam diverts over distance the AOM was moved as close to the laser as the setup allowed.

Reducing the distance between the laser and the AOM by about 20 cm increased the highest reached efficiency by more than 15 % to 66 %. To minimise the distance between the AOM and the laser without changing mirror 1 which is correctly positioned for 2PP experiments the moveable mirror (mirror 2) was placed as close as possible to mirror 1 and the AOM right after an additional mirror (mirror 3). Mirrors 2 and 3 are used to align the AOM. The current setup design is shown in figure 11.

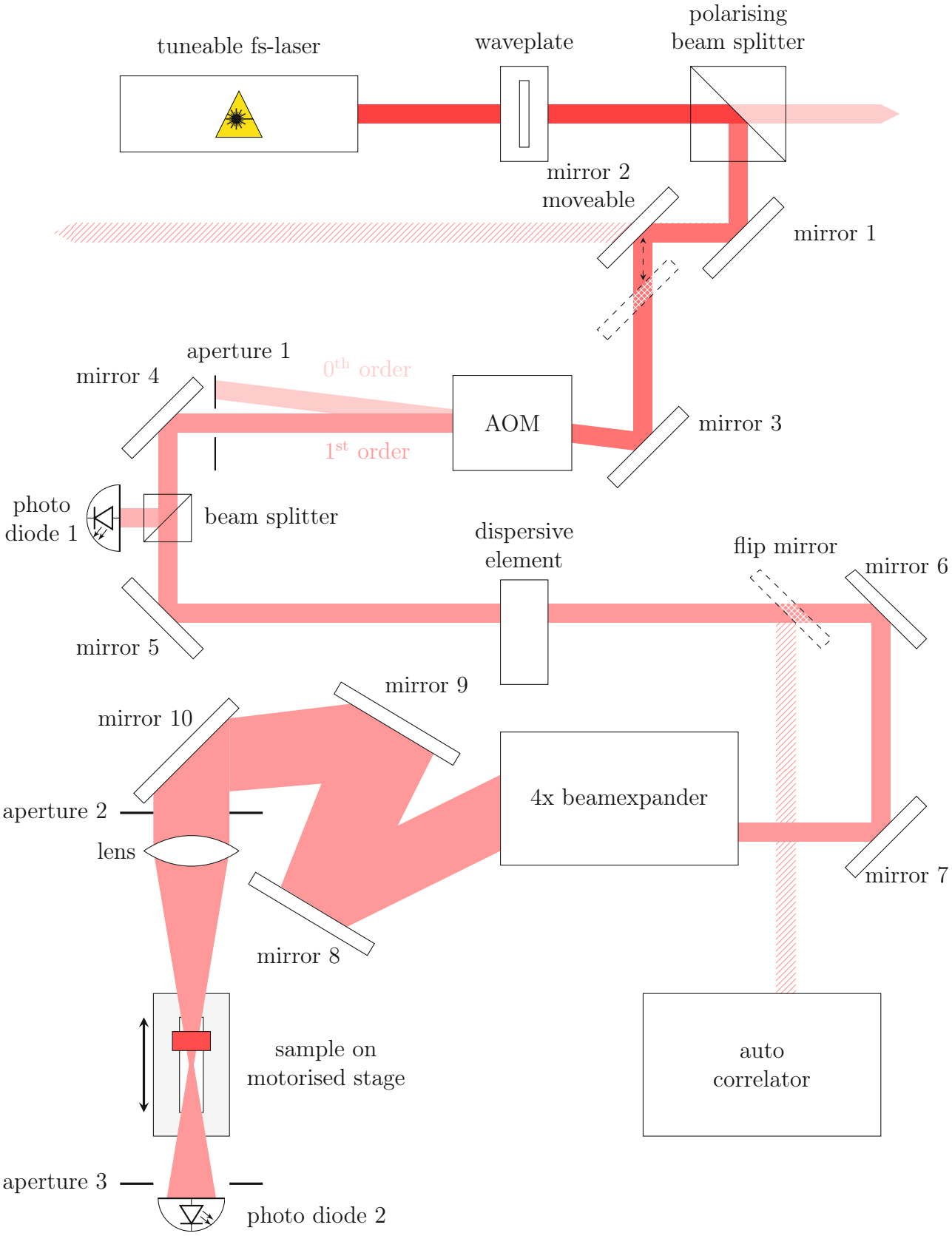

Figure 11: *Schematic of the optimised z-scan setup.*

3.1.2 AOM stability

To measure if movement of mirror 2 changes the alignment of the AOM a series of experiments were performed. Figure 12 shows five experiment sets, were the efficiency of the AOM was measured. In the first four sets the AOM was at the same position and the laser power was increased. Experiment 1 had an input power of 200 mW, experiment 4 had 500 mW with steps of 100 mW in between. At each power five measurements were taken. Between each measurement mirror 2 was moved out of position and back. At 200 mW an average efficiency of 58.9 % was measured. The average for 300, 400 and 500 mW was respectively at 66.6, 67.3 and 66.2 %. Each of those averages lies within the standard deviation of the other three measurements. In Experiment set 5 the distance between the AOM and the laser was reduced to its minimum possible distance without changing the setup for the 2PP experiments that utilises the same laser. It shows an average efficiency of 76.2 % and the deviation of each measurement after the AOM was moved to its current position. The individual values are given in table 1. Those values are still below the maximum possible efficiency according to the manufacturer of the utilised AOM. But since they are high enough to reliably have sufficient power for z-scan measurements no further optimisation steps were implemented.

Table 1: *Efficiency of the AOM over multiple days and the powers of the first order beam as well as the power of zeroth and first order. Both were measured with a sensitive photo diode (S121C, Thorlabs, Newton, United States) that was attached to a power meter (PM100D, Thorlabs, Newton, United States).*

date	1^{st} − order power (mW)	total power (mW)	efficiency (%)
21.06.2022	352	464	75.86
22.06.2022	358	464	77.16
22.06.2022	298	433	68.82
23.06.2022	362	466	77.68
23.06.2022	357	466	76.61
24.06.2022	360	463	77.75
24.06.2022	330	430	76.74
24.06.2022	328	433	75.75
27.06.2022	317	428	74.07
27.06.2022	324	426	76.06
27.06.2022	332	480	77.21
28.06.2022	319	425	75.06
29.06.2022	324	424	76.42
01.07.2022	323	431	74.94
04.07.2022	346	458	75.54
04.07.2022	321	423	75.89
04.07.2022	335	442	75.79
11.07.2022	321	426	75.35

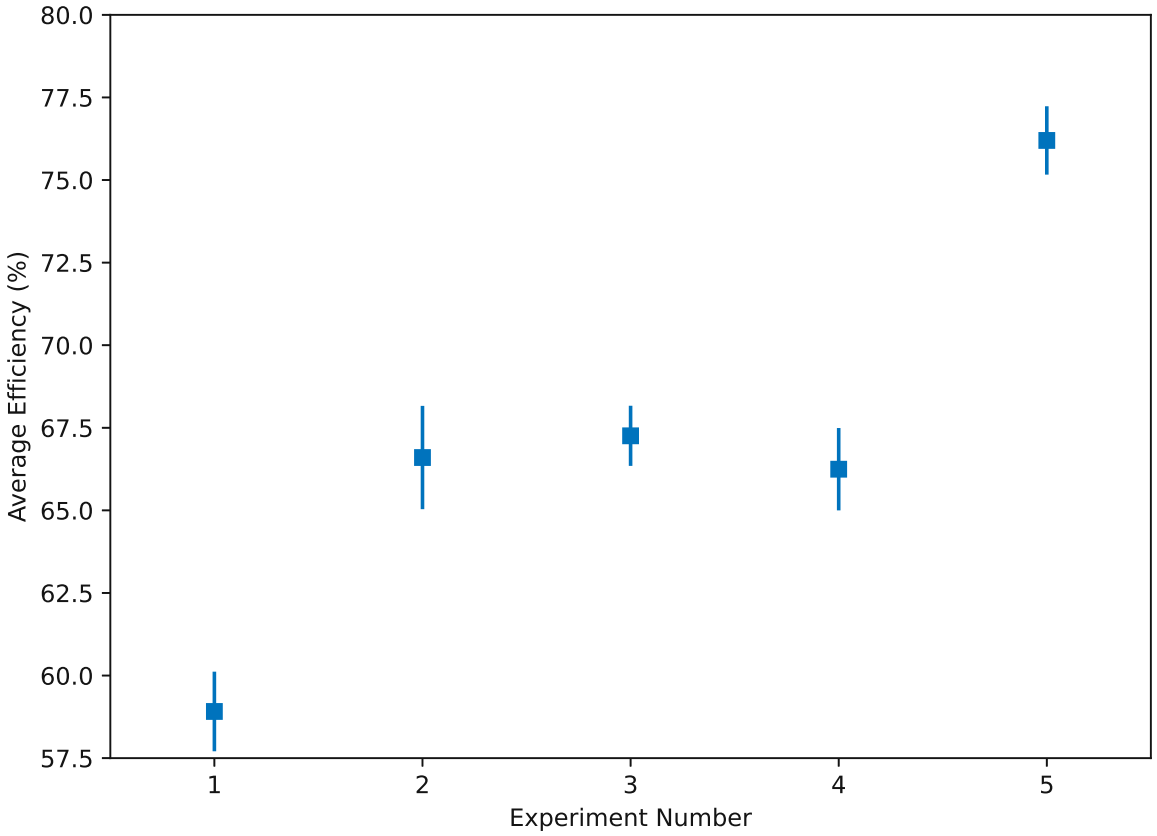

Figure 12: *Efficiency of the Acoustic Optic Modulator of five different experiment sets. In Experiments 1-4 the AOM was at the same position and the power of the laser was increased from* 200 mW *at experiment 1 to* 500 mW *at experiment 4 in* 100 mW *steps. Between the different measurements of each experimental set mirror 2 (see figure 11) was moved out of position and back. Experiment 5 was set at* 500 mW *laser power with a shorter distance between the AOM and the laser compared to sets 1-4. It shows an increase in efficiency when the laser power is higher than* 200 mW *and when the AOM is closer to the laser.*

To check if continuous operation affected the efficiency of the AOM, measurements were taken after eight hours of continuous run time. The first time this was done there was a noticeable drop of 8.33 % between the two measurements, as shown in figure 13 and table 1. On four additional days the efficiency was measured again approximately four hours after the experiment started and again after overall eight hours of continuous on-time. All of those measurements showed slight changes over the day. The highest change was an increase of 3.14 % over eight hours on the 27$^{\text{th}}$ June 2022.

A slight fluctuation is to be expected due to the manual placing of the measuring device. Although mechanical stops were installed to achieve high precision it is not guaranteed that the sensor was always at exactly the same position. Since the sensitivity of the utilised photo diode was non-uniform over its sensitive area it is possible that slight position changes affected the measurement.

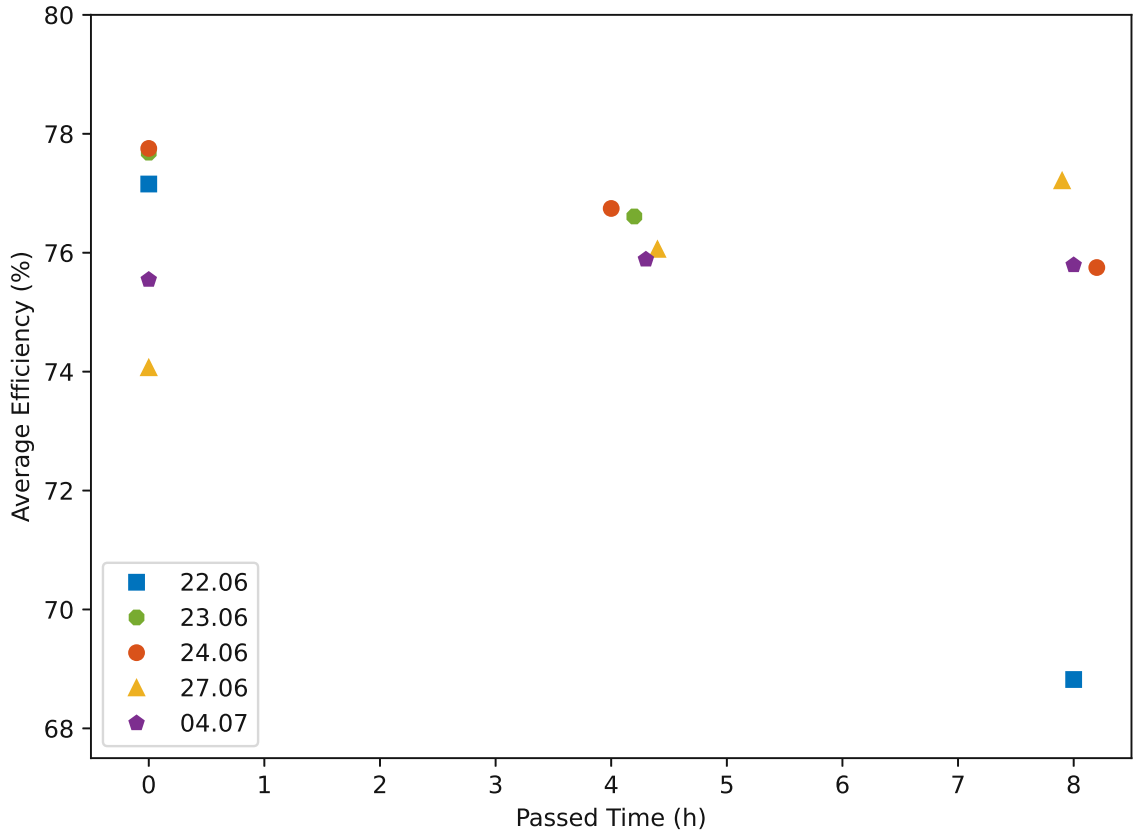

Figure 13: *Efficiency of the Acoustic Optic Modulator measured on five different days. The efficiency was measured again after approximately four hours of z-scan experiment and repeated after an additional four hours of z-scan experiment. On the 22 June 2022, blue square, there was no measurement after four hours and the efficiency decreased by 8 % over the course of eight hours. On the 23rd June 2022, green octagon, there was no measurement after eight hours. Except for the first measurement the efficiency change on one day never exceeded 3 %.*

3.2 Beam parameters

The laser beam was measured multiple times, and the average beam waist of 1.06 mm at the focal point was calculated. The laser shows signs of astigmatism. When a picture of the beam is analysed the extension in x direction is on average 85.76 % of the extension in y direction. The calculated beam quality factor of 1.19 it is still usable for z-scan measurements, with a Gaussian beam assumption.

3.3 Chopper settings

To test if an AOM is a suitable replacement for the chopper and if fine tuning of different opening times has an effect on the measured 2PA coefficient, a series of measurements was conducted. Each measurement was performed with a 10 mM solution of Rhodamine B in methanol. The frequency and duty of the function generator was changed in between each measurement set. The duty of the RF wave is a percentage value, that gives the percentage of the time it sends the high amplitude. Table 2 shows the frequency, duty, chopper open and closed time, as well as how many pulses reach the sample per chopper opening and per second.

Table 2: *Different frequency and duty settings on the function generator utilised in the various experiments in columns 1-3. Column 4 gives the duration that the* 1^{st} *order diffraction is generated as "open time". The 5th column gives the time when no* 1^{st} *or diffraction is generated as "closed time". The last two columns give the number of laser pulses that reach the sample in one "open time" and the number of pulses in one second*

Measurement-set (-)	Frequency (Hz)	Duty (%)	Open time (μs)	Closed time (ms)	Laser pulses per open time (t_O^{-1})	Laser pulses per second ($10^3 \cdot s^{-1}$)
1	137	1	72.99	7.226	5912	810
2	150	1	66.$\dot{6}$	6.$\dot{6}$	5400	810
3	200	1	50	4.95	4050	810
4	274	2	72.99	3.577	5912	1620
5	400	1	25	2.475	2025	810
6	300	1	33.$\dot{3}$	3.$\dot{3}$	2700	810
7	120	1	83.$\dot{3}$	8.25	6750	810
8	62	2.8	451.61	15.677	36 580	2268
9	137	99	7226.3	0.072	585 330	80 190 000

The chopper open-time is the time, where the AOM generates a first order diffraction and the laser beam can reach the sample. The closed time is the time where the AOM does not receive a voltage and no first order is generated, while the 0^{th} order is blocked. The open and closed time are calculated using the following formulas:

$$t_{\text{open}} = \frac{1}{f} \cdot D \tag{34}$$

$$t_{\text{closed}} = \frac{1}{f} \cdot (1 - D) \tag{35}$$

with:

t_{open}	chopper open time,
t_{closed}	chopper closed time,
f	frequency,
D	duty.

The amount of pulses that reach the sample in a second is only influenced by the duty factor. Figure 14 illustrates four different waves as produced by the function generator corresponding to measurement sets 1, 4, 5, and 8.

Each wave alternates between 5 V and 0 V. The former corresponds to the open time in table 2 and the latter to the closed time. The abscissa gives the time in milliseconds. All waves were plotted to start at 5 V at 0 ms except for measurement set 1 (dark blue) which has a delay of 1 ms compared to the other plotted waves. This is done to better compare it with measurement set 4 (yellow). Both have the same opening time and therefore number of laser pulses per open time. But since measurement set 4 has twice the duty factor of measurement set 1, there are twice as many pulses reaching the sample in one second. This can be visualised by comparing the number of yellow peaks with the dark blue ones in figure 14 which are twice as many.

The actual signal received by the AOM is not an instantaneous increase from 0 to 5 V as shown with the perfect rectangles in figure 14. Multiple tests at various frequencies resulted in a rise and drop time of (250 ± 50) ns to reach 5 V. Since the shortest utilised open time was 25 µs this rise time is short enough that the whole measurement can be performed without being modified by the time it takes to open and close this "shutter".

