



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

Fachbereich Ingenieurwissenschaften  
Studiengang Angewandte Mathematik  
Modul 9050: Bachelorarbeit

**Bachelorarbeit:**  
**Simulation der Strahlpropagation und Superposition  
teilkohärenter Wellen für den Einsatz in diffraktiven  
Systemen**

**Ulf-Vincent Sponholz**

**2020**

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>2</b>
1.1 Motivation . . . . .	2
1.2 Aufgabenstellung . . . . .	2
<b>2 Theoretische Grundlagen</b>	<b>3</b>
2.1 Beugung des Lichtes . . . . .	3
2.1.1 Huygen'sches Prinzip . . . . .	3
2.1.2 Beschreibung von Licht als Welle . . . . .	3
2.2 Herleitung des Beugungsintegrals . . . . .	5
2.3 Funktionsprinzip diffraktiver optischer Elemente . . . . .	7
2.3.1 Blazegitter . . . . .	8
2.3.2 Fanout-Gitter . . . . .	8
2.4 Fourier-Optik . . . . .	9
2.4.1 Fouriertransformation . . . . .	9
2.4.2 Eigenschaften der Fouriertransformation . . . . .	10
2.4.3 DFT-FFT . . . . .	11
2.4.4 Fraunhofer Beugung . . . . .	12
2.4.5 Verknüpfung . . . . .	14
2.5 Beschreibung "realen" Lichtes . . . . .	14
2.5.1 Kohärenz . . . . .	14
2.5.2 Leuchtdioden und Weißlicht . . . . .	19
2.5.3 Gaußstrahl . . . . .	20
<b>3 Konzeptionierung und Implementierung</b>	<b>22</b>
3.1 Vorüberlegungen . . . . .	22
3.1.1 Analyse der Ausgangssituation . . . . .	22
3.1.2 Wahl der Programmiersprache . . . . .	22
3.1.3 Beschreibung des Modells anhand des Programmablaufes . . . . .	23
3.2 Umsetzung . . . . .	27
3.2.1 GUI . . . . .	27
3.2.2 Ausgabeschema der Dateien . . . . .	31
3.2.3 Ausgewählte Funktionen und Parallelisierung . . . . .	32
3.2.4 Verbesserungspotential . . . . .	37
<b>4 Beispielberechnungen</b>	<b>38</b>
4.1 Einfluss der Kohärenzlänge . . . . .	39
4.2 Variation der Transmissionsfunktion . . . . .	46
4.3 Variation des Lichtspektrums . . . . .	48
<b>5 Fazit</b>	<b>50</b>
<b>6 Quellen</b>	<b>51</b>
<b>7 Abbildungsverzeichnis</b>	<b>52</b>

# 1 Einleitung

## 1.1 Motivation

Bereits im Jahre 1678 beschrieb Christiaan Huygens den Wellencharakter des Lichtes [3]. Dennoch dauerte es weitere 124 Jahre bis Thomas Young diesen experimentell durch Beugung am Doppelspalt bestätigte und die Welleneigenschaft des Lichtes kurz darauf wissenschaftlicher Konsens wurde [12].

Bis weit ins 20.Jhd hinein beruhten optische Bauteile fast ausschließlich auf dem Prinzip der Lichtbrechung (Refraktion), welches auch ohne Wellenmodell auf der Basis des Variationsprinzips erklärt werden kann: Licht nimmt den Weg, für den es die kürzeste Zeit benötigt (Prinzip von Fermat).

Erst neuere Entwicklungen verwenden gezielt Beugung (Diffraktion)[1], um Licht zu formen und zu lenken. In sogenannten diffraktiv optischen Elementen, ferner DOE genannt, erfolgt eine Segmentierung und Umleitung von Wellenfronten. Hierbei wird eine gewünschte Intensitäts- bzw. Phasenverteilung durch Einstellen der optischen Weglängen und der Superposition in Medien erreicht[6].

Nach heutigem Stand wird sich der Trend zu kompakteren Bauteilen weiter fortsetzen, sodass die Technik der DOE weiter in den Fokus der Industrie rückt.

Mathematische Beschreibungen auf der Basis von Wellengleichungen verwenden in den meisten Fällen nur eine zentrale Wellenlänge. Die in heutigen Beleuchtungssystemen verwendete, auf Halbleitertechnik basierende Leuchtdiode, lässt sich gut im Rahmen einer monochromatischen Beschreibung (d.h. eine Wellenlänge) erfassen. Allerdings ist im Bezug auf andere Leuchtkörper, Tageslicht und sogenannte Weißlicht-LED eine tiefergehende mathematische Betrachtung der Kohärenzlängen von Nöten. Die Kohärenzlänge ist hierbei ein Maß für die durchschnittliche Länge der von einer realen Lichtquelle ausgesendeten Wellenzüge. Der Trend zu dem Tageslicht nachempfundenen, homogenen Lichtquellen stellt aufgrund der geringen Kohärenzlänge eine besondere Herausforderung in der Simulation dar.

## 1.2 Aufgabenstellung

Ziel dieser Arbeit ist die mathematische Beschreibung und Programmierung der Strahlpropagation und Superposition teilkohärenter Wellen für den Einsatz in einem Projekt zur Erstellung diffraktiver Optiken. Im ersten Schritt soll das Programm in der Lage sein, Transmissionsfunktionen zur Beschreibung der Wirkung einer DOE sowie Lichtfelder als mathematische Funktionen anzugeben, diese in einem festzulegendem Intervall abzutasten und mittels Fouriertransformation in den Spektralbereich zu transformieren. Zudem sollen die Eigenschaften der Lichtquelle wie das Spektrum und die Strahleigenschaften berücksichtigt werden, sodass sich eine reale Lichtquelle abbilden lässt. Im Zuge einer späteren Erweiterbarkeit wird dies in Python 3 umgesetzt.