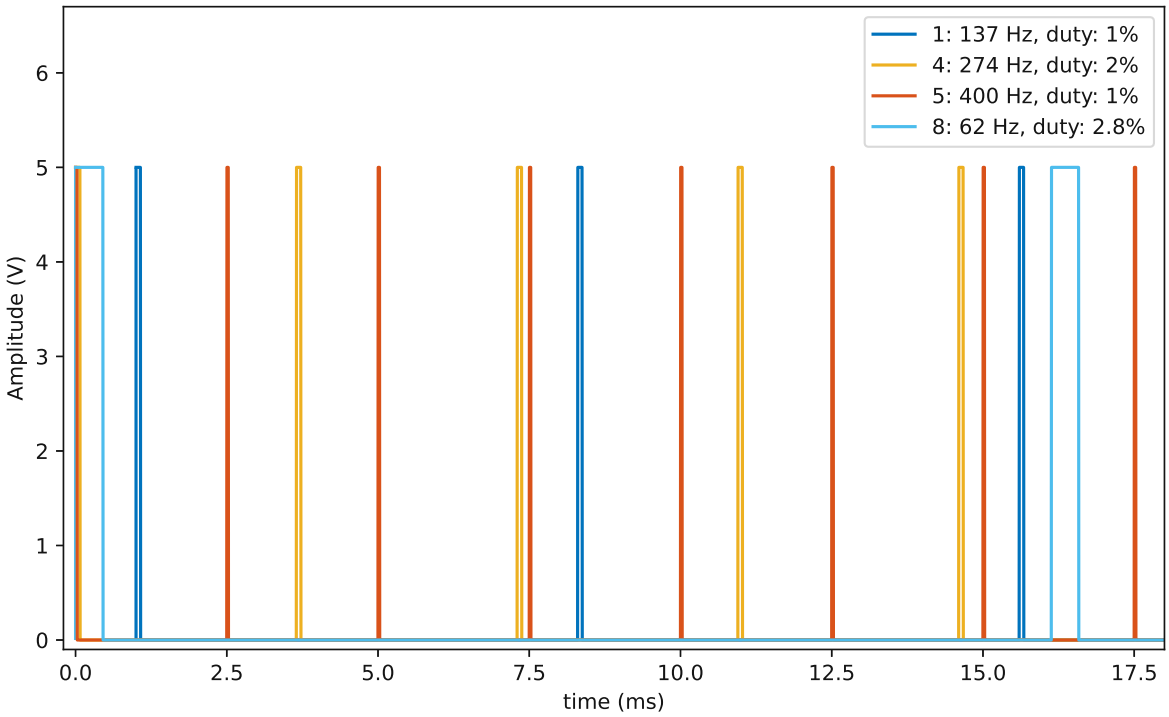

Figure 14: *Multiple square waves that alternate between* 5 V *and* 0 V *like those produced by the function generator in measurement sets 1, 4, 5, and 8. Except for 1 (dark blue) which has a delay of* 1 ms *all waves start at* 0 ms *and* 5 V*. The time period where they are at* 5 V *is the open time in table 2, while the period where the functions are at* 0 V *corresponds to the closed time.*

3.4 Open aperture measurements

For the open aperture measurements the sample was measured three times each at various power settings for each measurement set with different AOM frequencies and duties. In each measurement set the power was changed from 300–1100 mW in steps of 50 mW. The 9^{th} set with a duty cycle of 99 %, which represents measurements without a chopper, were performed with lower powers at 70, 80, 100 and 150 mW.

Figure 15 shows an example of one of the measurements. A custom Python program was utilised to automatically fit all measurements and further analyse the data. The fit for all measurements was plotted and manually checked for accuracy.

The transmission drop that is expected in 2PA is clearly visible at the focal point at 0 mm. The yellow line shows the curve fit with equation 10 which determined q_0 and allowed for the calculation of the 2PA absorption coefficient β and the 2PA cross section σ_2.

In table 3 are the results for the 2PA cross section of the different measurement sets measured with the open aperture z-scan technique. The given σ_2 in the second column is the average 2PA cross section within one measurement set, derived from the average σ_2 at each measured power. If a measurement showed multiple outliers in its data points, it was not included in the calculation of the average σ_2. If it had only one outlier, the outlier was not taken into account when the data was fitted. Each of the average values in the table are calculated from at least 3 measurements. In the third column is the average deviation of the various measurements at the different powers from the non-weighted

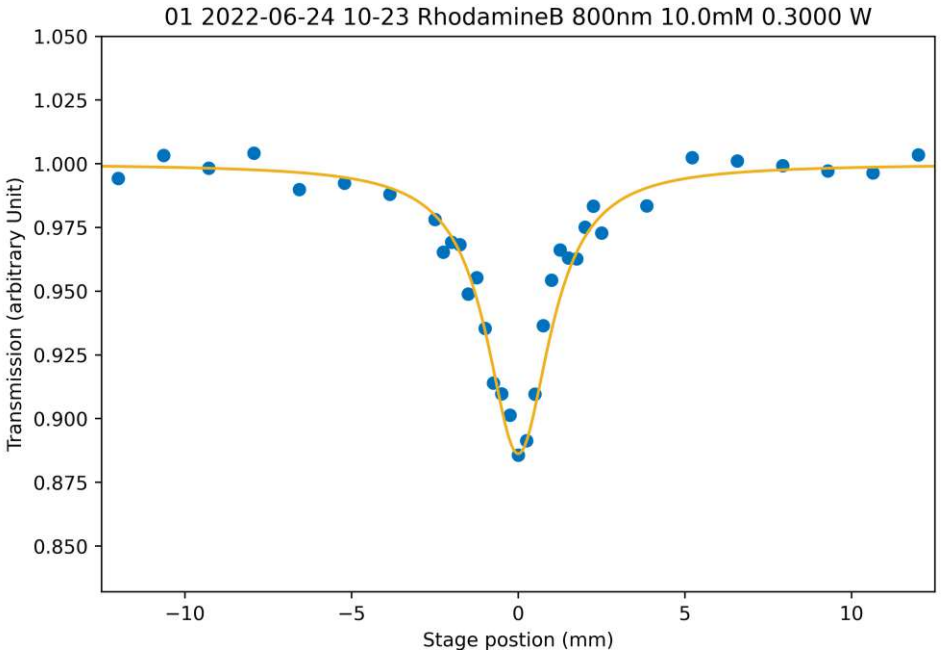

Figure 15: *The normalized transmission of the focus laser beam over the stage position of the sample with* 0 mm *as the focal position. The blue dots are the measurement points for* 10 mM *Rhodamine B in methanol measured on the 24.06.2022 with an* 800 nm *laser with a power of* 300 mW*. The yellow curve is the fit of the data with equation 10.*

average given in column 3. Table 5 in the appendix lists the average σ_2 and its deviation of each individual measurement per power setting that was performed. In that table it is also listed how many measurements were used to calculate the average.

Those measurements are also represented in figure 16, showing the average σ_2 in a logarithmic scale of each power per measurement set. Since the 2PA process is not power dependent, greater difference of different σ_2 values for multiple power measurements of the same measurement set indicate higher thermal effects.

Table 3: *The average two-photon absorption cross section of* 10 mM *Rhodamin B in methanol measured with open aperture z-scan. The first column gives the measurement set, the third and fourth the settings of the function generator that drives the AOM. The second column gives the average of the 2PA cross section of measurements within the same measurement set with different powers. In the third column is the average deviation of the measurements at different powers from the average* σ_2*.*

Measurement-set	average σ_2 (GM)	σ_2 deviation (GM)	Frequency (Hz)	Duty (%)
1	355.76	37.75	137	1.0
2	409.29	40.31	150	1.0
3	122.55	36.48	200	1.0
4	112.8	10.91	274	2.0
5	112.01	34.7	400	1.0
6	127.24	45.15	300	1.0
7	126.66	17.25	120	1.0
8	407.53	49.93	62	2.8
9	2598.65	693.14	137	99.0

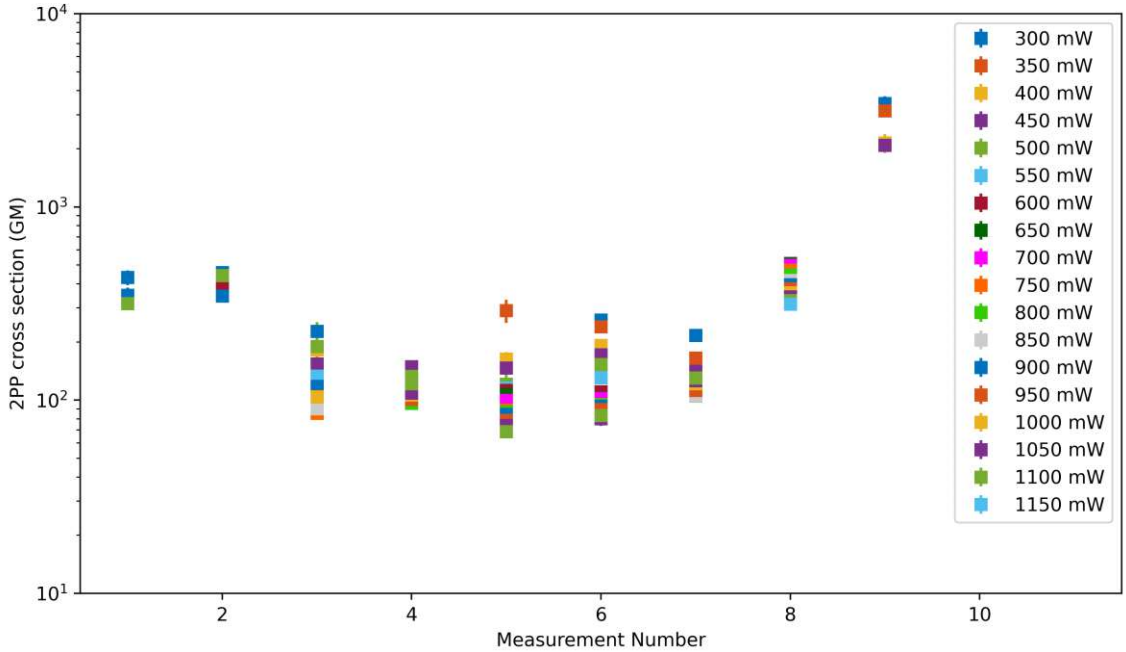

Figure 16: *The average two-photon absorption cross section of* 10 mM *Rhodamin B in methanol in a logarithmic plot for multiple measurement sets. Each measurement set from 1-9 uses different frequencies and duties on an AOM which functions as a chopper. The rectangles in various colours are the average* σ_2 *of one power setting. The farther each* σ_2 *is from the other measurements within the same experiment group, the greater the thermal effects.*

3.5 Closed aperture measurement

The same 10 mM Rhodamine B solution was also measured with the closed aperture z-scan technique to gain information about thermal effects. The same 9 different AOM settings as in the OA measurements were used. At each measurement set multiple measurements were taken, all at the same power of 600 mW for sets 1-8 and at 100 mW for measurement set 9 (see table 2). Aperture 3 (see figure 11) was closed until photo diode 2 only measured 20 % of the OA intensity.

The resulting data was then fitted twice to obtain the nonlinear refraction index n_2 with equation 12 and equation 13 and the thermal 2PA coefficient β_{th} using equation 23, equation 25 and equation 26. Those fits were performed automatically with a custom Python program and an additional custom Python program was used to analyse the data and compare it with OA results. The fit for all measurements was plotted and manually checked for accuracy.

Table 4: *The average two-photon absorption coefficient of* 10 mM *Rhodamine B in methanol. The first column gives the measurement-set, which differentiate by their AOM settings that are given in the last two columns. The* 2^{nd} *column gives the 2PA coefficient measured with the OA scan at the same power as the CA scan was performed. The* 3^{rd} *and* 5^{th} *column give the thermal 2PA coefficient and the nonlinear refractive index averaged over at least three measurements. Both are calculated from CA z-scan data. The* 4^{th} *column gives the ratio between the OA and CA 2PA coefficient. All measurements were performed at* 800 nm *with* 600 mW *except for measurement-set 9 which was measured at* 100 mW*,*

Measurement-set	average β (10^{-9}cm/W)	average β_{th} (10^{-9}cm/W)	ratio β_{th}/β	n_2 (10^{-14}cm^2/W)	Frequency (Hz)	Duty (%)
1	0.681	6.97	10.25	2.13	137	1.0
2	0.632	8.352	13.254	2.21	150	1.0
3	0.272	5.577	20.498	2.08	200	1.0
4	0.291	6.897	23.665	2.39	274	2.0
5	0.273	6.203	22.754	1.85	400	1.0
6	0.268	7.510	28.037	2.18	300	1.0
7	0.307	6.201	20.195	2.15	120	1.0
8	0.912	6.23	6.846	2.30	62	2.8
9	5.21	23.567	4.531	23.3	137	99.0

Figure 17 shows an example of a fit for the n_2 calculation and figure 18 shows the same data fitted for the thermal 2PA coefficient β_{th}. Table 4 gives the averaged results for the different measurement sets. All those averages are at a single power, 600 mW for measurement sets 1-8 and 100 mW for measurement set 9. The second column gives the 2PA coefficient β measured with the OA scan at the according power, averaged over at least 3 measurements. The second gives the averaged thermal 2PA coefficient β_{th} calculated with data from the CA measurements. The ratio β/β_{th} is an indicator for the influences of thermal effects in the sample. The more this ratio approaches 1, the more significant are thermal effects on the measurement. If the ratio is above 10 thermal effects are small enough to be negligible. The 5^{th} column gives the nonlinear refractive index, calculated with CA data and averaged over at least three measurements. The last 2 columns are the frequency and duty of the AOM. It shows that there are small changes of β_{th} with different frequencies, except for the expected big difference between measurement

set 9 and the rest. The changes of the ratio β/β_{th} show that the thermal effects are not only dependent on the duty but also on the utilised frequency. This is discussed in greater detail in subsubsection 4.2.1

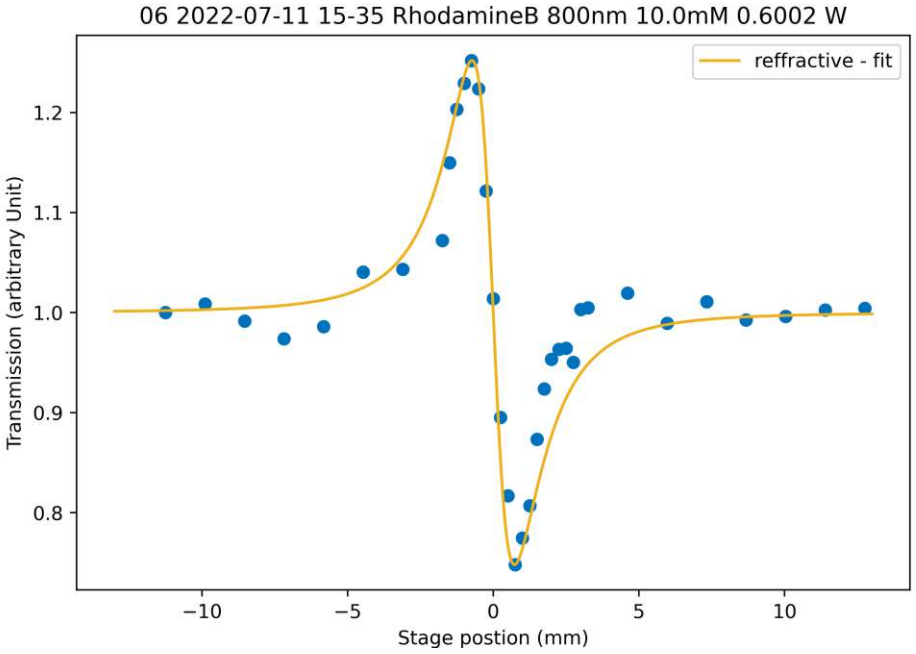

Figure 17: *The normalised transmission of the focused laser beam over the stage position of the sample with* 0 mm *being the focal position. The blue dots are the measurement points for* 10 mM *Rhodamine B in methanol measured on the 11.07.2022 with an* 800 nm *laser with a power of* 300 mW*. The yellow curve is the fit of equation 12.*

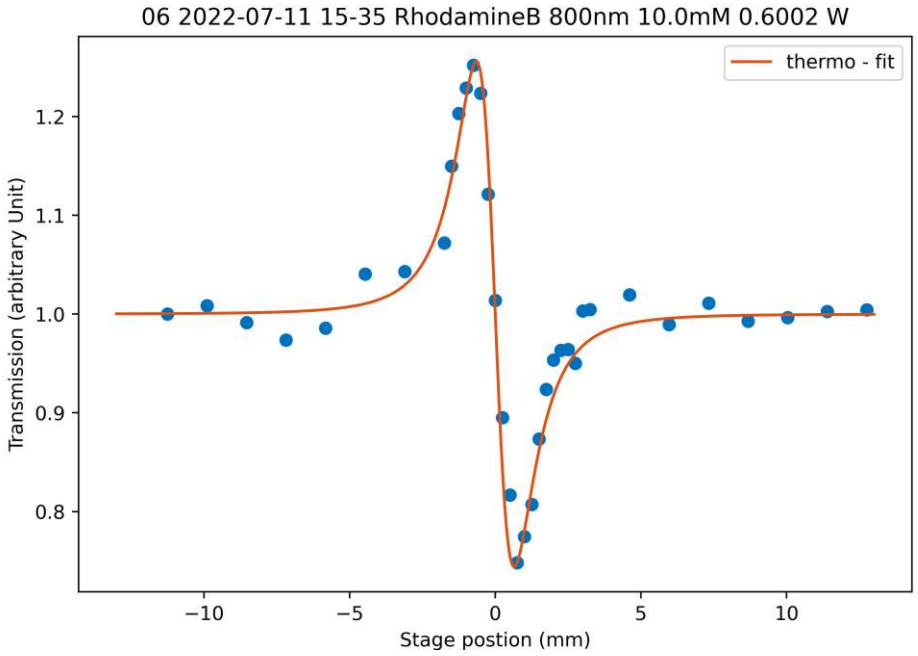

Figure 18: *The normalized transmission of the focus laser beam over the stage position of the sample with* 0 mm *as the focal positon. The blue dots are the measurement points for* 10 mM *Rhodamine B in methanol measured on the 11.07.2022 with an* 800 nm *laser with a power of* 300 mW. *The red curve is the fit of equation 23.*

4 Discussion

4.1 AOM efficiency and stability

The first measurements of the AOM efficiency have shown, that the beam of the utilised laser is too wide for the installed AOM. This could be changed by installing two additional lenses as a telescope. Since this would require additional components and space in the setup it was tested if it could be avoided by moving the AOM as close as possible to the laser. The results showed that, most likely due to beam propagation, the efficiency is still below the efficiency stated in the technical data sheet, but high enough for z-scan applications.