## 2 Theoretische Grundlagen

### 2.1 Beugung des Lichtes

#### 2.1.1 Huygen'sches Prinzip

Der niederländische Gelehrte Christiaan Huygens schlug im Jahre 1678 eine neue Theorie zur Ausbreitung von Licht vor, welche die zuvor von anderen Wissenschaftlern entdeckte Eigenschaft der Beugung an einem Spalt zu erklären suchte. Nach seiner 1690 veröffentlichten Arbeit "Tractatus de lumine"<sup>[3]</sup> ist jeder Punkt einer Wellenfront der Ausgangspunkt einer kugelförmigen Elementarwelle. Jede dieser Elementarwellen breite sich demnach mit der gleichen Geschwindigkeit aus wie die ursprünglich betrachtete Welle, woraus sich die ausbreitende Wellenfront durch die Einhüllenden der Elementarwellen ergibt. Fälschlicherweise nahm Huygens jedoch an, dass diese Elementarwellen lediglich in Ausbreitungsrichtung wirken. Fresnel erweiterte dies später mithilfe des Superpositionsprinzips und erklärte dadurch auch die scheinbare einseitige Ausbreitung.

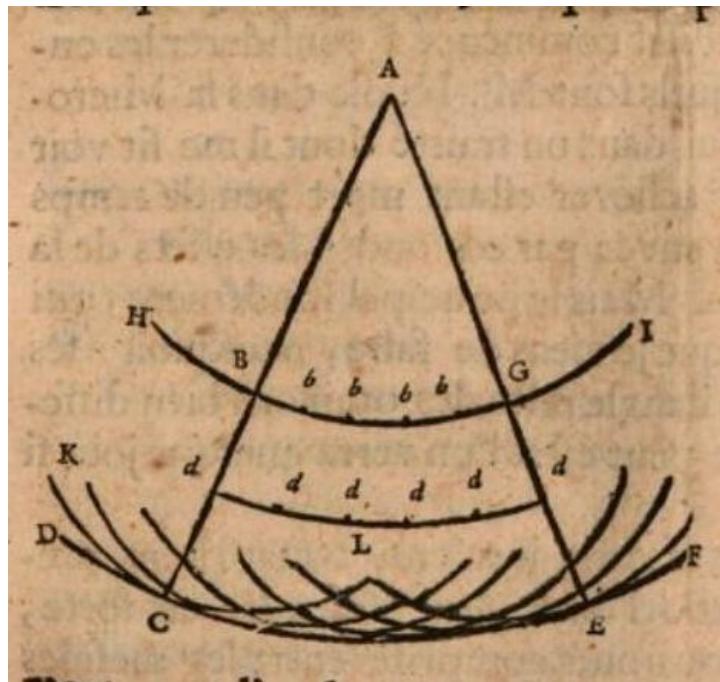


Abbildung 1: Beschreibung der Wellenfronten in Huygens' Originalmanuskript

#### 2.1.2 Beschreibung von Licht als Welle

Zur Herleitung der Wellengleichung des Lichtes sei zunächst von den nach dem schottischen Physiker James Clark Maxwell (1831-1879) benannten und von ebenjenem eingeführten Maxwell-Gleichungen, in der mikroskopischen Ausführung, ausgegangen

- *Gaußsches Gesetz, Ladungen sind die Quellen des el. Feldes und die Feldlinien divergieren unter Anwesenheit einer Ladung*

mit der Ladungsdichte  $\rho$  und der elektrischen Feldkonstante  $\epsilon_0$  gilt

$$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0}$$

- *Gaußsches Gesetz für Magnetfelder, Quellenfreiheit des magnetischen Feldes, keine magnetischen Monopole, Magnetfeldlinien sind geschlossen*

$$\vec{\nabla} \cdot \vec{B} = 0$$

- *Induktionsgesetz, Änderungen eines magnetischen Feldes erzeugen Wirbelströme*

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

- *erweitertes Durchflutungsgesetz, elektrische Ströme induzieren magnetisches Wirbelfeld* mit der elektrischen Stromdichte  $\vec{j}$ , der elektrischen  $\epsilon_0$ , bzw. magnetischen Feldkonstante  $\mu_0$  gilt

$$\vec{\nabla} \times \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t}$$

Wird nun die Ausbreitung in Abwesenheit von Ladungen (und damit Strömen) betrachtet, so lässt sich die elektrische Feldstärke anhand der Maxwell Gleichungen wie folgt auf eine partielle Differentialgleichung zurückführen [7]. Zunächst wird auf das Induktionsgesetz die Rotation angewandt, es ergibt sich

$$rot(rot(\vec{E})) = rot(-\frac{\partial \vec{B}}{\partial t})$$

Ferner ergibt sich durch Ableitung des Durchflutungsgesetzes nach der Zeit

$$rot(\frac{\partial \vec{B}}{\partial t}) = \mu_0 \epsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2}$$

Es folgt

$$rot(rot(\vec{E})) = -\mu_0 \epsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2}$$

Die Rotation der Rotation des elektrischen Feldes lässt sich folgendermaßen ausdrücken

$$rot(rot(\vec{E})) = -div(grad(\vec{E})) + grad(div(\vec{E}))$$

wobei nach dem Gaußschen Gesetz in Abwesenheit einer Ladung die Divergenz des elektrischen Feldes 0 ist. Mit der Definition des Laplace Operators im  $\mathbb{R}^3$

$$\Delta f := div(grad(f)) \stackrel{\text{kart.}}{=} \sum_{k=1}^3 \frac{\partial^2 f}{\partial x_k^2}$$

ergibt sich die partielle Differentialgleichung des elektrischen Feldes (in kartesischen Koordinaten) zu

$$(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}) \vec{E}(r, t) = \mu_0 \epsilon_0 \frac{\partial^2 \vec{E}(r, t)}{\partial t^2}$$