Since AOMs are very sensitive to changes of the beam path, it was important to test, if movement of mirror 2 would necessitate a realignment of the AOM after a certain period of time. The results, as shown in table 1, show, that the efficiency does change by up to 1.2 % between measurements. This is an acceptable margin of error. The measurements also showed, that it is very important to check the efficiency before starting an experiment set, because in one measurement the efficiency was almost 10 % worse than the average efficiency of the other 17 measurements. This outlier was probably due to human error while fixing the moveable mirror in place.

Figure 12 shows that even though AOMs should not be power dependent, there is a noticeable change in efficiency between measurements at 200 mV and measurements with higher voltage. This change of efficiency could be due to the AOM acting as a polarisation filter and the low fraction of polarised light that is cut off has a greater impact at very low powers. In z-scan applications this power dependency is negligible, since most measurements using this system are performed with powers above 300 mV

4.2 AOM as a chopper

The above results clearly show that an AOM can be used as a chopper. Using an AOM instead of a regular chopper offers the following advantages.

The first advantage is easier alignment of the system. With the AOM installed no parts of the z-scan setup have to be removed before a measurement. The chopper that was installed previous to the AOM had to be removed before an experiment started to check if the power of the laser was still correctly calibrated and afterwards it had to be reinstalled by hand. This could lead to small changes in the system, when the chopper was not placed at exactly the correct position. With an AOM the laser power can be measured without a chopper effect, by supplying a RF wave that supplies 5 V at all times.

The second advantage of the AOM is, one can finely tune the amount of time when the first harmonic is generated and pulses can reach the sample. This is hardly possible with an ordinary chopper and almost impossible with a chopper that has to be removed and reinstalled by hand before a new measurement set.

The third advantage is how quickly the laser is cut off with an AOM. With some redesign of the z-scan software the AOM can be implemented in the standardised measurement program. This allows the use of the AOM as a chopper and a fast working shutter simultaneously. By linking the RF wave with the measurement time slot of the photo diode, the sample can be blocked from the laser light when no measurements are performed. This would lead to a decrease in heating effects, since the sample would not be exposed to the laser light while it moves between the various measurement steps and only for the brief period when the photo diode measures the intensity.

The fourth advantage is that the AOM allows for new measuring methods. Due to the fine control of the exposure time, as well as the possibility to change the power that reaches the sample it is possible to perform intensity dependent measurements, measure the 2PA threshold as well as look how exposure time variations affect a sample.

4.2.1 Chopper frequency

In this thesis it was tested if changes in the frequencies of the AOM have an effect on the thermal effects, when the same number of pulses reached the sample within a given time span.
Figure 16 as well as table 4 show, that frequency changes do affect thermal effects inside a sample. Although as long as the time in which pulses reach the sample is smaller than approximately 100 µs the changes are not drastic in Rhodamine B.

The comparison between measurement set 1 and 4 is interesting since it did not show the expected results, that thermal effects would decrease with higher frequency and increase with higher duty [18]. Measurement set 1 uses a RF wave with a frequency of 137 Hz and 1 % duty while the RF wave in set 4 has a frequency of 274 Hz and 2 % duty. In both measurements ≈5900 continuous pulses reached the sample when the AOM was switched on, but in set 4 the AOM was opened twice as often as in set 1. Therefore the same energy reached the sample with each opening time, but the time between chopper openings was different and twice as many pulses reached the sample in set 4. This leads to ≈810 000 pulses reaching the sample in one second in set 1, and ≈1 620 000 in measurement set 4. Contrary to the expected results, the thermal effects in measurement set 4 were smaller than in set 1. This can either be due to chemical effects within Rhodamine B that needs a certain energy threshold, or due to external influences or changes between

the measurements. Finding proof for chemical reactions at a certain threshold is beyond the scope of this thesis, and no such reactions were found in the consulted literature. Changes of external factors like temperature and exact sample placement, between the two measurement sets are plausible, since the open aperture measurement sets 1 and 2 were measured on a different day than measurement sets 3-9. Since the experiment is very sensitive to small changes of the optical system, it could be possible, that temperature fluctuation between the measurements changed the outcome. This would be supported by the similarity of the values for measurement sets 1 and 2, and 3–6, see table 4.

If that is the case, it shows, that higher frequencies slightly improve the negation of thermal effects, while a higher duty slightly increases the measured thermal effects. The experiments with Rhodamine B showed high thermal effects, when the open time of the shutter exceeded 400 µs. As long as the on time of the AOM is below a certain threshold, these effects are much smaller, than the variation of the measurements on different days. This shows, that if possible, z-scan measurements of samples that should be compared to each other, should be measured without a break in between measurements. These variations could be due to temperature fluctuations in the lab, which could not be completely eliminated in the summer, or slight movement of the cuvette inside the sample holder.

4.3 Conclusion and outlook

Even considering only a slight reduction on the thermal effects by using an AOM instead of a mechanical chopper, introducing it into a z-scan system is advantageous.

- It allows for an easier and therefore faster alignment of the experiment.
- It can be utilised as a shutter.
- It allows to change the power that reaches the sample without changing the wave-plate. This is especially useful in the utilised lab, since it allows for scans at a single wavelength while the laser is also utilised for a 2PP experiment.
- The fine tuning of the frequency allows to adjust the opening time of the chopper when a sample is more sensitive to thermal effects and it gives an additional parameter that can be changed when examining a sample.

The possibility to change the power while the measurement is running also allows for a different way of performing z-scans. Instead of moving the sample on the stage it is positioned in the focal point of the laser. By changing the power of the laser and measuring the intensity after the sample the 2PA threshold can be measured.

Installing an AOM as a chopper gives new possibilities to perform experiments and streamlines the alignment of the setup. The single wavelength AOM should be replaced by an AOM that can be used at multiple wavelengths to allow spectral measurements.

A Appendix - Tables

Table 5 : *The average two photon absorption cross section of* 10 mM *Rhodamine B in Methanol measured with open aperture z-scan. Each column corresponds to one measurement set, which differs in the open and closed time and frequency of an AOM that functions as a chopper, see table 2. The first 3 rows give the average sigma of all measured powers for each measurement set, the second the average deviation of the individual results from the above average σ_2 and the third line the number of measurements that were used to calculate the average values above. Afterwards the table has the average values for each power setting individually, if there were less than 3 measurements without outliers there is a '-' in the table.*

Measurement set	1	2	3	4	5	6	7	8	9
average sigma all Powers [GM]	360.2	377.84	124.84	112.8	112.01	123.21	125.05	407.62	2692.82
sigma deviation all Powers [GM]	46.9	65.54	36.53	12.6	35.88	41.56	17.1	51.35	684.69
Number of measurements all Powers [GM]	17	12	54	42	48	49	49	53	17
average sigma at 70 mW [GM]	-	-	-	-	-	-	-	-	3434.84
sigma deviation at 70 mW [GM]	-	-	-	-	-	-	-	-	303.84
Number of measurements at 70 mW [-]	-	-	-	-	-	-	-	-	6
average sigma at 80 mW [GM]	-	-	-	-	-	-	-	-	3148.74
sigma deviation at 80 mW [GM]	-	-	-	-	-	-	-	-	222.41
Number of measurements at 80 mW [-]	-	-	-	-	-	-	-	-	3
average sigma at 100 mW [GM]	-	-	-	-	-	-	-	-	2144.79
sigma deviation at 100 mW [GM]	-	-	-	-	-	-	-	-	242.85
Number of measurements at 100 mW [-]	-	-	-	-	-	-	-	-	5
average sigma at 150 mW [GM]	-	-	-	-	-	-	-	-	1666.24
sigma deviation at 150 mW [GM]	-	-	-	-	-	-	-	-	139.90
Number of measurements at 150 mW [-]	-	-	-	-	-	-	-	-	3
average sigma at 300 mW [GM]	431.25	458.1	227.46	-	-	-	216.35	-	-

Continued on next page

Table 5 : *Average 2PA cross section of different powers. Continued*

Measurement set	1	2	3	4	5	6	7	8	9
sigma deviation at 300 mW [GM]	38.21	17.27	16.47	-	-	-	3.34	-	-
Number of measurements at 300 mW [-]	5	4	6	-	-	-	3	-	-
average sigma at 350 mW [GM]	-	-	-	-	291.01	239.58	-	351.7	-
sigma deviation at 350 mW [GM]	-	-	-	-	40.1	5.3	-	23.25	-
Number of measurements at 350 mW [-]	-	-	-	-	3	3	-	3	-
average sigma at 400 mW [GM]	-	-	181.94	-	163.01	-	-	377.86	-
sigma deviation at 400 mW [GM]	-	-	9.91	-	14.08	-	-	13.49	-
Number of measurements at 400 mW [-]	-	-	3	-	3	-	-	-	
average sigma at 450 mW [GM]	-	-	153.89	149.24	146.57	171.52	141.28	376.3	-
sigma deviation at 450 mW [GM]	-	-	7.21	7.85	8.86	26.81	11.67	10.04	-
Number of measurements at 450 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 500 mW [GM]	-	-	189.81	132.8	121.05	153.13	117.25	436.07	-
sigma deviation at 500 mW [GM]	-	-	63.27	3.93	16.43	6.64	10.44	28.45	-
Number of measurements at 500 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 550 mW [GM]	-	-	133.27	108.65	115.64	130.79	119.69	449.97	-
sigma deviation at 550 mW [GM]	-	-	9.45	4.6	6.88	3.07	9.26	10.17	-
Number of measurements at 550 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 600 mW [GM]	326.55	392.31	112.17	120.16	112.41	110.44	126.53	480.73	-
sigma deviation at 600 mW [GM]	11.04	18.19	6.25	12.28	4.17	5.64	7.53	25.59	-
Number of measurements at 600 mW [-]	4	4	3	3	3	3	3	3	-

Continued on next page

Table 5 : *Average 2PA cross section of different powers. Continued*

Measurement set	1	2	3	4	5	6	7	8	9
average sigma at 650 mW [GM]	-	-	107.63	116.08	107.11	102.86	115.61	509.73	-
sigma deviation at 650 mW [GM]	-	-	2.6	5.58	1.9	1.0	1.39	14.65	-
Number of measurements at 650 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 700 mW [GM]	-	-	91.63	113.19	99.92	102.18	111.5	499.23	-
sigma deviation at 700 mW [GM]	-	-	4.44	5.53	3.64	3.31	13.47	1.57	-
Number of measurements at 700 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 750 mW [GM]	-	-	85.66	98.15	88.98	95.01	106.01	471.66	-
sigma deviation at 750 mW [GM]	-	-	6.5	8.69	6.23	3.55	7.13	6.74	-
Number of measurements at 750 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 800 mW [GM]	-	-	90.95	96.4	86.89	94.91	105.03	444.98	-
sigma deviation at 800 mW [GM]	-	-	10.98	5.41	1.32	2.5	6.19	6.51	-
Number of measurements at 800 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 850 mW [GM]	-	-	90.27	104.33	84.4	83.5	105.53	417.3	-
sigma deviation at 850 mW [GM]	-	-	4.19	5.7	3.43	3.42	2.85	0.25	-
Number of measurements at 850 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 900 mW [GM]	349.0	-	118.12	102.44	84.54	93.09	115.41	396.02	-
sigma deviation at 900 mW [GM]	32.87	-	23.62	3.21	5.07	4.41	6.47	0.23	-
Number of measurements at 900 mW [-]	4	-	5	3	3	3	4	3	-
average sigma at 950 mW [GM]	-	-	104.45	101.03	78.67	89.35	112.27	376.92	-
sigma deviation at 950 mW [GM]	-	-	8.15	3.54	4.01	5.64	2.29	0.15	-

Continued on next page

Table 5 : *Average 2PA cross section of different powers. Continued*

Measurement set	1	2	3	4	5	6	7	8	9
Number of measurements at 950 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 1000 mW [GM]	-	-	104.17	106.36	69.03	81.23	121.87	359.2	-
sigma deviation at 1000 mW [GM]	-	-	6.96	4.77	0.64	5.17	4.81	0.2	-
Number of measurements at 1.000 mW [-]	-	-	3	3	3	3	3	3	-
average sigma at 1050 mW [GM]	-	-	-	108.66	74.12	80.08	126.12	342.67	-
sigma deviation at 1050 mW [GM]	-	-	-	10.83	2.71	6.49	2.59	0.26	-
Number of measurements at 1050 mW [-]	-	-	-	3	3	3	3	3	-
average sigma at 1.100 mW [GM]	316.23	-	46.83	121.67	68.83	83.25	130.24	327.97	-
sigma deviation at 1.100 mW [GM]	8.27	-	23.41	3.18	3.71	2.81	4.71	0.01	-
Number of measurements at 1100 mW [-]	4	-	4	3	3	3	3	3	-
average sigma at 1150 mW [GM]	-	-	-	-	-	-	-	314.36	-
sigma deviation at 1150 mW [GM]	-	-	-	-	-	-	-	0.03	-
Number of measurements at 1150 mW [-]	-	-	-	-	-	-	-	3	-

B Appendix - Figures

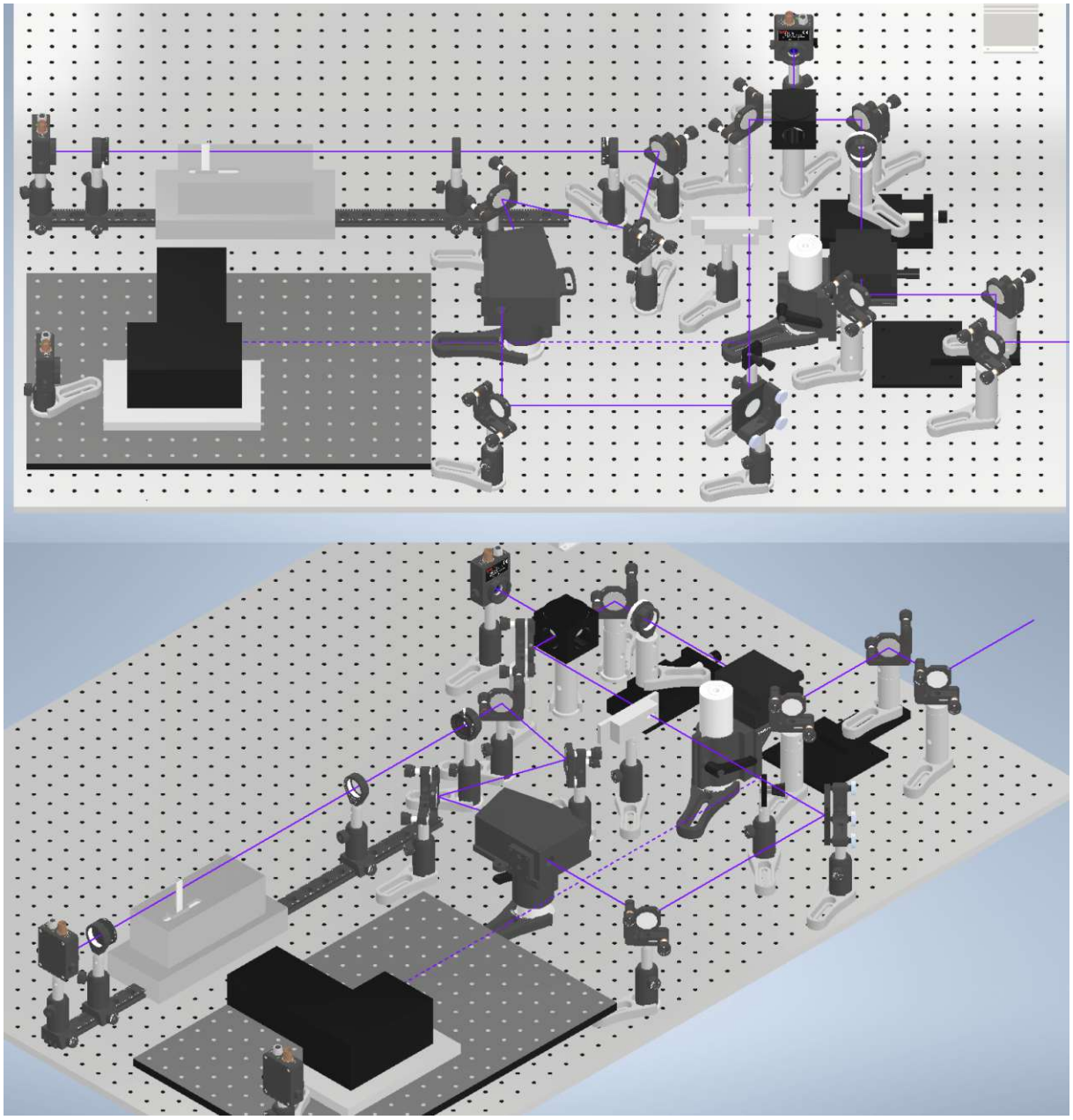

Figure 19: *A 3D model of the utilised z-scan setup from two angels. The blue path simulates the laser path. The laser, the waveplate and the polarising beam splitter is not shown in the model. The laser enters the set-up from the right side of the model and hits the first mirror. The second mirror is on a rail system and can be moved out of the beam path. The black box after the* 3^{rd} *mirror is the AOM. Next in the beampath is a pinhole on a manual stage. the next mirror directs the path through a beamsplitter. One part is directed to photodiode 1, the second continuous straight on, before mirror 5 directs hte beam path to the dispersive element. Mirror 6 and 7 are used to finely direct the beampath into the beam expander. The next three mirrors are used to align the beam in a straight line through, in that order, pinhole 2, the focusing lens, the sample on a motorised stage, pinhole 3 and photo diode 2. The black box on the lower left is the auto correlater. The beampath can be directed to it with a flip mirror which is positioned between the dispersive element and mirror 6.*