Dies lässt sich als die d'Alembert'sche Gleichung für Wellen identifizieren

$$\frac{1}{c^2} \frac{\partial^2 \vec{E}(r, t)}{\partial t^2} - \Delta \vec{E}(r, t) = 0$$

wobei es sich bei  $c$  nach d'Alembert um die Ausbreitungsgeschwindigkeit der Welle im betrachteten Medium handelt. Hier ergibt sich nun die Ausbreitungsgeschwindigkeit des Lichtes im Vakuum aus der elektrischen  $\epsilon_0$  und der magnetischen Feldkonstante  $\mu_0$

$$c_0 = \frac{1}{\sqrt{\mu_0 \epsilon_0}}$$

In anderen Medien hat das Licht die (geringere) Geschwindigkeit

$$c_{medium} = \frac{1}{\sqrt{\mu\epsilon}} = \frac{1}{\sqrt{\mu_0 \mu_r \epsilon_0 \epsilon_r}}$$

Dabei handelt es sich bei der relativen Permeabilität  $\mu_r$  respektive Permittivität  $\epsilon_r$  um frequenzabhängige Materialkonstanten.

Aus der Theorie über die Lösungen partieller Differentialgleichungen ergibt sich nun, dass sich jede Lösung als Linearkombination harmonischer Wellen in komplexer Darstellung schreiben lässt. So stellt die ebene Welle mit Ausbreitungsrichtung z

$$A(z, t) = A_0 \cdot \sin \left[ \omega \left( \frac{z}{c} - t \right) \right]$$

eine Lösung der partiellen Differentialgleichung dar, bei der die Amplitude an jedem Punkt in einer beliebigen (x,y)-Ebene gleich ist.

Insbesondere im Hinblick auf eine punktförmige Lichtquelle, bei welcher die Intensität und damit die Leistung pro Fläche gemäß der Definition der Intensität[10]

$$I := \frac{1}{2} c \epsilon |E|^2$$

quadratisch mit dem Abstand  $r$  zur Quelle abnimmt, folgt aus der Proportionalität der Intensität zum Quadrat der Feldstärke, dass diese mit dem Faktor  $\frac{1}{r}$  abnimmt [4].

Unter diesen Randbedingungen ergibt sich somit, für eine Punktquelle, die harmonische Welle mit abnehmender Feldstärke als eine weitere Lösung der d'Alembertschen Wellengleichung

$$E(r, t) = \frac{E_0}{r} e^{i(\omega t - kr)}$$

Dabei handelt es sich bei  $k$  um die Kreiswellenzahl  $\frac{2\pi}{\lambda}$  und bei der Amplitude  $E_0$  anders als bei ebenen Wellen um eine komplexe Zahl, da nun auch eine Phase in dieser kodiert sein muss.

## 2.2 Herleitung des Beugungsintegrals

Da sich nach dem Huygen'schen Prinzip jeder Punkt einer Wellenfront als Ausgangspunkt einer Elementarwelle interpretieren lässt, sei nun ein Schirm derart gegeben, dass eine Öffnung genau eine Teilwelle isoliert. (Siehe Abbildung)

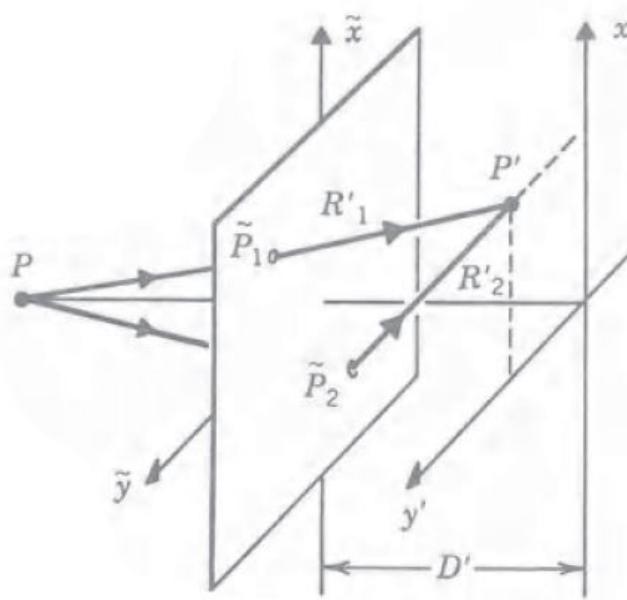


Abbildung 2: Überlagerung zweier von Lochblenden ausgehenden Elementarwellen,  
Young'sches Experiment[2]

Vereinfacht sei nun eine Punktquelle monochromatischen Lichts gegeben. Dadurch lässt sich das elektrische Feld als Kugelwelle folgendermaßen angeben:

$$E(R, t) = \frac{E_0}{R} e^{i(\omega t - kR)}$$

Wobei es sich bei  $R$  jeweils um den Abstand des zu betrachtenden Punktes vom Ausgangspunkt der betrachteten Elementarwelle handelt.

Das Feld hinter dem Schirm ist nun gegeben durch:

$$E_{Res} = CE_0 e^{i\omega t} \frac{e^{-ik(R+R')}}{RR'} \Delta\sigma$$

Bei  $\Delta\sigma$  handelt es sich um das Flächenelement der Öffnung. Unter der Voraussetzung, dass dieses klein genug ist, um eine einzelne Punktwelle zu isolieren, besteht eine Proportionalität. Die komplexe Zahl  $C$  röhrt aus der Notwendigkeit unterschiedlicher Phasen der Primär- und Sekundärwelle und wird an dieser Stelle nicht weiter erläutert.

Seien nun zwei derartige Öffnungen an den Punkten  $P_1$  und  $P_2$  gegeben, so ergibt sich das resultierende Feld als Summe der Huygen'schen Sekundärwellen:

$$E = C \left[ E_1 \frac{e^{-ikR_1}}{R_1} + E_2 \frac{e^{-ikR_2}}{R_2} \right] \Delta\sigma$$

Erweitert auf  $N$  Lochblenden ergibt sich:

$$E = C \sum_{n=1}^N E_n \frac{e^{-ikR_n}}{R_n} \Delta\sigma_n$$

Bei  $E_n, R_n$  und  $\sigma_n$  handelt es sich dabei um die Eigenschaften der  $n$ -ten Lochblende. Da es in der

Realität nicht machbar ist, eine Lochblende derart zu gestalten, dass lediglich eine Elementarwelle durch diese entsteht, muss die Theorie auf Lochblenden mit einem real existentierendem Durchmesser erweitert werden.

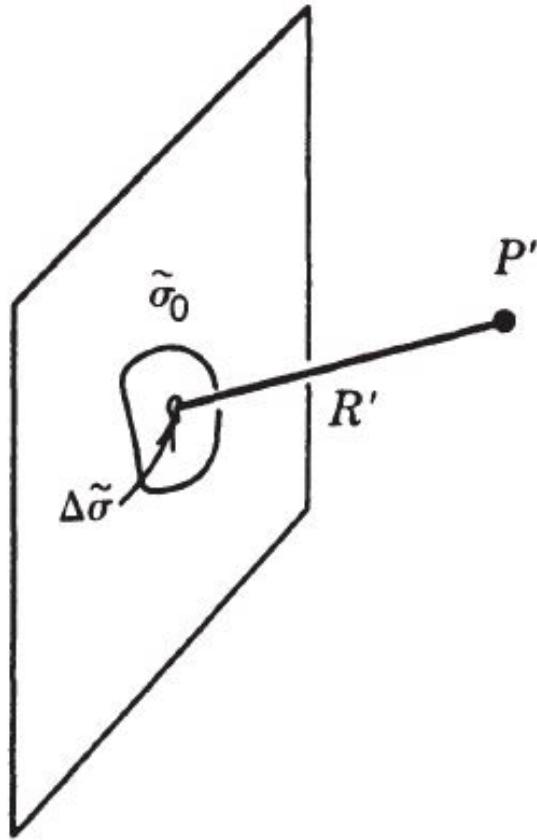


Abbildung 3: Eine reale Blende kann als Kombination infinitesimaler Lochblenden interpretiert werden[2]

Im Grenzfall einer dichten Anordnung unendlich vieler Lochblenden infinitesimaler Durchmesser, welche in ihrer Gesamtheit eine kontinuierliche Öffnung  $\sigma_0$  bilden, ergibt sich das Feld über ein Flächenintegral:

$$E = C \iint_{\sigma_0} E \frac{e^{-ikR}}{R} \Delta\sigma$$

### 2.3 Funktionsprinzip diffraktiver optischer Elemente

Ein diffraktives optisches Element basiert, anders als refraktive optische Elemente, welche auf dem Prinzip der Lichtbrechung aufbauen, auf der Beugung des Lichtes. Gegenüber refraktiven optischen Elementen haben diese einige Vor- und Nachteile. Da diffraktiv optische Elemente vergleichsweise schwer zu realisieren sind, basieren die meisten optischen Aufbauten sowohl historisch als auch aktuell nicht auf diesen. Im Zuge der Miniaturisierung bieten DOE jedoch deutliche Vorteile wie kompaktere Bauweisen und Gewichtsersparnis, so dass sie, zunächst in Spezialanwendungen, immer häufiger zum Einsatz kommen. Die Strahlteilung und Strahlformung stellt einen typischen Anwendungsfall diffraktiver Optiken dar. Kombiniert man diese mit herkömmlichen Optiken zu Hybrid-Elementen, so bieten diese auch die Möglichkeit Farb-

und Abbildungsfehler zu korrigieren [6]. Da es sich bei dieser Arbeit um eine mathematische Betrachtung handelt, sei die Beschreibung der Herstellungsmöglichkeiten diffraktiver Optiken nur kurz angerissen. Dem interessierten Leser sei die einschlägige Literatur nahegelegt.

### 2.3.1 Blazegitter

Das sogenannte Blazegitter ist ein anschauliches Beispiel für die Kompaktheit diffraktiver Optiken im Vergleich zu einer analogen refraktiven Optik. Auf der linken Seite ist ein auf Lichtbrechung basierendes Prisma zu sehen, welches das Licht einer bestimmten Wellenlänge auf die erste Ordnung lenkt, auf der rechten Seite ein analoges Blazegitter, welches denselben Effekt durch Beugung erzeugt.