C Appendix - Figures

C.1 Analyse_OA.py

```
'''
Textfiles von Open Aperture Measurement einlesen, outliers rausschmeissen,
Fit erstellen/anpassen, Grafik erstellen, Daten Speichern
'''
from fileinput import filename
import numpy as np #numerische methoden
import matplotlib.pyplot as plt #plotlibrary
from matplotlib.ticker import FormatStrFormatter
import os   #pfade
import csv
from glob import glob #find specific stuff in directorys (can probably do more)
from scipy.optimize import curve_fit
import scipy.constants as constant
import time
import Farben

def find_index(array, value):
    return array[array <= value].size

def cm2inch(cm):
    inch = 2.54
    return int(cm/inch)

fig_width = cm2inch(20)             # cm
fig_high = cm2inch(13)              # cm
my_dpi = 100                        # cm

path = r"D:\Documents\Uni\MasterArbeit\RhodamineB_cleaned"  #url to order with textfiles

divide_data = False                 #Divide data from diode2/diode1
manyplot = False

#Bekannte Variablen:
w_0 = 28.33064e-06                  #beam Waist
z_0 = 0.001060083796                #Raileygh Lenght
z = np.linspace(-13, 13, 910)/1000    #z array for fitting
P = 0                               #Power ... Varieiert
R = 81.0e06                         #Repetition Rate
tau = 72.0e-15                      #pulse duration
lin_trans = 0.999                   #linear transmission (of Rhodamine B at 800nm)
al = -np.log(lin_trans)*1000        #linear absorption factor
L = 0.001                           #Cuvette length
L_eff = (1-np.exp(-L * al))/al      #effective cuvette length


#Formeln
'''
def trans(z, betta, name = '2022-06-21 17-46 RhodamineB 800nm 10.0mM 0.7996 W'):
    Power = float(name[-8:-2])
    q_0 = get_q_0(betta, Power, z)
    trans = 1 - (q_0)/(2) ** 1.5
    return trans

def get_q_0(betta, Power, z):#, L_eff = 0.0009999994997593358, z_0 = 0.001060083796):
    I_0 = P_to_I(Power)
    q_0 = betta * I_0 * L_eff / (1 + z ** 2/(z_0 ** 2))
    return q_0

def P_to_I(Power):#, w_0 = 28.33064e-06, R = 81.0e06, tau = 72.0e-15):
    I_0 = 4 * np.sqrt(np.log(2)/np.pi) * Power/ (np.pi * w_0 ** 2 * R * tau)
    return I_0

def Open_Aperture_Fit(z, data, norm, name, p0=4.8e-10):
    popt, pcov = curve_fit(lambda z, betta: trans(z, betta, name),
        (data[:,0]-norm[1])/1000, (data[:,1]/norm[0]), p0)
    return popt, pcov
```

```
'''
def trans(z, betta, name = '2022-06-21 17-46 RhodamineB 800nm 10.0mM 0.7996 W'):
    Power = float(name[-8:-2])
    q_0 = get_q_0(betta, Power, z)
    trans = 0
    for m in range(50):
        trans += (-q_0) **m / (m + 1) ** 1.5
    return trans

def get_q_0(betta, Power, z):#, L_eff = 0.0009999994997593358, z_0 = 0.001060083796):
    I_0 = P_to_I(Power)
    q_0 = betta * I_0 * L_eff / (1 + z ** 2/(z_0 ** 2))
    return q_0

def P_to_I(Power):#, w_0 = 28.33064e-06, R = 81.0e06, tau = 72.0e-15):
    I_0 = 4 * np.sqrt(np.log(2)/np.pi) * Power/ (np.pi * w_0 ** 2 * R * tau)
    return I_0

def Open_Aperture_Fit(z, data, norm, name, sigma, p0=1e-11):
    popt, pcov = curve_fit(lambda z, betta: trans(z, betta, name),
        (data[:,0]-norm[1])/1000, (data[:,1]/norm[0]), p0, sigma = sigma, maxfev=1000)
    return popt, pcov


filenames = glob(path + "/*.txt")
number_of_files = len(filenames)
print(number_of_files)

def load_data(path):
    data = np.loadtxt(path, skiprows=18, dtype='float')
    return data

def Chi_Square(obs_freq, exp_freq):
    chi_sq = 0
    for i in range(len(obs_freq)):
        x = (obs_freq[i] - exp_freq[i]) ** 2
        x = x / exp_freq[i]
        chi_sq += x
    return chi_sq

def save_image(data, trans_fit, z, chi_sq, num, norm, name, folder="Pictures_OA/"):
    fig = plt.figure(name)
    data[:,1] /=norm[0]
    data[:4,1] /= np.mean(data[:4,1])
    data[4:7,1] /= np.mean(data[4:7,1])*1.01
    data[-2:,1] /= np.mean(data[-2:,1])
    norm[0] = 1
    plt.plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
        linestyle = '', marker = 'o')#, label = 'Data ')
    plt.plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit ')
    #plt.xticks(np.arange(740,955,20))
    plt.title(name)
    #plt.legend(loc = "upper right", title = 'Chi Suqare = %.3f' %chi_sq)
    #plt.axis([-12.5, 12.5, 0.6, 1.1])
    #if np.min(data[:,1]/np.max(data[:,1])) < 0.65:
    plt.axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
    plt.xlabel('Stage postion (mm)')
    plt.ylabel('Transmission (arbitrary Unit)')
    #--- set figure properties ---
    fig.set_size_inches(fig_width, fig_high)
    mgr = plt.get_current_fig_manager()
    mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,fig_width*my_dpi,
        fig_high*my_dpi)
    #tight_layout()
    #plt.subplots_adjust(top = 0.95, left = 0.17, right = 0.98, bottom = 0.18)
        # distance small picture 11, 8
    plt.subplots_adjust(top = 0.95, left = 0.11, right = 0.97, bottom = 0.09, hspace = 0.26,
        wspace = 0.27)   # distance between the subplots, legend
    #plt.show()
    fig.savefig(folder + name + '.png', dpi=my_dpi*5)
    #fig.savefig('OA_example.eps', format='eps', dpi=my_dpi*5)
    plt.close()
```

```
#make one figure with 12 sufigures
if manyplot:
    fig_width_m = cm2inch(20)               # cm
    fig_high_m = cm2inch(27)                # cm
    #fig2, axs = plt.subplots(4, 3)
    subaxnum = round(num/12 - int(num/12), 6)
    multiname = 'MulitPicsOA/partie' + str(int(num/12))
    name = name[:3] + name[11:13] + '.' + name[8:10] + name[13:16]
        + ':' + name[17:19] +name[-9:]
    if subaxnum == 0/12:
        axs[0, 0].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[0, 0].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[0, 0].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[0, 0].set_title(name)
    if subaxnum == round(1/12, 6):
        axs[0, 1].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[0, 1].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[0, 1].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[0, 1].set_title(name)
    if subaxnum == round(2/12, 6):
        axs[0, 2].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[0, 2].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[0, 2].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[0, 2].set_title(name)
    if subaxnum == round(3/12, 6):
        axs[1, 0].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[1, 0].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[1, 0].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[1, 0].set_title(name)
    if subaxnum == round(4/12, 6):
        axs[1, 1].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[1, 1].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[1, 1].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[1, 1].set_title(name)
    if subaxnum == round(5/12, 6):
        axs[1, 2].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[1, 2].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[1, 2].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[1, 2].set_title(name)
    if subaxnum == round(6/12, 6):
        axs[2, 0].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[2, 0].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[2, 0].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[2, 0].set_title(name)
    if subaxnum == round(7/12, 6):
        axs[2, 1].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[2, 1].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[2, 1].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[2, 1].set_title(name)
    if subaxnum == round(8/12, 6):
        axs[2, 2].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[2, 2].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[2, 2].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[2, 2].set_title(name)
    if subaxnum == round(9/12, 6):
        axs[3, 0].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
        axs[3, 0].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
        axs[3, 0].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
        axs[3, 0].set_title(name)
    if subaxnum == round(10/12, 6):
        axs[3, 1].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
            linestyle = '', marker = 'o')#, label = 'Data ')
```

```
            axs[3, 1].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
            axs[3, 1].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
            axs[3, 1].set_title(name)
        if subaxnum == round(11/12, 6):
            axs[3, 2].plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0],
                linestyle = '', marker = 'o')#, label = 'Data')
            axs[3, 2].plot(z*1000, trans_fit, color = Farben.co[2], linestyle = '-')#, label = 'fit')
            axs[3, 2].axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
            axs[3, 2].set_title(name)

            for ax in axs.flat:
                ax.get_yaxis().set_major_formatter(FormatStrFormatter('%.2f'))
            axs[0,0].set(ylabel='Transmission (arbitrary Unit)')
            axs[1,0].set(ylabel='Transmission (arbitrary Unit)')
            axs[2,0].set(ylabel='Transmission (arbitrary Unit)')
            axs[3,0].set(xlabel='Stage postion (mm)', ylabel='Transmission (arbitrary Unit)')
            axs[3,1].set(xlabel='Stage postion (mm)')
            axs[3,2].set(xlabel='Stage postion (mm)')

               #--- set figure properties ---
            fig2.set_size_inches(fig_width_m, fig_high_m)
            mgr = plt.get_current_fig_manager()
            mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,
                fig_width_m*my_dpi,fig_high_m*my_dpi)
            plt.tight_layout()
            plt.subplots_adjust(top = 0.97, left = 0.1, right = 0.98,
                bottom = 0.05, wspace = 0.24, hspace = 0.245)
            #plt.show()
            print(multiname)
            fig2.savefig(multiname + '.png', dpi=my_dpi*5)
            fig2.savefig(multiname + '.eps', format='eps', dpi=my_dpi*5)
            plt.close()

def remove_outliers(data, name):
    count = np.array([0])
    for i in range(len(data)):
        if (data[i,1] > 100 or data[i,1] == 0):
            count = np.append(count, i)
            count[0] += 1
    if not count[0] == 0:
        print('%i outliers detected and removed in ' %count[0] + name)
    count = np.delete(count, 0)
    data = np.delete(data, count,0)
    return data

def save_data(betta_fit, fit_error, chi_sq, name):
    fileName = 'Z_Scan_Data.csv'  #'Z_Scan_Data_OA_CA_Power.csv'
    date_time_str = name[0:16]
    Power = float(name[-8:-2])
    lam = float(name[-21:-18]) * 10 ** (-9)
    rho = 0.01e4
    sigma = constant.h * constant.c / lam * betta_fit / (constant.N_A * rho)
        #rho, lamda, c should be changed from m to cm or afterwards everything * 1e58 for GM

    if os.path.exists(fileName) == False:
        createHeader = True
    else:
        createHeader = False
    with open(fileName, 'a',newline='') as fileObj:
        writer = csv.writer(fileObj)
        if createHeader:
            writer.writerow(['date and time', 'Power (W)', 'sigma', 'betta', 'Chi Square',
                'AOM frequenzy (Hz)', 'AOM duty (%)', 'Open Chopper'])
        if name[1] == '1':
            frq = 137
            duty = 1
            on = 1000/frq * (0.01 * duty)
        if name[1] == '2':
            frq = 150
            duty = 1
            on = 1000/frq * (0.01 * duty)
        if name[1] == '3':
```

```
            frq = 200
            duty = 1
            on = 1000/frq * (0.01 * duty)
        if name[1] == '4':
            frq = 274
            duty = 2
            on = 1000/frq * (0.01 * duty)
        if name[1] == '5':
            frq = 400
            duty = 1
            on = 1000/frq * (0.01 * duty)
        if name[1] == '6':
            frq = 300
            duty = 1
            on = 1000/frq * (0.01 * duty)
        if name[1] == '7':
            frq = 120
            duty = 1
            on = 1000/frq * (0.01 * duty)
        if name[1] == '8':
            frq = 62
            duty = 2.8
            on = 1000/frq * (0.01 * duty)
        if name[1] == '9':
            frq = 137
            duty = 99
            on = 1000/frq * (0.01 * duty)
        writer.writerow([date_time_str, Power, sigma[0], betta_fit[0], chi_sq, frq, duty, on])

chi_sq = np.zeros([number_of_files])
fit_start = time.perf_counter()
time_20 = fit_start
count = 0
for i in range(number_of_files):
    #i=1
    data = load_data(filenames[i])
    name = filenames[i].replace(path,'')
    name = name.replace('\\','')
    data = remove_outliers(data, name)
    name = name.replace('_+0.0_ml_p_h.txt','')
    if not data[:,1].size < 34:
        if manyplot:
            if count in range(0, number_of_files, 12):
                print(i, count)
                plt.close()
                font = {'size'   : 9}
                plt.rc('font', **font)
                fig2, axs = plt.subplots(4, 3, sharex='col')#, sharey='row')
        if divide_data:
            data[:,1] = data[:,1] / data[:,3]
        max_ind = np.argpartition(data[:,1], -4)[-4:]
        min_ind = np.argpartition(data[:,1], 3)[:3]
        norm = [np.sum(data[max_ind,1])/4, np.sum(data[min_ind,0])/3]
        if not 11 < norm[1] <13:
            print(norm, name)
        sigma =np.ones(len(data[:,1]))
        sigma[min_ind] = 0.01
        betta_fit, fit_error = Open_Aperture_Fit(z, data, norm, name, sigma = sigma)
        trans_fit = trans(z, betta_fit, name)
        if np.min(trans_fit) - np.min(data[:,1]/norm[0]) > 0.002:
            betta_fit = betta_fit*(1.02+4*np.abs(np.min(trans_fit) - np.min(data[:,1]/norm[0])))
            trans_fit = trans(z, betta_fit, name)
        chi_sq[i] = Chi_Square(data[:,1], trans_fit)
        #save_image(data, trans_fit, z, chi_sq[i], count, norm, name)
        save_data(betta_fit, fit_error, chi_sq[i], name)
        count +=1
    if i in range(20, number_of_files, 20):      #Fortschritt sehen
        time_20 = time.perf_counter() - fit_start
        print(i, time_20)
if manyplot:
    for ax in axs.flat:
        ax.get_yaxis().set_major_formatter(FormatStrFormatter('%.2f'))
```

```
    axs[0,0].set(ylabel='Transmission (arbitrary Unit)')
    axs[1,0].set(ylabel='Transmission (arbitrary Unit)')
    axs[2,0].set(ylabel='Transmission (arbitrary Unit)')
    axs[3,0].set(xlabel='Stage postion (mm)', ylabel='Transmission (arbitrary Unit)')
    axs[3,1].set(xlabel='Stage postion (mm)')
    axs[3,2].set(xlabel='Stage postion (mm)')
    fig_width_m = cm2inch(20)              # cm
    fig_high_m = cm2inch(27)               # cm
    #--- set figure properties ---
    fig2.set_size_inches(fig_width_m, fig_high_m)
    mgr = plt.get_current_fig_manager()
    mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,fig_width_m*my_dpi,fig_high_m*my_dpi)
    plt.tight_layout()
    plt.subplots_adjust(top = 0.97, left = 0.1, right = 0.98, bottom = 0.05,
        wspace = 0.24, hspace = 0.245)
    #plt.show()
    fig2.savefig('MulitPicsOA/partieLast.png', dpi=my_dpi*5)
    fig2.savefig('MulitPicsOA/partieLast.eps', format='eps', dpi=my_dpi*5)
    plt.close()

fit_time = time.perf_counter()-fit_start
print('Time for the fit: {:.1f} s'.format(fit_time))
```

C.2 Analyse_CA.py

```
'''
Textfiles von Closed Aperture Measurement einlesen, outliers rausschmeisssn,
Fit erstellen/anpassen, Grafik erstellen, Daten Speichern
'''
from fileinput import filename
import numpy as np #numerische methoden
import matplotlib.pyplot as plt #plotlibrary
import os   #pfade
import csv
from glob import glob #find specific stuff in directorys (can probably do more)
from scipy.optimize import curve_fit
import scipy.constants as constant
import time
import Farben

def find_index(array, value):  #finds index of first occurence if value in array
    i = 0
    for i in range(len(array)):
        if array[i] == value:
            return i

def cm2inch(cm):
    inch = 2.54
    return int(cm/inch)

fig_width = cm2inch(20)              # cm
fig_high = cm2inch(13)               # cm
my_dpi = 100                         # cm

path = r"D:\Documents\Uni\MasterArbeit\RhodamineB_cleaned\closedAP"
path_OA = r"D:\Documents\Uni\MasterArbeit\RhodamineB_cleaned"

divide_data = False #divide the data by data from Open Aperture experiments


#Bekannte Variablen:
w_0 = 28.33064e-06              #beam Waist
z_0 = 0.001060083796            #Raileygh Lenght
z = np.linspace(-13, 13, 910)/1000    #z array for fitting
P = 0                           #Power ... Varieiert
R = 81.0e06                     #Repetition Rate
tau = 72.0e-15                  #pulse duration
lin_trans = 0.999               #linear transmission (of Rhodamine B at 800nm)
al = -np.log(lin_trans) *1000   #linear absorption factor
```

```
L = 0.001                           #Cuvette length
L_eff = (1-np.exp(-L * al))/al      #effective cuvette length
eta = 2                             #order of absorption

def P_to_I(Power):#, w_0 = 28.33064e-06, R = 81.0e06, tau = 72.0e-15):
    I_0 = 4 * np.sqrt(np.log(2)/np.pi) * Power/ (np.pi * w_0 ** 2 * R * tau)
    return I_0

def trans_CA_ref(z, dn):
    T_ref = 1-(4 * dn * (z/z_0))/(((z/z_0) ** 2 + 1) * ((z/z_0) **2 +9))
    return T_ref

def CA_fit_ref(z, norm, data, p0, sigma):
    popt, pcov = curve_fit(lambda z, dn: trans_CA_ref(z, dn), (data[:,0]-norm[1])/1000,
        (data[:,1]/norm[0]), p0, sigma = sigma)
    return popt, pcov

def get_n2(lam, dn, Power):
    I = P_to_I(Power)
    n2 = (lam * dn) / (2 * np.pi * I * L_eff)
    return n2



def trans_CA_therm(z, theta):
    T_therm = 1 + theta/eta * 1 / ((1 + (z/z_0) ** 2) ** (eta - 1)) *
        np.arctan((2 * eta * (z/z_0)) / (2 * eta + 1 + (z/z_0) ** 2)) * 180/np.pi
    return T_therm

def CA_fit_therm(z, norm, data, p0, sigma):
    popt, pcov = curve_fit(lambda z, theta: trans_CA_therm(z, theta),
        (data[:,0]-norm[1])/1000, (data[:,1]/norm[0]), p0, sigma = sigma)
    return popt, pcov

def get_N_sig(theta, dn, P, lam):
    N_sig = np.sqrt(2 * np.log(2)) * (constant.h * constant.c * al * theta) /
        (eta * P * lam * dn) * (2 / (np.pi * w_0 ** 2)) ** (1 - eta)
    return N_sig

def get_b_therm(N_sig, lam):
    return 2 * N_sig *lam / (constant.h * constant.c)

filenames = glob(path + "/*.txt")
number_of_files = len(filenames)
print('files in folder:', number_of_files)

filenames_OA = glob(path_OA + "/*.txt")

def load_data(path):
    data = np.loadtxt(path, skiprows=18, dtype='float')
    return data

def remove_outliers(data, name):
    count = np.array([0])
    for i in range(len(data)):
        if (data[i,1] > 100 or data[i,1] == 0):
            count = np.append(count, i)
            count[0] += 1
    if not count[0] == 0:
        print('%i outliers detected and removed in ' %count[0] + name)
    count = np.delete(count, 0)
    data = np.delete(data, count,0)
    return data



def save_data(theta, dn, name):
    fileName = 'Z_Scan_Data_CA.csv'
    date_time_str = name[3:19]
    Power = float(name[-8:-2])
    lam = float(name[-21:-18]) * 10 ** (-9)
    N_sig = get_N_sig(np.abs(theta), dn, Power, lam)
```

```
    b_therm = get_b_therm(N_sig, lam) #rho, lamda, c should be changed from m to cm
    n2 = get_n2(lam, dn, Power)
    I = P_to_I(Power)
    H = np.sqrt(2/np.pi) * (Power * lam) ** 2 / ((R * constant.h * constant.c) ** 2 * tau)
    dn_2 = dn*lam/(2 * np.pi * L_eff)
    #N_sigma_2nd = (np.abs(theta) * lam ** 2 * al * w_0**2 * I)/ (L_eff * eta * constant.h *
        * constant.c * H * R * dn_2 * 4) * ((2/(np.pi * w_0 ** 2)) ** (1-eta))
    frq = 'what'
    duty = 'the'
    on = 'fuck'
    if name[1] == '1':
        frq = 137
        duty = 1
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '2':
        frq = 150
        duty = 1
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '3':
        frq = 200
        duty = 1
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '4':
        frq = 274
        duty = 2
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '5':
        frq = 400
        duty = 1
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '6':
        frq = 300
        duty = 1
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '7':
        frq = 120
        duty = 1
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '8':
        frq = 62
        duty = 2.8
        on = 1000000/frq * (0.01 * duty)
    if name[1] == '9':
        frq = 137
        duty = 99
        on = 1000000/frq * (0.01 * duty)
    pulses_chopper = on*1e-6 *R
    pulses_second = pulses_chopper * frq
    ueberrpuefung = pulses_second / duty *100

    if os.path.exists(fileName) == False:
        createHeader = True
    else:
        createHeader = False

    with open(fileName, 'a',newline='') as fileObj:
        writer = csv.writer(fileObj)
        if createHeader:
            writer.writerow(['date and time', 'Power (W)', 'N * sigma_therm', 'betta_therm',
                'non linear refractive index','theta', "dphi", 'dn', 'H', 'I',
                            'AOM frequenzy (Hz)', 'AOM duty (%)', 'Open Chopper (us)',
                               'pulses/shutter opening', 'pulses/second', 'pulse richtig?'])

        writer.writerow([date_time_str, Power, N_sig[0], b_therm[0], n2[0], theta[0], dn[0],
            dn_2[0], H, I, frq, duty, on, pulses_chopper, pulses_second, ueberrpuefung])

def save_image(data, trans_fit_therm, trans_fit_ref, z, norm, name, folder="PicturesCA/"):
    fig = plt.figure(name)
    plt.plot((data[:,0]-norm[1]), data[:,1]/norm[0], color= Farben.co[0], linestyle = '',
        marker = 'o')#, label = 'Data ')
    plt.plot(z*1000, trans_fit_therm, color = Farben.co[1], linestyle = '-',
        label = 'thermo - fit')
```

```
#plt.plot(z*1000, trans_fit_ref, color = Farben.co[2], linestyle = '-',
    label = 'reffractive - fit')
#plt.xticks(np.arange(740,955,20))
plt.title(name)
plt.legend(loc = "upper right")
#plt.axis([-12.5, 12.5, 0.6, 1.1])
#if np.min(data[:,1]/np.max(data[:,1])) < 0.65:
#plt.axis([-norm[1]-0.5, norm[1]+0.5, np.min(data[:,1]/np.max(data[:,1]))-0.05, 1.05])
plt.xlabel('Stage postion (mm)')
plt.ylabel('Transmission (arbitrary Unit)')
#--- set figure properties ---
fig.set_size_inches(fig_width, fig_high)
mgr = plt.get_current_fig_manager()
mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,
    fig_width*my_dpi,fig_high*my_dpi)
#tight_layout()
#plt.subplots_adjust(top = 0.75, left = 0.15, right = 0.98, bottom = 0.15)
    # distance small picture 11, 8
plt.subplots_adjust(top = 0.95, left = 0.11, right = 0.97, bottom = 0.09,
    hspace = 0.26, wspace = 0.27)   # distance between the subplots, legend
plt.show()
fig.savefig(folder + name + '.png', dpi=my_dpi*5)
#fig.savefig('CA_example.eps', format='eps', dpi=my_dpi*5)
plt.close()

fit_start = time.perf_counter()
time_20 = fit_start
data_OA_600 = np.zeros([9,35])
if divide_data:
    for i in range(len(filenames_OA)):
        if filenames_OA[i][50] == '1' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[0,:] = data_OA[:,1]
        if filenames_OA[i][50] == '2' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[1,:] = data_OA[:,1]
        if filenames_OA[i][50] == '3' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[2,:] = data_OA[:,1]
        if filenames_OA[i][50] == '4' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[3,:] = data_OA[:,1]
        if filenames_OA[i][50] == '5' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[4,:] = data_OA[:,1]
        if filenames_OA[i][50] == '6' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[5,:] = data_OA[:,1]
        if filenames_OA[i][50] == '7' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[6,:] = data_OA[:,1]
        if filenames_OA[i][50] == '8' and filenames_OA[i][-24:-20] == '0.60':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[7,:] = data_OA[:,1]
        if filenames_OA[i][50] == '9' and filenames_OA[i][-24:-20] == '0.10':
            data_OA = load_data(filenames_OA[i])
            data_OA_600[8,:] = data_OA[:,1]

for i in range(1):#number_of_files):
    i=-3
    data = load_data(filenames[i])
    name = filenames[i].replace(path,'')
    name = name.replace('\\','')
    data = remove_outliers(data, name)
    name = name.replace('_+0.0_ml_p_h.txt','')
    if not data[:,1].size < 34:
        if divide_data:
            int_exp = int(name[1]) -1
            data[:,1] /= data_OA_600[int_exp,:]
        lam = float(name[-21:-18]) * 10 ** (-9)
        max_index = find_index(data[:,1], np.max(data[:,1]))
        min_index = find_index(data[:,1], np.min(data[:,1]))
```

```
        focus = int(max_index + (min_index - max_index)/2)
        norm = [(data[max_index, 1] + data[min_index, 1])/2, data[focus,0]]
            #sum(data[1:7,1]/6), data[focus,0]]#norm = [np.mean(data[:,1]),
            data[focus,0]]#sum(data[1:7,1]/6), data[focus,0]]
        sigma =np.ones(len(data[:,1]))
        sigma[[max_index, min_index]] = 0.01
        ref_fit, ref_error = CA_fit_ref(z, norm, data, p0 = 0.5, sigma = sigma)
        therm_fit, fit_error = CA_fit_therm(z, norm, data, p0 = 0.5, sigma = sigma)
        trans_fit_ref = trans_CA_ref(z, ref_fit)
        trans_fit_therm = trans_CA_therm(z, therm_fit)
        print(ref_fit, therm_fit)
        #save_data(therm_fit, ref_fit, name)
        save_image(data, trans_fit_therm, trans_fit_ref, z, norm, name)
    if i in range(20, number_of_files, 20):      #Fortschritt sehen
        time_20 = time.perf_counter() - fit_start
        print(i, time_20)
fit_time = time.perf_counter()-fit_start
print('Time for the fit: {:.1f} s'.format(fit_time))
```