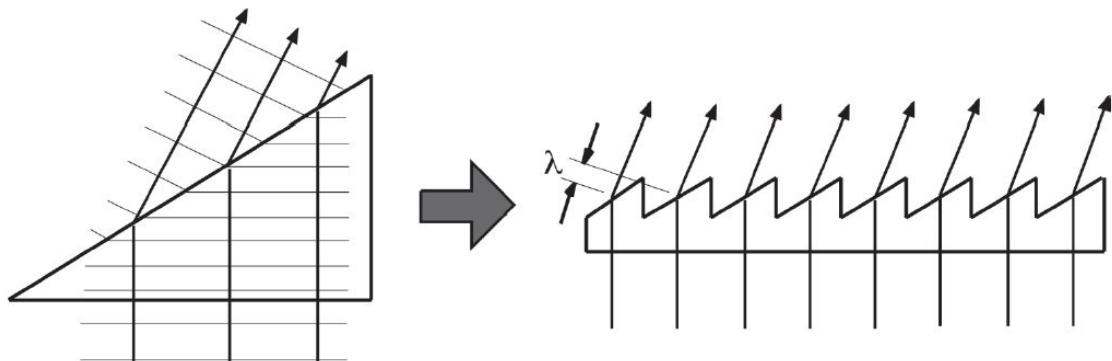


Abbildung 4: Lenkung von Wellenfronten im Prisma (refraktiv) und im Blazegitter (diffraktiv) [6]

Die Beugung an einem Blazegitter wird durch die Blazegleichung beschrieben

$$m\lambda = D \sin(\Theta_m)$$

Wobei es sich bei  $m$  um die Ordnung handelt, auf welche das Licht gelenkt wird, bei  $\Theta_m$  um den Ablenkungswinkel dieser Ordnung und bei  $D$  um die Periodenlänge des Blazegitters. Aufgrund dieser Wellenlängenabhängigkeit wird Weißlicht wie bei der Lichtbrechung an einem Prisma in sein Farbspektrum aufgespalten [6].

### 2.3.2 Fanout-Gitter

Ein weiteres diffraktiv optisches Element stellt das sogenannte Fanout-Gitter dar, welches zur Strahlteilung eingesetzt wird [6]. Anhand der Abbildung ist leicht zu erkennen, dass dieses Element sehr flach und damit raumsparend ist.



Abbildung 5: Das Dammann Gitter teilt den Strahl, es entstehen fünf Strahlen gleicher Intensität und mehrere Strahlen kleinerer Intensitäten [6]

Bei dem hier dargestellten DOE handelt es sich um ein sogenanntes Dammann-Gitter, welches einen Lichtstrahl in fünf Strahlen gleicher Intensität, sowie mehrere Strahlen geringerer Intensität splittet. Dieses Gitter lässt sich zudem weiter verbessern, um die Strahlen kleiner Intensitäten verschwinden zu lassen.

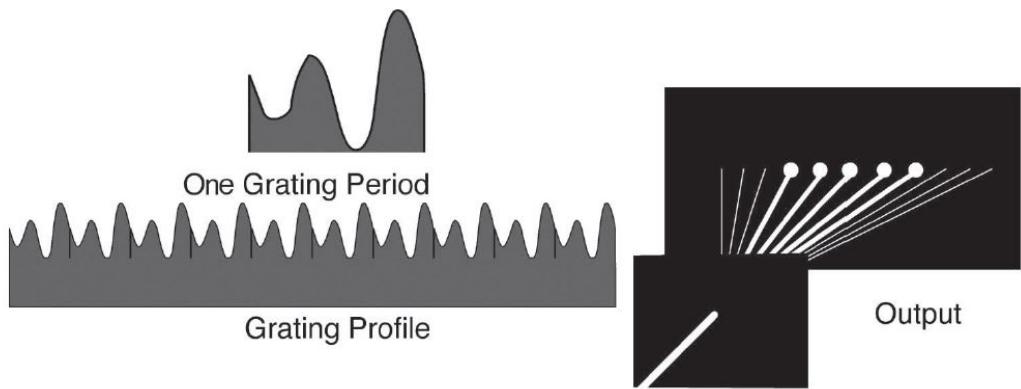


Abbildung 6: Dieses Dammann Gitter ist darauf optimiert die Effizienz zu erhöhen in dem die Sekundären Strahlen unterdrückt werden [6]

## 2.4 Fourier-Optik

### 2.4.1 Fouriertransformation

Die nach dem franz. Mathematiker und Physiker Jean Baptiste Joseph Fourier(1768-1830) benannte Fourier-Transformation ist eine Integraltransformation, welche das Eingangssignal vom Zeit- in den Frequenzbereich transformiert. Sie findet unter anderem in der Nachrichtentechnik (Spektralanalyse), der Physik (Schwingungsanalyse) und auch der Bildverarbeitung und Statistik (Glättung von Messdaten) eine Anwendung.

Jeder Originalfunktion  $f(t)$  ordnet die Fouriertransformation  $\mathcal{F}$  eine Bildfunktion  $F(\omega)$  zu

$$F(\omega) := \mathcal{F}(f(t)) := \int_{-\infty}^{\infty} e^{-2\pi i \omega t} f(t) dt$$

Oft wird das Paar  $(f(t), F(\omega))$  auch als sog. Fourier-Korrespondenz geschrieben.

$$f(t) \circ F(\omega)$$

Analog dazu ist die inverse Fourier-Transformation eine Rücktransformation vom Frequenzbereich in den Zeitbereich.

$$f(t) := \mathcal{F}^{-1}\{\mathcal{F}(\omega)\} := \int_{-\infty}^{\infty} e^{2\pi i \omega t} F(\omega) d\omega$$

$$F(\omega) \longrightarrow f(t)$$

Höherdimensionale Fourier-Transformationen werden durch Interpretation von  $\omega$  und  $t$  als Vektoren sowie deren Produkt im Exponenten als Skalarprodukt berechnet. Als Beispiel sei hier die örtliche Fouriertransformation für zwei Koordinaten gegeben

$$F(\xi_1, \xi_2) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-2\pi i(x_1 \xi_1 + x_2 \xi_2)} f(x_1, x_2) dx_1 dx_2$$

Dementsprechend ist die Inverse Fouriertransformation in zwei Dimensionen gegeben als:

$$f(x_1, x_2) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{2\pi i(x_1 \xi_1 + x_2 \xi_2)} F(\xi_1, \xi_2) d\xi_1 d\xi_2$$

An dieser Stelle sei angemerkt, dass hier die Konvention ohne Vorfaktoren verwendet wurde, welche vor allem in der Signalverarbeitung geläufig ist. In der Literatur finden sich je nach Anwendungsgebiet unterschiedliche Definitionen der Fouriertransformation.