C.3 AOM_Opening_times.py

```
from fileinput import filename
import numpy as np #numerische methoden
import math as math
import matplotlib.pyplot as plt #plotlibrary
import os   #pfade
import csv
from glob import glob #find specific stuff in directorys (can probably do more)
from scipy.optimize import curve_fit
import scipy.constants as constant
import Farben
import pandas as pd
import Farben

def find_index(array, value):
    return array[array <= value].size

def cm2inch(cm):
    inch = 2.54
    return int(cm/inch)

fig_width = cm2inch(25)             # cm
fig_high = cm2inch(14)              # cm
my_dpi = 100

df = pd.read_csv('Z_Scan_data.csv', header = 0)
data = df.to_numpy()
df = pd.read_csv('Z_Scan_Data_CA_etachange.csv', header = 0)
data_CA = df.to_numpy()

delete_bad_fits = False     #I marked some mediocre fits as "schiach", if True those are deleted
delete_undivided = False    #Only takes takes data where the CA data was divided by OA data

open_chopper = np.zeros([9,3])      #Saves AOM frequency, Duty and On time
average_sigma = np.zeros([9,19])
average_betta = np.zeros([9,19])

count = 0
num = 0
power_count = 0
power_num = 0
all_sigmas_m1 = []
all_sigmas_m2 = []
all_sigmas_m3 = []
all_sigmas_m4 = []
all_sigmas_m5 = []
all_sigmas_m6 = []
all_sigmas_m7 = []
all_sigmas_m8 = []
```

```
all_sigmas_m9 = []
m_per_power = np.zeros([9,18])
for i in range(len(data)-1):                    #goes to the number of files -1 so each line can be compared to the one after it

    if not data[i, 5] == 0:
        if 0.29 < data[i,1] < 0.31:             #gives the different measured powers the number power num from 0-17 corresponding to 300-1150 mW in 50 mW steps

            power_num = 0                       #one measurement was performed at 70, 80, 100 and 150 mW, those are saved at the same place as 300, 350, 400, 450 mW measurement from the other experiments


        elif 0.068 < data[i,1] < 0.072:
            power_num = 0
        elif 0.078 < data[i,1] < 0.082:
            power_num = 1
        elif 0.09 < data[i,1] < 0.11:
            power_num = 2
        elif 0.14 < data[i,1] < 0.16:
            power_num = 3
        elif 0.34 < data[i,1] < 0.36:
            power_num = 1
        elif 0.39 < data[i,1] < 0.41:
            power_num = 2
        elif 0.44 < data[i,1] < 0.46:
            power_num = 3
        elif 0.49 < data[i,1] < 0.51:
            power_num = 4
        elif 0.54 < data[i,1] < 0.56:
            power_num = 5
        elif 0.59 < data[i,1] < 0.61:
            power_num = 6
        elif 0.64 < data[i,1] < 0.66:
            power_num = 7
        elif 0.69 < data[i,1] < 0.71:
            power_num = 8
        elif 0.74 < data[i,1] < 0.76:
            power_num = 9
        elif 0.79 < data[i,1] < 0.81:
            power_num = 10
        elif 0.84 < data[i,1] < 0.86:
            power_num = 11
        elif 0.89 < data[i,1] < 0.91:
            power_num = 12
        elif 0.94 < data[i,1] < 0.96:
            power_num = 13
        elif 0.99 < data[i,1] < 1.01:
            power_num = 14
        elif 1.04 < data[i,1] < 1.06:
            power_num = 15
        elif 1.09 < data[i,1] < 1.11:
            power_num = 16
        elif 1.14 < data[i,1]:
            power_num = 17
        if num == 0:
            all_sigmas_m1.append(data[i][2]*1e58)
        if num == 1:
            all_sigmas_m2.append(data[i][2]*1e58)
        if num == 2:
            all_sigmas_m3.append(data[i][2]*1e58)
        if num == 3:
            all_sigmas_m4.append(data[i][2]*1e58)
        if num == 4:
            all_sigmas_m5.append(data[i][2]*1e58)
        if num == 5:
            all_sigmas_m6.append(data[i][2]*1e58)
        if num == 6:
            all_sigmas_m7.append(data[i][2]*1e58)
        if num == 7:
            all_sigmas_m8.append(data[i][2]*1e58)
        if num == 8:
            all_sigmas_m9.append(data[i][2]*1e58)
        average_sigma[num, power_num] = average_sigma[num, power_num] + data[i][2]
        #saves the data of experiement num with the same power in average sigma[num, power num]
```

```
            average_sigma[num, 18] = average_sigma[num, 18] + data[i][2]
            average_betta[num, power_num] = average_betta[num, power_num] + data[i][3]
            average_betta[num, 18] = average_betta[num, 18] + data[i][3]
            count += 1                  #for counting the number of files in each ecperiment
            power_count += 1            #counting the number of files of each power in each experiment
            if data[i,1] != data[i+1, 1]:
            #checks if the power changes, if so the average sigma/betta of the current power is taken
            #and the power counter resetted.
                average_sigma[num, power_num] /= power_count
                average_betta[num, power_num] /= power_count
                m_per_power[num, power_num] = power_count
                power_count = 0
            if (data[i, 5] != data[i+1, 5] and data[i, 7] != data[i+1, 7]):
            #checks if an experiment set changed, takes averages resets power count & count, increases num
                average_sigma[num, 18] /= count
                average_betta[num, 18] /= count
                open_chopper[num,0] = data[i,5]
                open_chopper[num,1] = data[i,6]
                open_chopper[num,2] = data[i,7]
                num += 1
                #print(i, num, data[i,0])
                count = 0
    #because the loop only goes to len-1 for the last element
    print(num, i, power_num)
    all_sigmas_m9.append(data[len(data)-1][2]*1e58)
    average_sigma[num, power_num] = average_sigma[num, power_num] + data[len(data)-1][2]
    average_sigma[num, 18] = average_sigma[num, 18] + data[len(data)-1][2]
    average_betta[num, power_num] = average_betta[num, power_num] + data[len(data)-1][3]
    average_betta[num, 18] = average_betta[num, 18] + data[len(data)-1][3]
    count += 1
    power_count += 1
    average_sigma[num, power_num] /= power_count
    average_betta[num, power_num] /= power_count
    m_per_power[num, power_num] = power_count
    average_sigma[num, 18] /= count
    average_betta[num, 18] /= count
    open_chopper[num,0] = data[len(data)-1,5]
    open_chopper[num,1] = data[len(data)-1,6]
    open_chopper[num,2] = data[len(data)-1,7]

    print(m_per_power)
    #As above for the average values, now for the deviation
    average_deviation = np.zeros([9,19])
    average_deviation_betta = np.zeros([9,19])
    measurements_number = np.zeros([9,19])
    count = 0
    num = 0
    power_count = 0
    for i in range(len(data)-1):
        if not data[i, 5] == 0 or i > len(data):
            if 0.29 < data[i,1] < 0.31:
                power_num = 0
            elif 0.068 < data[i,1] < 0.072:
                power_num = 0
            elif 0.078 < data[i,1] < 0.082:
                power_num = 1
            elif 0.09 < data[i,1] < 0.11:
                power_num = 2
            elif 0.14 < data[i,1] < 0.16:
                power_num = 3
            elif 0.34 < data[i,1] < 0.36:
                power_num = 1
            elif 0.39 < data[i,1] < 0.41:
                power_num = 2
            elif 0.44 < data[i,1] < 0.46:
                power_num = 3
            elif 0.49 < data[i,1] < 0.51:
                power_num = 4
            elif 0.54 < data[i,1] < 0.56:
                power_num = 5
            elif 0.59 < data[i,1] < 0.61:
                power_num = 6
```

```
        elif 0.64 < data[i,1] < 0.66:
            power_num = 7
        elif 0.69 < data[i,1] < 0.71:
            power_num = 8
        elif 0.74 < data[i,1] < 0.76:
            power_num = 9
        elif 0.79 < data[i,1] < 0.81:
            power_num = 10
        elif 0.84 < data[i,1] < 0.86:
            power_num = 11
        elif 0.89 < data[i,1] < 0.91:
            power_num = 12
        elif 0.94 < data[i,1] < 0.96:
            power_num = 13
        elif 0.99 < data[i,1] < 1.01:
            power_num = 14
        elif 1.04 < data[i,1] < 1.06:
            power_num = 15
        elif 1.09 < data[i,1] < 1.11:
            power_num = 16
        elif 1.14 < data[i,1]:
            power_num = 17

        average_deviation[num, power_num] = average_deviation[num, power_num] +
            np.abs(data[i][2] - average_sigma[num, power_num])
        average_deviation[num, 18] = average_deviation[num, 18] + np.abs(data[i][2] - average_sigma[num, 18])
        average_deviation_betta[num, power_num] = average_deviation_betta[num, power_num] +
            np.abs(data[i][3] - average_betta[num, power_num])
        average_deviation_betta[num, 18] = average_deviation_betta[num, 18] +
        np.abs(data[i][3] - average_betta[num, 18])
        count += 1
        power_count += 1
        if data[i,1] != data[i+1, 1]:
            average_deviation[num, power_num] /= power_count
            average_deviation_betta[num, power_num] /= power_count
            measurements_number[num, power_num] = power_count
            if not average_sigma[num, power_num] == 0:
                if math.isclose(average_sigma[num, power_num],  average_deviation[num, power_num], rel_tol=0.8)
                        or average_deviation[num, power_num] == 0:
                    print('SIGMA Fcuek', data[i, 0])
                    print(num, power_num, power_count, average_deviation[num, power_num])
                if math.isclose(average_betta[num, power_num], average_deviation_betta[num, power_num],
                        rel_tol=0.8) or average_deviation_betta[num, power_num] == 0:
                    print("YO Betta was geht", data[i, 0])
                    print(num, power_num, power_count, average_deviation[num, power_num])
            power_count = 0

        if (data[i, 5] != data[i+1, 5] and data[i, 7] != data[i+1, 7]):
            average_deviation[num, 18] /= count
            average_deviation_betta[num, 18] /= count
            measurements_number[num, 18] = count
            num += 1
            count = 0
average_deviation[num, power_num] = average_deviation[num, power_num] +
    np.abs(data[len(data)-1][2] - average_sigma[num, power_num])
average_deviation[num, 18] = average_deviation[num, 18] + np.abs(data[len(data)-1][2] - average_sigma[num, 18])
average_deviation_betta[num, power_num] = average_deviation_betta[num, power_num] +
    np.abs(data[i][2] - average_betta[num, power_num])
average_deviation_betta[num, 18] = average_deviation_betta[num, 18] +
    np.abs(data[i][2] - average_betta[num, 18])
count += 1
power_count += 1
average_deviation[num, power_num] /= power_count
average_deviation[num, 18] /= count
average_deviation_betta[num, power_num] /= power_count
average_deviation_betta[num, 18] /= count
measurements_number[num, 18] = count
measurements_number[num, power_num] = power_count


#_____________
#average sigma2 of the averagepowers for non weigheted averages. Afterwards deviation
sigma_power_averages = np.zeros([9,2])
```

```
count = 0
for j in range(9):
    for i in range(18):
        if not average_sigma[j,i] == 0:
            sigma_power_averages[j,0] += average_sigma[j,i]
            count += 1
            #print('in power loop', j, i, count, average_sigma[j,i], count)
    sigma_power_averages[j,0] /= count
    #print('in experiment loop', j, count)
    count = 0

for j in range(9):
    for i in range(18):
        if not average_sigma[j,i] == 0:
            sigma_power_averages[j,1] += np.abs(average_sigma[j,i]-sigma_power_averages[j,0])
            count += 1
    sigma_power_averages[j,1] /= count
    count = 0




betta_plot = np.zeros([9,2])          #OA betta werte mit passender power zu CA messungen
for i in range(8):
    betta_plot[i,0] = average_betta[i, 6]
    betta_plot[i,1] = average_deviation_betta[i, 6]
betta_plot[8,0] = average_betta[8, 2]
betta_plot[8,1] = average_deviation_betta[8, 2]
#print(betta_plot)



#####################EINSCHUB VON DEN CA SCANS#####################################
averages_CA = np.zeros([9,7])
measurements_number_CA = np.zeros(9)
count = 0
num = 0
#print(len(data_CA))
for i in range(len(data_CA)-1):
    averages_CA[num][1] += data_CA[i][2]      #N*Sigma
    averages_CA[num][2] += data_CA[i][3]      #betta_therm
    averages_CA[num][3] += data_CA[i][4]      #n2
    averages_CA[num][4] += data_CA[i][5]      #theta
    averages_CA[num][5] += data_CA[i][6]      #dphi
    averages_CA[num][6] += data_CA[i][7]      #dn
    count += 1
    if delete_bad_fits:
        if data_CA[i][-2] == 1:
            averages_CA[num][1] -= data_CA[i][2]      #N*Sigma
            averages_CA[num][2] -= data_CA[i][3]      #betta_therm
            averages_CA[num][3] -= data_CA[i][4]      #n2
            averages_CA[num][4] -= data_CA[i][5]      #theta
            averages_CA[num][5] -= data_CA[i][6]      #dphi
            averages_CA[num][6] -= data_CA[i][7]      #dn
            count -= 1
    if delete_undivided:
        if data_CA[i][-1] == 1:
            averages_CA[num][1] -= data_CA[i][2]      #N*Sigma
            averages_CA[num][2] -= data_CA[i][3]      #betta_therm
            averages_CA[num][3] -= data_CA[i][4]      #n2
            averages_CA[num][4] -= data_CA[i][5]      #theta
            averages_CA[num][5] -= data_CA[i][6]      #dphi
            averages_CA[num][6] -= data_CA[i][7]      #dn
            count -= 1
    if not data_CA[i][-4] == data_CA[i+1][-4]:
        #print(i, num, data_CA[i][-4], data_CA[i+1][-4])
        averages_CA[num][0] = data_CA[i][1]      #Power
        averages_CA[num][1] = averages_CA[num][1]/count
        averages_CA[num][2] = averages_CA[num][2]/count
        averages_CA[num][3] = averages_CA[num][3]/count
        averages_CA[num][4] = averages_CA[num][4]/count
```

```
            averages_CA[num][5] = averages_CA[num][5]/count
            averages_CA[num][6] = averages_CA[num][6]/count
            measurements_number_CA[num] = count
            count = 0
            num += 1
    #print(num)
    averages_CA[num][0] = data_CA[len(data_CA)-1][1]        #Power
    averages_CA[num][1] += data_CA[len(data_CA)-1][2]        #N*Sigma
    averages_CA[num][2] += data_CA[len(data_CA)-1][3]        #betta_therm
    averages_CA[num][3] += data_CA[len(data_CA)-1][4]        #n2
    averages_CA[num][4] += data_CA[len(data_CA)-1][5]        #theta
    averages_CA[num][5] += data_CA[len(data_CA)-1][6]        #dphi
    averages_CA[num][6] += data_CA[len(data_CA)-1][7]        #dn
    count += 1
    if delete_bad_fits:
        if data_CA[i][-2] == 1:
            averages_CA[num][1] -= data_CA[i][2]        #N*Sigma
            averages_CA[num][2] -= data_CA[i][3]        #betta_therm
            averages_CA[num][3] -= data_CA[i][4]        #n2
            averages_CA[num][4] -= data_CA[i][5]        #theta
            averages_CA[num][5] -= data_CA[i][6]        #dphi
            averages_CA[num][6] -= data_CA[i][7]        #dn
            count -= 1
    if delete_undivided:
        if data_CA[i][-1] == 1:
            averages_CA[num][1] -= data_CA[i][2]        #N*Sigma
            averages_CA[num][2] -= data_CA[i][3]        #betta_therm
            averages_CA[num][3] -= data_CA[i][4]        #n2
            averages_CA[num][4] -= data_CA[i][5]        #theta
            averages_CA[num][5] -= data_CA[i][6]        #dphi
            averages_CA[num][6] -= data_CA[i][7]        #dn
            count -= 1
averages_CA[num][1] = averages_CA[num][1]/count
averages_CA[num][2] = averages_CA[num][2]/count
averages_CA[num][3] = averages_CA[num][3]/count
averages_CA[num][4] = averages_CA[num][4]/count
averages_CA[num][5] = averages_CA[num][5]/count
averages_CA[num][6] = averages_CA[num][6]/count
measurements_number_CA[num] = count
count = 0
num = 0


average_deviation_CA = np.zeros([9,7])
for i in range(len(data_CA)-1):
    average_deviation_CA[num][1] += np.abs(averages_CA[num][1] - data_CA[i][2])      #N*Sigma
    average_deviation_CA[num][2] += np.abs(averages_CA[num][2] - data_CA[i][3])      #betta_therm
    average_deviation_CA[num][3] += np.abs(averages_CA[num][3] - data_CA[i][4])      #n2
    average_deviation_CA[num][4] += np.abs(averages_CA[num][4] - data_CA[i][5])      #theta
    average_deviation_CA[num][5] += np.abs(averages_CA[num][5] - data_CA[i][6])      #dphi
    average_deviation_CA[num][6] += np.abs(averages_CA[num][6] - data_CA[i][7])      #dn
    count += 1
    if delete_bad_fits:
        if data_CA[i][-2] == 1:
            average_deviation_CA[num][1] -= np.abs(averages_CA[num][1] - data_CA[i][2])      #N*Sigma
            average_deviation_CA[num][2] -= np.abs(averages_CA[num][2] - data_CA[i][3])      #betta_therm
            average_deviation_CA[num][3] -= np.abs(averages_CA[num][3] - data_CA[i][4])      #n2
            average_deviation_CA[num][4] -= np.abs(averages_CA[num][4] - data_CA[i][5])      #theta
            average_deviation_CA[num][5] -= np.abs(averages_CA[num][5] - data_CA[i][6])      #dphi
            average_deviation_CA[num][6] -= np.abs(averages_CA[num][6] - data_CA[i][7])      #dn
            count -= 1
    if delete_undivided:
        if data_CA[i][-1] == 0:
            average_deviation_CA[num][1] -= np.abs(averages_CA[num][1] - data_CA[i][2])      #N*Sigma
            average_deviation_CA[num][2] -= np.abs(averages_CA[num][2] - data_CA[i][3])      #betta_therm
            average_deviation_CA[num][3] -= np.abs(averages_CA[num][3] - data_CA[i][4])      #n2
            average_deviation_CA[num][4] -= np.abs(averages_CA[num][4] - data_CA[i][5])      #theta
            average_deviation_CA[num][5] -= np.abs(averages_CA[num][5] - data_CA[i][6])      #dphi
            average_deviation_CA[num][6] -= np.abs(averages_CA[num][6] - data_CA[i][7])      #dn
            count -= 1
    if not data_CA[i][-4] == data_CA[i+1][-4]:
        average_deviation_CA[num][0] = data_CA[i][1]        #Power
```

```
            average_deviation_CA[num][1] = average_deviation_CA[num][1]/count
            average_deviation_CA[num][2] = average_deviation_CA[num][2]/count
            average_deviation_CA[num][3] = average_deviation_CA[num][3]/count
            average_deviation_CA[num][4] = average_deviation_CA[num][4]/count
            average_deviation_CA[num][5] = average_deviation_CA[num][5]/count
            average_deviation_CA[num][6] = average_deviation_CA[num][6]/count
            count = 0
            num += 1
    average_deviation_CA[num][0] = data_CA[len(data_CA)-1][1]      #Power
    average_deviation_CA[num][1] += np.abs(averages_CA[num][1] - data_CA[len(data_CA)-1][2])      #N*Sigma
    average_deviation_CA[num][2] += np.abs(averages_CA[num][2] - data_CA[len(data_CA)-1][3])      #betta_therm
    average_deviation_CA[num][3] += np.abs(averages_CA[num][3] - data_CA[len(data_CA)-1][4])      #n2
    average_deviation_CA[num][4] += np.abs(averages_CA[num][4] - data_CA[len(data_CA)-1][5])      #theta
    average_deviation_CA[num][5] += np.abs(averages_CA[num][5] - data_CA[len(data_CA)-1][6])      #dphi
    average_deviation_CA[num][6] += np.abs(averages_CA[num][6] - data_CA[len(data_CA)-1][7])      #dn
    count += 1
    if delete_bad_fits:
        if data_CA[i][-2] == 1:
            average_deviation_CA[num][1] -= np.abs(averages_CA[num][1] - data_CA[i][2])     #N*Sigma
            average_deviation_CA[num][2] -= np.abs(averages_CA[num][2] - data_CA[i][3])      #betta_therm
            average_deviation_CA[num][3] -= np.abs(averages_CA[num][3] - data_CA[i][4])     #n2
            average_deviation_CA[num][4] -= np.abs(averages_CA[num][4] - data_CA[i][5])      #theta
            average_deviation_CA[num][5] -= np.abs(averages_CA[num][5] - data_CA[i][6])      #dphi
            average_deviation_CA[num][6] -= np.abs(averages_CA[num][6] - data_CA[i][7])      #dn
            count -= 1
    if delete_undivided:
        if data_CA[i][-1] == 0:
            average_deviation_CA[num][1] -= np.abs(averages_CA[num][1] - data_CA[i][2])     #N*Sigma
            average_deviation_CA[num][2] -= np.abs(averages_CA[num][2] - data_CA[i][3])      #betta_therm
            average_deviation_CA[num][3] -= np.abs(averages_CA[num][3] - data_CA[i][4])     #n2
            average_deviation_CA[num][4] -= np.abs(averages_CA[num][4] - data_CA[i][5])      #theta
            average_deviation_CA[num][5] -= np.abs(averages_CA[num][5] - data_CA[i][6])      #dphi
            average_deviation_CA[num][6] -= np.abs(averages_CA[num][6] - data_CA[i][7])      #dn
            count -= 1
    average_deviation_CA[num][1] = average_deviation_CA[num][1]/count
    average_deviation_CA[num][2] = average_deviation_CA[num][2]/count
    average_deviation_CA[num][3] = average_deviation_CA[num][3]/count
    average_deviation_CA[num][4] = average_deviation_CA[num][4]/count
    average_deviation_CA[num][5] = average_deviation_CA[num][5]/count
    average_deviation_CA[num][6] = average_deviation_CA[num][6]/count
    ############## FIGURES #################

    x = np.arange(1,10)

    fig = plt.figure('fig')
    plt.errorbar(x, average_sigma[:, 18]*10**58, average_deviation[:, 18]*10**58,
        color = Farben.co[0], linestyle = '', marker = 's')
    plt.xlabel('Measurement Number')
    plt.ylabel('2PP cross section (GM)')
    #--- set figure properties ---
    fig.set_size_inches(fig_width, fig_high)
    mgr = plt.get_current_fig_manager()
    mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,fig_width*my_dpi,fig_high*my_dpi)
    #tight_layout()
    #subplots_adjust(top = 0.95, left = 0.15, right = 0.98, bottom = 0.1)
    plt.subplots_adjust(top = 0.95, left = 0.11, right = 0.97, bottom = 0.09, hspace = 0.26, wspace = 0.27)
    # distance between the subplots, legend
    fig.savefig('AOM_opening_time_average.png', dpi=my_dpi*5)
    plt.show()
    plt.close()
    average_deviation[2,10] = 0
    average_deviation[8,3] = 0
    average_deviation[2,16] = 0
    average_sigma[2,16] = -2
    fig = plt.figure('fig2')
    for i in range(18):
        for j in range(9):
            if average_deviation[j,i] == average_sigma[j,i]:
                #print(i, j, average_deviation[j,i], average_sigma[j,i])
                average_deviation[j,i] = 0

        plt.errorbar(x, average_sigma[:, i]*10**58, average_deviation[:, i]*10**58,
```

```
        color = Farben.co[i], linestyle = '', marker = 's', label = '%i mW'%(300+i*50))
#plt.errorbar(x, average_sigma_max4, average_deviation_max4,
    color = Farben.co[1], linestyle = '', marker = 's')
#plt.errorbar(x, average_sigma_first4, average_deviation_first4, color =
    Farben.co[2], linestyle = '', marker = 's')
#plt.yscale('log')
plt.legend(loc = 'upper right')
plt.axis([0.5, 11.5, 10, 10000])
plt.xlabel('Measurement Number')
plt.ylabel('2PP cross section (GM)')
#--- set figure properties ---
fig.set_size_inches(fig_width, fig_high)
mgr = plt.get_current_fig_manager()
mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,fig_width*my_dpi,fig_high*my_dpi)
#tight_layout()
#subplots_adjust(top = 0.95, left = 0.15, right = 0.98, bottom = 0.1)
plt.subplots_adjust(top = 0.95, left = 0.11, right = 0.97, bottom = 0.09, hspace = 0.26, wspace = 0.27)
# distance between the subplots, legend
fig.savefig('AOM_opening_time_power.png', dpi=my_dpi*5)
plt.show()
plt.close()

fig3 = plt.figure('fig3')
plt.errorbar(x, betta_plot[:, 0], betta_plot[:, 1],
    color = Farben.co[1], linestyle = '', marker = 's', label = 'betta OA')
plt.errorbar(x, averages_CA[:, 2], average_deviation_CA[:, 2],
    color = Farben.co[0], linestyle = '', marker = '^', label = 'betta CA')
#plt.yscale('log')
plt.legend(loc = 'upper left')
#plt.axis([0.5, 11.5, 10, 10000])
plt.xlabel('Measurement Number')
plt.ylabel('2PP cross section (GM)')
#--- set figure properties ---
fig3.set_size_inches(fig_width, fig_high)
mgr = plt.get_current_fig_manager()
mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,fig_width*my_dpi,fig_high*my_dpi)
#tight_layout()
#subplots_adjust(top = 0.95, left = 0.15, right = 0.98, bottom = 0.1)
plt.subplots_adjust(top = 0.95, left = 0.11, right = 0.97, bottom = 0.09, hspace = 0.26, wspace = 0.27)
# distance between the subplots, legend
fig3.savefig('betta_betta_th.png', dpi=my_dpi*5)
#fig3.savefig('betta_betta_th.eps', format='eps', dpi=my_dpi*5)
plt.show()
plt.close()

fig4 = plt.figure('fig4')
plt.plot(x, averages_CA[:, 2]/betta_plot[:, 0],
    color = Farben.co[1], linestyle = '', marker = 's')#, label = 'betta OA')
#plt.plot(x, betta_plot[:, 0]/averages_CA[:, 2],
    color = Farben.co[2], linestyle = '', marker = 's')
#plt.yscale('log')
plt.legend(loc = 'best')
#plt.axis([0.5, 11.5, 10, 10000])
plt.xlabel('Measurement Number')
plt.ylabel('2PP cross section (GM)')
#--- set figure properties ---
fig4.set_size_inches(fig_width, fig_high)
mgr = plt.get_current_fig_manager()
mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,fig_width*my_dpi,fig_high*my_dpi)
#tight_layout()
#subplots_adjust(top = 0.95, left = 0.15, right = 0.98, bottom = 0.1)
plt.subplots_adjust(top = 0.95, left = 0.11, right = 0.97, bottom = 0.09, hspace = 0.26, wspace = 0.27)
# distance between the subplots, legend
fig4.savefig('betta_by_betta_th.png', dpi=my_dpi*5)
#fig3.savefig('betta_by_betta_th.eps', format='eps', dpi=my_dpi*5)
plt.show()
plt.close()

all_sigmas_m9
fig5 = plt.figure('fig5')
x_var = np.ones(len(all_sigmas_m1))
t.plot(x_var, all_sigmas_m1, color = 'black', linestyle = '', marker = 's',
```

```
    label = f'1: {len(all_sigmas_m1)} measurements')
x_var = np.ones(len(all_sigmas_m2))
x_var *= 2
plt.plot(x_var, all_sigmas_m2, color = 'black', linestyle = '', marker = 's',
    label = f'2: {len(all_sigmas_m2)} measurements')
x_var = np.ones(len(all_sigmas_m3))
x_var *= 3
plt.plot(x_var, all_sigmas_m3, color = 'black', linestyle = '', marker = 's',
    label = f'3: {len(all_sigmas_m3)} measurements')
x_var = np.ones(len(all_sigmas_m4))
x_var *= 4
plt.plot(x_var, all_sigmas_m4, color = 'black', linestyle = '', marker = 's',
    label = f'4: {len(all_sigmas_m4)} measurements')
x_var = np.ones(len(all_sigmas_m5))
x_var *= 5
plt.plot(x_var, all_sigmas_m5, color = 'black', linestyle = '', marker = 's',
    label = f'5: {len(all_sigmas_m5)} measurements')
x_var = np.ones(len(all_sigmas_m6))
x_var *= 6
plt.plot(x_var, all_sigmas_m6, color = 'black', linestyle = '', marker = 's',
    label = f'6: {len(all_sigmas_m6)} measurements')
x_var = np.ones(len(all_sigmas_m7))
x_var *= 7
plt.plot(x_var, all_sigmas_m7, color = 'black', linestyle = '', marker = 's',
    label = f'7: {len(all_sigmas_m7)} measurements')
x_var = np.ones(len(all_sigmas_m8))
x_var *= 8
plt.plot(x_var, all_sigmas_m8, color = 'black', linestyle = '', marker = 's',
    label = f'8: {len(all_sigmas_m8)} measurements')
x_var = np.ones(len(all_sigmas_m9))
x_var *= 9
#plt.plot(x_var, all_sigmas_m9, color = Farben.co[1], linestyle = '', marker = 's',
    label = f'9: {len(all_sigmas_m9)} measurements')
start = 0
end = 0
for i in range(18):
    if not i == 0:
        start += m_per_power[0,i-1]
    end += m_per_power[0,i]
    end = int(end)
    start = int(start)
    x_var = np.ones(end-start)
    plt.plot(x_var, all_sigmas_m1[start:end], color = Farben.co[i], linestyle = '', marker = 's')
        #, label = f'1: {len(all_sigmas_m1)} measurements')
start = 0
end = 0
for i in range(18):
    if not i == 0:
        start += m_per_power[1,i-1]
    end += m_per_power[1,i]
    end = int(end)
    start = int(start)
    x_var = np.ones(end-start)
    x_var *= 2
    plt.plot(x_var, all_sigmas_m2[start:end], color = Farben.co[i], linestyle = '', marker = 's')
        #, label = f'1: {len(all_sigmas_m1)} measurements')
start = 0
end = 0
for i in range(18):
    if not i == 0:
        start += m_per_power[2,i-1]
    end += m_per_power[2,i]
    end = int(end)
    start = int(start)
    x_var = np.ones(end-start)
    x_var *= 3
    plt.plot(x_var, all_sigmas_m3[start:end], color = Farben.co[i], linestyle = '', marker = 's')
        #, label = f'1: {len(all_sigmas_m1)} measurements')
start = 0
end = 0
for i in range(18):
    if not i == 0:
```

```
            start += m_per_power[3,i-1]
        end += m_per_power[3,i]
        end = int(end)
        start = int(start)
        x_var = np.ones(end-start)
        x_var *= 4
        plt.plot(x_var, all_sigmas_m4[start:end], color = Farben.co[i], linestyle = '', marker = 's')
            #, label = f'1: {len(all_sigmas_m1)} measurements')
    start = 0
    end = 0
    for i in range(18):
        if not i == 0:
            start += m_per_power[4,i-1]
        end += m_per_power[4,i]
        end = int(end)
        start = int(start)
        x_var = np.ones(end-start)
        x_var *= 5
        plt.plot(x_var, all_sigmas_m5[start:end], color = Farben.co[i], linestyle = '', marker = 's')
            #, label = f'1: {len(all_sigmas_m1)} measurements')
    start = 0
    end = 0
    for i in range(18):
        if not i == 0:
            start += m_per_power[5,i-1]
        end += m_per_power[5,i]
        end = int(end)
        start = int(start)
        x_var = np.ones(end-start)
        x_var *= 6
        plt.plot(x_var, all_sigmas_m6[start:end], color = Farben.co[i], linestyle = '', marker = 's')
            #, label = f'1: {len(all_sigmas_m1)} measurements')
    start = 0
    end = 0
    for i in range(18):
        if not i == 0:
            start += m_per_power[6,i-1]
        end += m_per_power[6,i]
        end = int(end)
        start = int(start)
        x_var = np.ones(end-start)
        x_var *= 7
        plt.plot(x_var, all_sigmas_m7[start:end], color = Farben.co[i], linestyle = '', marker = 's')
            #, label = f'1: {len(all_sigmas_m1)} measurements')
    start = 0
    end = 0
    for i in range(18):
        if not i == 0:
            start += m_per_power[7,i-1]
        end += m_per_power[7,i]
        end = int(end)
        start = int(start)
        x_var = np.ones(end-start)
        x_var *= 8
        plt.plot(x_var, all_sigmas_m8[start:end], color = Farben.co[i], linestyle = '', marker = 's')
            #, label = f'1: {len(all_sigmas_m1)} measurements')

    #plt.yscale('log')
    plt.legend(loc = 'lower right')
    plt.axis([0.5, 10.5, 10, 550])
    plt.xlabel('Measurement Number')
    plt.ylabel('2PP cross section (GM)')
    #--- set figure properties ---
    fig5.set_size_inches(fig_width, fig_high)
    mgr = plt.get_current_fig_manager()
    mgr.window.setGeometry(cm2inch(2)*my_dpi,cm2inch(2)*my_dpi,fig_width*my_dpi,fig_high*my_dpi)
    #tight_layout()
    #subplots_adjust(top = 0.95, left = 0.15, right = 0.98, bottom = 0.1)
    plt.subplots_adjust(top = 0.95, left = 0.075, right = 0.97, bottom = 0.095, hspace = 0.26, wspace = 0.27)
    # distance between the subplots, legend
    fig5.savefig('Asigma_all.png', dpi=my_dpi*5)
    fig5.savefig('Asigma_all.eps', format='eps', dpi=my_dpi*5)
```

```
plt.show()
plt.close()


##############save data
fileName = 'Z_Scan_average_sigma.csv'
if os.path.exists(fileName) == False:
    createHeader = True
else:
    createHeader = False
with open(fileName, 'a',newline='') as fileObj:
    writer = csv.writer(fileObj)
    if createHeader:
        writer.writerow(['average_sigma', 'av_deviation', 'Number_of_measurements', '0.3_W_av_Sigma',
            'av_deviation', 'Number_of_measurements', '0.35_W_av_Sigma', 'av_deviation',
            'Number_of_measurements', '0.4_W_av_Sigma', 'av_deviation', 'Number_of_measurements',
            '0.45_W_av_Sigma', 'av_deviation','Number_of_measurements','0.5_W_av_Sigma',
            'av_deviation','Number_of_measurements','0.55_W_av_Sigma', 'av_deviation',
            'Number_of_measurements','0.6_W_av_Sigma', 'av_deviation','Number_of_measurements',
            '0.65_W_av_Sigma', 'av_deviation','Number_of_measurements','0.7_W_av_Sigma',
            'av_deviation','Number_of_measurements','0.75_W_av_Sigma', 'av_deviation',
            'Number_of_measurements','0.8_W_av_Sigma', 'av_deviation','Number_of_measurements',
            '0.85_W_av_Sigma', 'av_deviation','Number_of_measurements','0.9_W_av_Sigma',
            'av_deviation','Number_of_measurements','0.95_W_av_Sigma', 'av_deviation',
            'Number_of_measurements','1.0_W_av_Sigma', 'av_deviation','Number_of_measurements',
            '1.05_W_av_Sigma', 'av_deviation','Number_of_measurements','1.1_W_av_Sigma',
            'av_deviation', 'Number_of_measurements','1.15_W_av_Sigma', 'av_deviation',
            'Number_of_measurements','AOM_frequenzy_(Hz)', 'AOM_duty_(%)', 'Open_Chopper'])
    for i in range(9):
        writer.writerow([average_sigma[i,18], average_deviation[i, 18], measurements_number[i, 18],
                         average_sigma[i,0], average_deviation[i, 0],measurements_number[i, 0],
                         average_sigma[i,1], average_deviation[i, 1],measurements_number[i, 1],
                         average_sigma[i,2], average_deviation[i, 2],measurements_number[i, 2],
                         average_sigma[i,3], average_deviation[i, 3],measurements_number[i, 3],
                         average_sigma[i,4], average_deviation[i, 4],measurements_number[i, 4],
                         average_sigma[i,5], average_deviation[i, 5],measurements_number[i, 5],
                         average_sigma[i,6], average_deviation[i, 6],measurements_number[i, 6],
                         average_sigma[i,7], average_deviation[i, 7],measurements_number[i, 7],
                         average_sigma[i,8], average_deviation[i, 8],measurements_number[i, 8],
                         average_sigma[i,9], average_deviation[i, 9],measurements_number[i, 9],
                         average_sigma[i,10], average_deviation[i, 10],measurements_number[i, 10],
                         average_sigma[i,11], average_deviation[i, 11],measurements_number[i, 11],
                         average_sigma[i,12], average_deviation[i, 12],measurements_number[i, 12],
                         average_sigma[i,13], average_deviation[i, 13],measurements_number[i, 13],
                         average_sigma[i,14], average_deviation[i, 14],measurements_number[i, 14],
                         average_sigma[i,15], average_deviation[i, 15],measurements_number[i, 15],
                         average_sigma[i,16], average_deviation[i, 16],measurements_number[i, 16],
                         average_sigma[i,17], average_deviation[i, 17],measurements_number[i, 17],
                         open_chopper[i,0], open_chopper[i,1], open_chopper[i,2]])


fileName = 'Z_Scan_averages_OA_CA.csv'
if os.path.exists(fileName) == False:
    createHeader = True
else:
    createHeader = False
with open(fileName, 'a',newline='') as fileObj:
    writer = csv.writer(fileObj)
    if createHeader:
        writer.writerow(['OA_average_sigma', 'OA_av_deviation', 'OA_average_betta',
            'OA_average_Betta_deviation', 'Number_of_measurements', 'CA_Average_Betta_therm',
            'Average_Betta_therm_deviation', 'average_N*Sigma', 'average_n2', 'average_theta',
            'average_dphi', 'average_dn', 'CA_Number_of_Measurements', 'Power_(mW)',
            'AOM_frequenzy_(Hz)', 'AOM_duty_(%)', 'Open_Chopper'])
    for i in range(9):
        P = 600
        j = 6
        if i == 8:
            P = 100
            j = 2
        writer.writerow([average_sigma[i,j], average_deviation[i, j], betta_plot[i,0], betta_plot[i,1],
            measurements_number[i, j], averages_CA[i, 2], average_deviation_CA[i, 2], averages_CA[i,1],
```

```
            averages_CA[i,3], averages_CA[i,4], averages_CA[i,5], averages_CA[i,6],
            measurements_number_CA[i], P, open_chopper[i,0], open_chopper[i,1], open_chopper[i,2]])


fileName = 'Z_Scan_betta_OAvCA.csv'
if os.path.exists(fileName) == False:
    createHeader = True
else:
    createHeader = False
with open(fileName, 'a',newline='') as fileObj:
    writer = csv.writer(fileObj)
    if createHeader:
        writer.writerow(['Average_Betta', 'Average_Betta_deviation', 'Average_Betta_therm',
            'Average_Betta_therm_deviation', 'AOM_frequenzy_(Hz)', 'AOM_duty_(%)', 'Open_Chopper'])
    for i in range(9):
        writer.writerow([betta_plot[i,0], betta_plot[i,1], averages_CA[i, 2], average_deviation_CA[i, 2],
            open_chopper[i,0], open_chopper[i,1], open_chopper[i,2]])

#_____________________________
#Latex tabellen
#_____________________________________
latex_collumn_sigma=['\t{average_sigma_at\\_\SI{300}{mW}_[GM]}','\t{average_sigma_at\\_\SI{350}{mW}_[GM]}',
                     '\t{average_sigma_at\\_\SI{400}{mW}_[GM]}','\t{average_sigma_at\\_\SI{450}{mW}_[GM]}',
                     '\t{average_sigma_at\\_\SI{500}{mW}_[GM]}','\t{average_sigma_at\\_\SI{550}{mW}_[GM]}',
                     '\t{average_sigma_at\\_\SI{600}{mW}_[GM]}','\t{average_sigma_at\\_\SI{650}{mW}_[GM]}',
                     '\t{average_sigma_at\\_\SI{700}{mW}_[GM]}','\t{average_sigma_at\\_\SI{750}{mW}_[GM]}',
                     '\t{average_sigma_at\\_\SI{800}{mW}_[GM]}','\t{average_sigma_at\\_\SI{850}{mW}_[GM]}',
                     '\t{average_sigma_at\\_\SI{900}{mW}_[GM]}','\t{average_sigma_at\\_\SI{950}{mW}_[GM]}',
                     '\t{average_sigma_at\\_\SI{1000}{mW}_[GM]}','\t{average_sigma_at\\_\SI{1050}{mW}_[GM]}',
                     '\t{average_sigma\\_at_\SI{1.100}{mW}_[GM]}','\t{average_sigma\\_at_'
                     '\SI{1.1500}{mW}_[GM]}','\t{average_sigma\\_all_Powers_[GM]}']
latex_collum_dev=['\t{sigma_deviation_at\\_\SI{300}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{350}{mW}_[GM]}',
                  '\t{sigma_deviation_at\\_\SI{400}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{450}{mW}_[GM]}',
                  '\t{sigma_deviation_at\\_\SI{500}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{550}{mW}_[GM]}',
                  '\t{sigma_deviation_at\\_\SI{600}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{650}{mW}_[GM]}',
                  '\t{sigma_deviation_at\\_\SI{700}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{750}{mW}_[GM]}',
                  '\t{sigma_deviation_at\\_\SI{800}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{850}{mW}_[GM]}',
                  '\t{sigma_deviation_at\\_\SI{900}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{950}{mW}_[GM]}',
                  '\t{sigma_deviation_at\\_\SI{1000}{mW}_[GM]}','\t{sigma_deviation_at\\_\SI{1050}{mW}_[GM]}',
                  '\t{sigma_deviation\\_at_\SI{1.100}{mW}_[GM]}','\t{sigma_deviation\\_at'
                  '\SI{1150}{mW}_[GM]}','\t{sigma_deviation\\_all_Powers_[GM]}']
latex_collumn_count=['\t{Number_of_measurements\\_at_\SI{300}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{350}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{400}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{450}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{500}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{550}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{600}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{650}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{700}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{750}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{800}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{850}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{900}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{950}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{1000}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{1050}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{1100}{mW}_[-]}',
                     '\t{Number_of_measurements\\_at_\SI{1150}{mW}_[-]}',
                     '\t{Number_of_measurements\\_all_Powers_[GM]}']

fileName = 'Latex_Z_Scan_average_sigma.csv'
if os.path.exists(fileName) == False:
    createHeader = True
else:
    createHeader = False
with open(fileName, 'a',newline='') as fileObj:
    writer = csv.writer(fileObj, delimiter='&')
    if createHeader:
        writer.writerow(['\\begin{longtblr}['])
        writer.writerow(['\tcaption_=_{Long_Title},'])
        writer.writerow(['\tlabel_=_{tb:test},'])
```

```
    writer.writerow(['']{'])
    writer.writerow(['\tcolspec = {c|ccccccccc},'])
    writer.writerow(['\trowhead = 1,'])
    writer.writerow(['}'])
    writer.writerow(['\t\hline\hline'])
    writer.writerow(['\tMeasurement set', 1, 2, 3, 4, 5, 6, 7, 8, '9\\\ '])
    writer.writerow(['\t\hline'])
for i in range(-1, 18):
    latex_sigma = []
    latex_deviation = []
    latex_count = []
    if i == -1:
        i=18
    for j in range(9):
        latex_sigma.append(round(average_sigma[j,i]*10**58, 2))
        latex_deviation.append(round(average_deviation[j,i]*10**58, 2))
        latex_count.append(int(measurements_number[j,i]))
        if j == 8:
            latex_sigma[8] = f'{average_sigma[8,i]*10**58:.2f}\\\ '
            latex_deviation[8] = f'{average_deviation[8,i]*10**58:.2f}\\\ '
            latex_count[8] = f'{measurements_number[8,i]:.0f}\\\ '
        if measurements_number[j,i] <3:
            average_sigma[j,i] = 0.0
        if average_sigma[j,i] == 0.0:
            latex_sigma[j] = '-'
            latex_deviation[j] = '-'
            latex_count[j] = '-'
            if j == 8:
                latex_sigma[j] = '-\\\ '
                latex_deviation[j] = '-\\\ '
                latex_count[j] = '-\\\ '
    writer.writerow([latex_collumn_sigma[i], latex_sigma[0], latex_sigma[1], latex_sigma[2],
                     latex_sigma[3], latex_sigma[4], latex_sigma[5], latex_sigma[6],
                     latex_sigma[7], latex_sigma[8]])
    writer.writerow([latex_collumn_deviation[i], latex_deviation[0], latex_deviation[1],
                     latex_deviation[2], latex_deviation[3], latex_deviation[4],
                     latex_deviation[5], latex_deviation[6], latex_deviation[7],
                     latex_deviation[8]])
    writer.writerow([latex_collumn_count[i], latex_count[0], latex_count[1], latex_count[2],
                     latex_count[3], latex_count[4], latex_count[5], latex_count[6],
                     latex_count[7], latex_count[8]])
```