## 2.4.2 Eigenschaften der Fouriertransformation

In der Anwendung ist es oft von Vorteil, eine Funktion in den Zeit- oder Frequenzbereich zu transformieren, um bestimmte Eigenschaften auszunutzen. Im Folgenden sind einige nützliche Eigenschaften aufgelistet.

### Linearität:

*Für die Originalfunktionen  $f(t), g(t)$ , die korrespondierenden Bildfunktionen  $F(\omega), G(\omega)$  und komplexe Zahlen  $\lambda, \mu$  gilt*

$$\mathcal{F}\{\lambda f(t) + \mu g(t)\} = \lambda F(\omega) + \mu G(\omega)$$

### Faltungssatz:

*Das Faltungsprodukt zweier Originalfunktionen korrespondiert mit dem gewöhnlichen Produkt der Bildfunktionen und umgekehrt*

$$\mathcal{F}\{(f * g)(t)\} = F(\omega) \cdot G(\omega)$$

### Verschiebungssatz:

*Hat die Originalfunktion  $f(t)$  eine Fouriertransformierte  $F(\omega)$ , so ergibt sich die verschobene Funktion  $f(t - T)$  zu*

$$\mathcal{F}\{f(t - T)\} = e^{-i\omega T} F(\omega)$$

### Ähnlichkeitssatz

*Hat die Originalfunktion  $f(t)$  eine Fouriertransformierte  $F(\omega)$ , so gilt für  $a > 0$  die ähnliche Funktion  $g(t) = f(at)$*

$$\mathcal{F}\{f(at)\} = \frac{1}{a} F\left(\frac{\omega}{a}\right)$$

### Differentiationssatz:

Hat die  $k$ -te Ableitung der Originalfunktion  $f(t)$  eine Fouriertransformierte, so lässt sich diese mithilfe der Fouriertransformierten der Originalfunktion angeben

$$\mathcal{F}\{f^{(k)}(t)\} = i^k \omega^k F(\omega)$$

**Multiplikationssatz:** Die Fouriertransformierte der mit einer natürlichen Potenz  $t^k$  multiplizierten Originalfunktion  $f(t)$  ergibt sich zu

$$\mathcal{F}\{t^k f(t)\} = i^k F^{(k)}(\omega)$$

### 2.4.3 DFT-FFT

Da die klassische Fourier-Transformation stückweise stetige Funktionen voraussetzt und damit kontinuierliche Eingangssignale, bietet sich diese für analoge Schaltkreise an. Bei digitaler Datenverarbeitung liegen jedoch diskrete Werte vor, damit ist für die computergestützte Verarbeitung eine Anpassung der Theorie nötig.

Die **Diskrete-Fourier-Transformation (DFT)** ist eine Abwandlung der Fourier-Transformation, welche statt stückweise stetiger Funktionen als äquidistant (im Sinne zeitlich und/oder örtlich) abgetastete Wertepaare einer solchen betrachtet werden.

$$x_k = \begin{pmatrix} \text{Zeit/Ort}(k) \\ \text{Messwert}(k) \end{pmatrix}$$

Sie ist folgendermaßen definiert:

$$X_s := \frac{1}{N} \sum_{v=0}^{N-1} x(v) \omega^{sv}$$

mit den  $N$ -ten komplexen Einheitswurzeln

$$\omega := e^{-\frac{i2\pi}{N}}$$

Die schnelle **Fourier-Transformation (FFT)** ist eine performanceorientierte Abwandlung der DFT. Diese verringert den Rechenaufwand bei  $N$  Messwerten von  $N^2$  auf  $N \log_2(N)$ , allerdings muss hierfür die Anzahl der Messwerte eine Zweierpotenz sein, um die Symmetrieeigenschaften der komplexen Einheitswurzeln auszunutzen.

Zunächst wird die Signal-Folge  $x(n)$  in zwei Teilstufen gespalten:

$$x_1(n) = x(2n)$$

$$x_2(n) = x(2n + 1)$$

Für diese Folge lautet die DFT wie folgt:

$$x_1(k) = \frac{1}{N} \sum_{n=0}^{N/2-1} x_1(n) e^{-ik2n(2\pi/N)}$$

$$x_2(k) = \frac{1}{N} \sum_{n=0}^{N/2-1} x_2(n) e^{-ik(2n+1)(2\pi/N)}$$

Wir summieren beide Teilstufen:

$$\begin{aligned}
X(k) &= \frac{1}{N} \sum_{n=0}^{N/2-1} [x_1(n)e^{-ik2n(2\pi/N)} + x_2(n)e^{-ik(2n+1)(2\pi/N)}] \\
&= \frac{1}{N} \sum_{n=0}^{N/2-1} [x_1(n)e^{-ik2n(2\pi/N)} + x_2(n)e^{-ik((4\pi n/N)(2\pi/N))}] \\
&= \frac{1}{N} \sum_{n=0}^{N/2-1} [x_1(n)e^{-ik2n(2\pi/N)} + x_2(n)e^{-ik((4\pi/N)n)e^{-ik(2\pi/N)}}] \\
&= \frac{1}{N} \sum_{n=0}^{N/2-1} x_1(n)e^{-ik2n(2\pi/N)} + e^{-ik(2\pi/N)} \sum_{n=0}^{N/2-1} x_2(n)e^{-ik((4\pi/N)n)}
\end{aligned}$$

für  $k = 0, 1, 2 \dots N - 1$

Nun werden die komplexen Einheitswurzeln substituiert. Aus der Gruppentheorie folgt:

$$w^{k+\frac{N}{2}} = -w^k$$

$$X(k) = X_1(k) + w^k X_2(k) \text{ für } 0 \leq k \leq \frac{1}{N}$$

$w$  steht in diesem Fall für die komplexen Einheitswurzeln.

Durch diese Gleichung können alle Punkte von 0 bis  $\frac{N}{2} - 1$  berechnet werden.

Dementsprechend sind nur noch die Punkte von  $\frac{N}{2}$  zu berechnen. Da das Ursprungssignal periodisch war, ergibt sich dafür folgende Formel:

$$X(k + \frac{N}{2}) = X_1(k) - w^k X_2(k) \text{ für } 0 \leq k \leq \frac{1}{N}$$

#### 2.4.4 Fraunhofer Beugung

Die Fraunhofer Beugung ist eine nach dem deutschen Optiker Joseph von Fraunhofer (1787-1826) benannte Fernfeldnäherung des Beugungsintegrals. Unter den Voraussetzungen kleiner Blendenöffnungen sowie ausreichend großer Entfernung von Quelle und Schirm (Fernfeld), lässt sich die optische Weglänge linear approximieren. Ausgehend vom Beugungsintegral der von einer Punktquelle bestrahlten Blende, erweitert durch eine Transmissionsfunktion (auch Blendenfunktion genannt),  $\tau$

$$E = CE_0 e^{i\omega t} \iint_{\sigma_0} \tau E \frac{e^{-ik(R+R')}}{RR'} \Delta\sigma$$

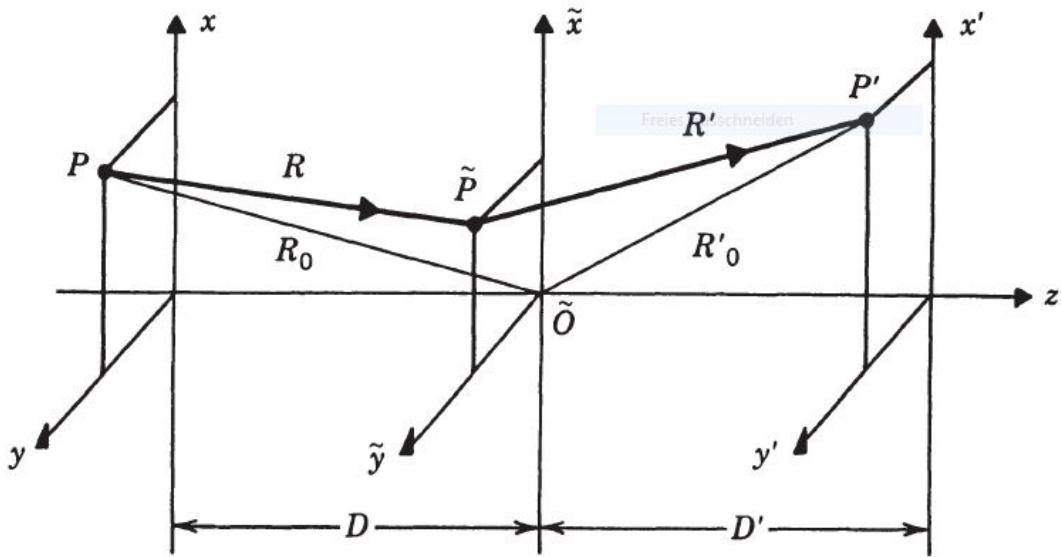


Abbildung 7: Illustration der Beugung eines Lichtstrahles ausgehend von  $P$ , an einem Punkt  $\tilde{P}$  in der Blendenebene [2]

Werden nun folgende Kleinwinkelnäherungen des Kosinus vorgenommen:

$$R \approx R_0 - \frac{x}{R_0} \tilde{x} - \frac{y}{R_0} \tilde{y}$$

$$R' \approx R'_0 - \frac{x'}{R'_0} \tilde{x} - \frac{y'}{R'_0} \tilde{y}$$

Da der Faktor  $RR'$  nicht im Exponenten vorkommt, kann dieser zudem weiter genähert werden:

$$RR' \approx R_0 R'_0$$

Durch die Variablentransformationen

$$u \equiv -\left(\frac{x}{R_0} + \frac{x'}{R'_0}\right) \frac{1}{\lambda}$$

$$v \equiv -\left(\frac{y}{R_0} + \frac{y'}{R'_0}\right) \frac{1}{\lambda}$$

folgt

$$R + R' = (R_0 + R'_0) + (u\tilde{x} + v\tilde{y})\lambda$$

wodurch sich mit dem koordinatenunabhängigen Faktor

$$\phi_0 = \omega t - k(R_0 + R'_0)$$

die Phasenfaktoren folgendermaßen umschreiben lassen.

$$e^{i\omega t} e^{-ik(R+R')} = e^{i\phi_0} e^{-i2\pi(u\tilde{x} + v\tilde{y})}$$