References

[1] M. Sheik-Bahae, A.A. Said, and E.W. Van Stryland. "High-sensitivity, single-beam n2 measurements". In: *Opt. Lett.* 14 (17) (Sept. 1989), pp. 955–957. DOI: 10.1364/OL.14.000955. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-14-17-955.

[2] M. Sheik-Bahae et al. "Sensitive measurement of optical nonlinearities using a single beam". In: *IEEE Journal of Quantum Electronics* 26 (4) (1990), pp. 760–769. DOI: 10.1109/3.53394.

[3] E.W. Van Stryland and M. Sheik-Bahae. "Z-scan measurements of optical nonlinearities". In: *Characterization techniques and tabulations for organic nonlinear materials* 18 (3) (1998), pp. 655–692.

[4] M.C. Fischer et al. "Simultaneous self-phase modulation and two-photon absorption measurement by a spectral homodyne Z-scan method". In: *Opt. Express* 16 (6) (Mar. 2008), pp. 4192–4205. DOI: 10.1364/OE.16.004192. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-16-6-4192.

[5] V. Singh and P. Aghamkar. "Z-scan: A simple technique for determination of third-order optical nonlinearity". In: *AIP Conference Proceedings* 1675 (1) (2015), p. 030095. DOI: 10.1063/1.4929311. eprint: https://aip.scitation.org/doi/pdf/10.1063/1.4929311. URL: https://aip.scitation.org/doi/abs/10.1063/1.4929311.

[6] A. Jabbar, S. Abbas, and M. Jaduaa. "Studying the Linear and Nonlinear Optical Properties by Using Z-Scan Technique for CdS Thin Films Prepared by CBD Technique". In: *Applied Physics Research* 10 (Apr. 2018). DOI: 10.5539/apr.v10n3p7.

[7] M. Falconieri. "Thermo-optical effects in Z -scan measurements using high-repetition-rate lasers". In: *Journal of Optics A: Pure and Applied Optics* 1 (6) (1999). DOI: 10.1063/1.1713919. eprint: https://iopscience.iop.org/article/10.1088/1464-4258/1/6/302/meta#references. URL: https://iopscience.iop.org/article/10.1088/1464-4258/1/6/302/meta#references.

[8] R. Miedzinski et al. "Z-scan measurements of the third-order optical nonlinearities and linear optical properties of 70TeO2 - 5MxOy - 10P2O5 - 10ZnO - 5PbF2 glasses doped with Er3+ ions modified by transition metals". In: *Optical Materials* 85 (2018), pp. 48–54. ISSN: 0925-3467. DOI: https://doi.org/10.1016/j.optmat.2018.08.033. URL: https://www.sciencedirect.com/science/article/pii/S0925346718305718.

[9] K.J. Schafer et al. "Two-photon absorption cross-sections of common photoinitiators". In: *Journal of Photochemistry and Photobiology A: Chemistry* 162 (2-3) (2004), pp. 497–502.

[10] R. Schwalm. "Photoinitiators and photopolymerization". In: *Encyclopedia of Materials: Science and Technology* (2001), pp. 6946–6951.

[11] H.W. Lee et al. "Ultrafast third-order optical nonlinearities of vertically-aligned ZnO nanorods". In: *Chemical Physics Letters* 447 (1) (2007), pp. 86–90. ISSN: 0009-2614. DOI: https://doi.org/10.1016/j.cplett.2007.08.086. URL: https://www.sciencedirect.com/science/article/pii/S0009261407011955.

[12] A. Fischer, C. Cremer, and E.H.K. Stelzer. "Fluorescence of coumarins and xanthenes after two-photon absorption with a pulsed titanium–sapphire laser". In: *Appl. Opt.* 34 (12) (Apr. 1995), pp. 1989–2003. DOI: 10.1364/AO.34.001989. URL: https://opg.optica.org/ao/abstract.cfm?URI=ao-34-12-1989.

[13] M.A. Albota, C. Xu, and W.W. Webb. "Two-photon fluorescence excitation cross sections of biomolecular probes from 690 to 960 nm". In: *Appl. Opt.* 37 (31) (Nov. 1998), pp. 7352–7356. DOI: 10.1364/AO.37.007352. URL: https://opg.optica.org/ao/abstract.cfm?URI=ao-37-31-7352.

[14] H. Rath et al. "Core-modified expanded porphyrins with large third-order nonlinear optical response". In: *Journal of the American Chemical Society* 127 (33) (2005), pp. 11608–11609.

[15] S. Das et al. "Zinc (II)-and copper (I)-mediated large two-photon absorption cross sections in a bis-cinnamaldiminato Schiff base". In: *Journal of the American Chemical Society* 128 (2) (2006), pp. 402–403.

[16] R. Misra et al. "Modified (22π) smaragdyrins with large two-photon absorption cross section: a structure function correlation". In: *Organic Letters* 8 (4) (2006), pp. 629–631.

[17] H. Rath et al. "Aromatic core modified decaphyrins with the largest two-photon absorption cross-sections: syntheses and characterization". In: *Organic Letters* 8 (11) (2006), pp. 2325–2328.

[18] A. Nag, A.K. De, and D. Goswami. "Two-photon cross-section measurements using an optical chopper: z-scan and two-photon fluorescence schemes". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 42 (6) (2009), p. 065103.

[19] A.K. De and D. Goswami. "Exploring the nature of photo-damage in two-photon excitation by fluorescence intensity modulation". In: *Journal of fluorescence* 19 (2009), pp. 381–386.

[20] A. Nag, S. Singh, and D. Goswami. "Structure property correlations in alcohols through two-photon absorption cross-section measurements". In: *Chemical physics letters* 430 (4-6) (2006), pp. 420–423.

[21] M. Göppert-Mayer. "Über elementarakte mit zwei quantensprüngen". In: *Annalen der Physik* 401 (3) (1931), pp. 273–294.

[22] W. Kaiser and C.G.B. Garrett. "Two-photon excitation in Ca F 2: Eu 2+". In: *Physical review letters* 7 (6) (1961), p. 229.

[23] A. Jablonski. "Efficiency of anti-Stokes fluorescence in dyes". In: *Nature* 131 (3319) (1933), pp. 839–840.

[24] B. Valeur and J.-C. Brochon. *New trends in fluorescence spectroscopy: applications to chemical and life sciences*. Vol. 1. Springer Science & Business Media, 2012.

[25] R. Šípoš and J. Šima. "Jablonski Diagram Revisited". In: *Revista Cubana de Física* 37 (2) (2020), pp. 125–130. ISSN: 2224-7939. URL: https://revistacubanadefisica.org/index.php/rcf/article/view/2020v37p125.

[26] B. Wardle. *Principles and applications of photochemistry*. John Wiley & Sons, 2009.

[27] R.C. Evans, P. Douglas, and H.D. Burrow. *Applied photochemistry*. Vol. 36. Springer, 2013.

[28] C. Greant et al. "Multiphoton lithography as a promising tool for biomedical applications". In: *Advanced Functional Materials* 33 (39) (2023), p. 2212641.

[29] H.H. Jaffé and A.L. Miller. "The fates of electronic excitation energy". In: *Journal of chemical education* 43 (9) (1966), p. 469.

[30] E.B. Priestley and A. Haug. "Phosphorescence spectrum of pure crystalline naphthalene". In: *The Journal of Chemical Physics* 49 (2) (1968), pp. 622–629.

[31] "radical (free radical)". In: (2019). DOI: doi:10.1351/goldbook.R05066. URL: https://doi.org/10.1351/goldbook.R05066.

[32] P.B. Armentrout and J. Simons. "Understanding heterolytic bond cleavage". In: *Journal of the American Chemical Society* 114 (22) (1992), pp. 8627–8633.

[33] J. Clayden, N. Greeves, and S. Warren. *Organic chemistry*. Oxford University Press, USA, 2012.

[34] J. Collins. "Two-photon absorption and applications to biological systems". In: *Biophotonics: Spectroscopy, Imaging, Sensing, and Manipulation*. Springer, 2010, pp. 261–285.

[35] A.K. Nguyen and R.J. Narayan. "Two-photon polymerization for biological applications". In: *Materials Today* 20 (6) (2017), pp. 314–322. ISSN: 1369-7021. DOI: https://doi.org/10.1016/j.mattod.2017.06.004. URL: https://www.sciencedirect.com/science/article/pii/S1369702117303917.

[36] J.M. Hales et al. "Two-photon absorption: concepts, molecular materials and applications". In: *The WSPC Reference on Organic Electronics: Organic Semiconductors*. Chap. CHAPTER 13, pp. 397–442. DOI: 10.1142/9789813148611_0013. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789813148611_0013. URL: https://www.worldscientific.com/doi/abs/10.1142/9789813148611_0013.

[37] J. Kerr. "XL. A new relation between electricity and light: Dielectrified media birefringent". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50 (332) (1875), pp. 337–348. DOI: 10.1080/14786447508641302. eprint: https://doi.org/10.1080/14786447508641302. URL: https://doi.org/10.1080/14786447508641302.

[38] R.H. Stolen and A. Ashkin. "Optical Kerr effect in glass waveguide". In: *Applied Physics Letters* 22 (6) (1973), pp. 294–296. DOI: 10.1063/1.1654644. eprint: https://doi.org/10.1063/1.1654644. URL: https://doi.org/10.1063/1.1654644.

[39] M. Sheik-Bahae, D.J. Hagan, and E.W. Van Stryland. "Dispersion and band-gap scaling of the electronic Kerr effect in solids associated with two-photon absorption". In: *Phys. Rev. Lett.* 65 (1 July 1990), pp. 96–99. DOI: 10.1103/PhysRevLett.65.96. URL: https://link.aps.org/doi/10.1103/PhysRevLett.65.96.

[40] W. Steiger. *Investigation of multi-photon processing parameters and materials*. eng. Wien: TU Wien, 2018.

[41] W.M. Steen. "Laser Sources and Applications". In: *Optics and Laser Technology* 5 (29) (1997), pp. ix–x.

[42] M. Pawlicki et al. "Two-Photon Absorption and the Design of Two-Photon Dyes". In: *Angewandte Chemie International Edition* 48 (18) (2009), pp. 3244–3266. DOI: https://doi.org/10.1002/anie.200805257. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.200805257. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.200805257.

[43] X. Huang, X. Wang, and Y. Zhao. "Study on a series of water-soluble photoinitiators for fabrication of 3D hydrogels by two-photon polymerization". In: *Dyes and Pigments* 141 (2017), pp. 413–419. ISSN: 0143-7208. DOI: https://doi.org/10.1016/j.dyepig.2017.02.040. URL: https://www.sciencedirect.com/science/article/pii/S0143720817300281.

[44] D. Cao et al. "Synthesis and blue-violet two-photon excited fluorescence of a new organoboron compound". In: *Journal of Molecular Structure* 874 (1-3) (2008), pp. 46–50.

[45] S. de Reguardati et al. "High-accuracy reference standards for two-photon absorption in the 680–1050 nm wavelength range". In: *Optics Express* 24 (8) (2016), pp. 9053–9066.

[46] A. Ajami et al. "Measurement of degenerate two-photon absorption spectra of a series of developed two-photon initiators using a dispersive white light continuum Z-scan". In: *Applied Physics Letters* 111 (7) (2017), p. 071901.

[47] M. Balu et al. "White-light continuum Z-scan technique for nonlinear materials characterization". In: *Optics Express* 12 (16) (2004), pp. 3820–3826.

[48] A. Gnoli, L. Razzari, and M. Righini. "Z-scan measurements using high repetition rate lasers: how to manage thermal effects". In: *Optics express* 13 (20) (2005), pp. 7976–7981.

[49] R. Miedzinski et al. "Z-scan measurements of the third-order optical nonlinearities and linear optical properties of $70TeO_2$ - $5M_xO_y$ - $10P_2O_5$ - 10ZnO - $5PbF_2$ glasses doped with Er^{3+} ions modified by transition metals". In: *Optical Materials* 85 (Nov. 2018), pp. 48–54. DOI: 10.1016/j.optmat.2018.08.033.

[50] L.C. Aamodt and J.C. Murphy. "Size considerations in the design of cells for photoacoustic spectroscopy. II. Pulsed excitation response". In: *Journal of Applied Physics* 49 (6) (1978), pp. 3036–3045. DOI: 10.1063/1.325318. eprint: https://doi.org/10.1063/1.325318. URL: https://doi.org/10.1063/1.325318.

[51] J. N. Hayes. "Thermal Blooming of Laser Beams in Fluids". In: *Appl. Opt.* 11 (2) (Feb. 1972), pp. 455–461. DOI: 10.1364/AO.11.000455. URL: http://opg.optica.org/ao/abstract.cfm?URI=ao-11-2-455.

[52] J.P. Gordon et al. "Long-Transient Effects in Lasers with Inserted Liquid Samples". In: *Journal of Applied Physics* 36 (1) (1965), pp. 3–8. DOI: 10.1063/1.1713919. eprint: https://doi.org/10.1063/1.1713919. URL: https://doi.org/10.1063/1.1713919.

[53] C. Hu and J.R. Whinnery. "New Thermooptical Measurement Method and a Comparison with Other Methods". In: *Appl. Opt.* 12 (1) (Jan. 1973), pp. 72–79. DOI: 10.1364/AO.12.000072. URL: http://opg.optica.org/ao/abstract.cfm?URI=ao-12-1-72.

[54] A.J. Twarowski and D.S. Kliger. "Multiphoton absorption spectra using thermal blooming. I. Theory". In: *Chemical Physics* 20 (2) (Mar. 1977), pp. 253–258. DOI: 10.1016/0301-0104(77)85029-5.

[55] I.S. Gradshteyn and I.M. Ryzhik. *Table of integrals, series, and products*. Seventh. Elsevier/Academic Press, Amsterdam, 2007, pp. xlviii+1171. ISBN: 978-0-12-373637-6.

[56] S. Wu and N.J. Dovichi. "Fresnel diffraction theory for steady-state thermal lens measurements in thin films". In: *Journal of Applied Physics* 67 (3) (1990), pp. 1170–1182. DOI: 10.1063/1.346057. eprint: https://doi.org/10.1063/1.346057. URL: https://doi.org/10.1063/1.346057.

[57] J.D. Gaskill. *Linear Systems, Fourier Transforms, and Optics*. eng. New York: J. Wiley & Sons, 1978, p. 576. ISBN: 978-0-471-29288-3.

[58] S.J. Sheldon, L.V. Knight, and J.M. Thorne. "Laser-induced thermal lens effect: a new theoretical model". In: *Appl. Opt.* 21 (9) (May 1982), pp. 1663–1669. DOI: 10.1364/AO.21.001663. URL: http://opg.optica.org/ao/abstract.cfm?URI=ao-21-9-1663.

[59] E.A. Donley et al. "Double-pass acousto-optic modulator system". In: *Review of Scientific Instruments* 76 (6) (2005), p. 063112.

[60] E.I. Gordon. "A review of acoustooptical deflection and modulation devices". In: *Applied optics* 5 (10) (1966), pp. 1629–1639.

[61] M.D. Fox. "Acousto-optics". In: *Proceedings of the IEEE* 68 (10) (1980), pp. 1357–1357.

[62] E.H. Young and S.-K. Yao. "Design considerations for acousto-optic devices". In: *Proceedings of the IEEE* 69 (1) (1981), pp. 54–64.

[63] V. Harinarayana and Y.C. Shin. "Two-photon lithography for three-dimensional fabrication in micro/nanoscale regime: A comprehensive review". In: *Optics & Laser Technology* 142 (2021), p. 107180. ISSN: 0030-3992. DOI: https://doi.org/10.1016/j.optlastec.2021.107180. URL: https://www.sciencedirect.com/science/article/pii/S0030399221002681.

[64] S. Kawata et al. "Finer features for functional microdevices". In: *Nature News* (). URL: https://www.nature.com/articles/35089130#citeas.

[65] M.T. Raimondi et al. "Two-Photon Laser Polymerization: From Fundamentals to Biomedical Application in Tissue Engineering and Regenerative Medicine". In: *Journal of Applied Biomaterials & Functional Materials* 10 (1) (2012), pp. 56–66. DOI: 10.5301/JABFM.2012.9278. URL: https://doi.org/10.5301/JABFM.2012.9278.

[66] W. Haske et al. "65 nm feature sizes using visible wavelength 3-D multiphoton lithography". In: *Opt. Express* 15 (6) (Mar. 2007), pp. 3426–3436. DOI: 10.1364/OE.15.003426. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-15-6-3426.

[67] J. Stampfl, R. Liska, and A. Ovsianikov. *Multiphoton lithography: Techniques, materials, and applications*. John Wiley & Sons, 2016.

[68] L. Amato et al. "Integrated three-dimensional filter separates nanoscale from microscale elements in a microfluidic chip". In: *Lab on a Chip* 12 (6) (2012), pp. 1135–1142.

[69] C. Schizas et al. "On the design and fabrication by two-photon polymerization of a readily assembled micro-valve". In: *The International Journal of Advanced Manufacturing Technology* 48 (5) (2010), pp. 435–441.

[70] P. Galajda and P. Ormos. "Rotation of microscopic propellers in laser tweezers". In: *Journal of Optics B: Quantum and Semiclassical Optics* 4 (2) (2002), S78.

[71] R.J. Narayan et al. "Medical prototyping using two photon polymerization". In: *Materials Today* 13 (12) (2010), pp. 42–48. ISSN: 1369-7021. DOI: https://doi.org/10.1016/S1369-7021(10)70223-6. URL: https://www.sciencedirect.com/science/article/pii/S1369702110702236.

[72] R.J. Young and P.A. Lovell. *Introduction to polymers*. CRC press, 2011.

[73] J.-P. Fouassier. *Photoinitiation, photopolymerization, and photocuring: fundamentals and applications*. Hanser, 1995.

[74] S.B. Allin. *Introduction to Physical Polymer Science, (Sperling, LH)*. 2001.

[75] X.-Z. Dong, Z.-S. Zhao, and X.-M. Duan. "Improving spatial resolution and reducing aspect ratio in multiphoton polymerization nanofabrication". In: *Applied Physics Letters* 92 (9) (2008), p. 091113.

[76] J.-F. Xing et al. "Improving spatial resolution of two-photon microfabrication by using photoinitiator with high initiating efficiency". In: *Applied physics letters* 90 (13) (2007), p. 131106.

[77] D. Tan et al. "Reduction in feature size of two-photon polymerization using SCR500". In: *Applied physics letters* 90 (7) (2007), p. 071106.

[78] T. Stichel et al. "Two-photon Polymerization as Method for the Fabrication of Large Scale Biomedical Scaffold Applications." In: *Journal of Laser Micro/Nanoengineering* 5 (3) (2010).

[79] B.E.A. Saleh and M.C. Teich. *Fundamentals of photonics; 2nd ed.* Wiley series in pure and applied optics. New York, NY: Wiley, 2007. URL: https://cds.cern.ch/record/1084451.

[80] T. O'Brian and S. Diddams. "What is the fastest event (shortest time duration) that can be measured with today's technology and how is this done?" en. In: (Dec. 2004). URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=30032.

[81] Newport-Corporation. *Application Note 27 Long Scan Autocorrelator*. 2015. URL: https://www.newport.com/medias/sys_master/images/images/h69/ha2/8797197205534/Long-Scan-Autocorrelator-App-Note-27.pdf.

[82] N.S. Makarov, M. Drobizhev, and A. Rebane. "Two-photon absorption standards in the 550–1600 nm excitation wavelength range". In: *Opt. Express* 16 (6) (Mar. 2008), pp. 4029–4047. DOI: 10.1364/OE.16.004029. URL: http://www.osapublishing.org/oe/abstract.cfm?URI=oe-16-6-4029.