Beschreibt man nun die Transmissionsfunktion in  $\tilde{x}$  und  $\tilde{y}$ , so lässt sich das umgeschriebene Fraunhofer'sche Beugungsintegral

$$E(u, v) = \frac{CE_0 e^{i\phi_0}}{R_0 R'_0} \iint_{-\infty}^{\infty} \tau(\tilde{x}, \tilde{y}) e^{-i2\pi(u\tilde{x} + v\tilde{y})} d\tilde{x} d\tilde{y}$$

als Fouriertransformation identifizieren:

$$E(u, v) = \frac{CE_0 e^{i\phi_0}}{R_0 R'_0} T(u, v)$$

## 2.4.5 Verknüpfung

## 2.5 Beschreibung "realen" Lichtes

### 2.5.1 Kohärenz

Die Kohärenz ist ein Maß für die Endlichkeit der Wellenzüge einer Lichtquelle und beschreibt damit, inwiefern diese durch Überlagerung einzelner Wellenpakete dargestellt werden kann. Sie liefert damit einen Anhaltspunkt für die Interferenzfähigkeit einer Lichtquelle [11] [9]. Es wird zwischen zeitlicher, räumlicher und raumzeitlicher Kohärenz unterschieden. Dabei stellt die raumzeitliche Kohärenz eine Kombination aus zeitlicher und räumlicher Kohärenz dar.

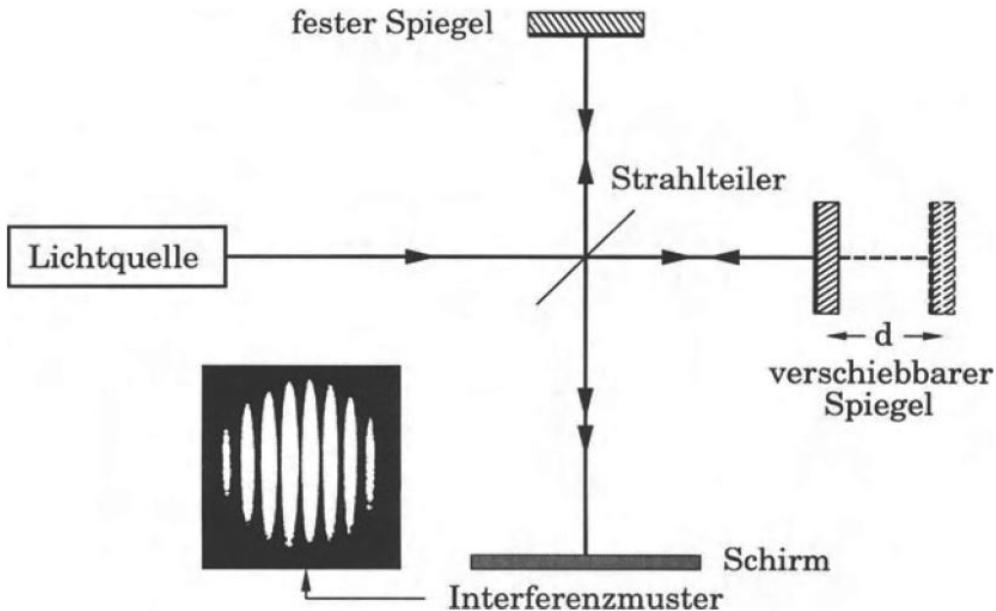


Abbildung 8: Das Michelson-Interferometer teilt einen Lichtstrahl in zwei Hälften und lässt diese durch unterschiedliche Weglängen interferieren[11]

Ausgehend von dem hier skizzierten Michelson Interferometer, welches unter anderem zur Widerlegung der Theorie, dass Licht sich in einem Äther genannten Medium ausbreite, verwendet wurde, sei zunächst nun die **zeitliche Kohärenz** erklärt. Angenommen der Strahlteiler teile das ankommende Licht zu gleichen Teilen, so habe die eine Hälfte die Laufzeit  $t$  und die andere Hälfte eine Laufzeit von  $t + \tau$ .  $\tau$  lässt sich aus der Lichtgeschwindigkeit und der Verschiebung  $d$  errechnen. Es gilt daher die Beziehung  $\tau = \frac{2d}{c}$ .

Aufgrund des Strahlteilers setzt sich die Gesamtfeldstärke aus den beiden Teilstärken zusammen  $E(t) = E_1(t) + E_2(t)$  und aufgrund der Beziehung  $E_1(t) = E_2(t + \tau)$  gilt

$$E(t) = E_1(t) + E_1(t + \tau)$$

Aus der Definition der Intensität ergibt sich nun

$$I = I_1 + I_2 + 2\operatorname{Re}(E_1^*(t)(E_1(t + \tau)))$$

Die Intensität ist also aus der Summe der Teilintensitäten sowie einem Interferenzterm gegeben. Das komplexe Skalarprodukt der beiden Feldstärkefunktionen lässt sich nun über die Definition des Skalarproduktes für Funktionen als Integral identifizieren, diese sog. Selbstkohärenzfunktion ist definiert als:

$$\Gamma(\tau) := E_1^*(t)(E_1(t + \tau)) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} E_1^*(t)(E_1(t + \tau)) dt$$

Angewendet auf eine harmonische Welle der Form

$$E_1(t) = E_0 e^{-i\omega t}$$

ergibt sich die Selbstkohärenzfunktion zu

$$\Gamma(\tau) = I_1 e^{-i\omega\tau}$$

Es zeigt sich also, dass diese ebenfalls harmonisch ist. Die Gesamtintensität ergibt sich deshalb zu

$$I = 2I_1 + 2\operatorname{Re}(\Gamma(\tau)) = 2I_1(1 + \cos(\omega\tau))$$

Die normierte Selbstkohärenzfunktion ist definiert als

$$\gamma(\tau) = \frac{\Gamma(\tau)}{\Gamma(0)} = \frac{\Gamma(\tau)}{I_1}$$

und ist somit betraglich immer kleiner oder gleich 1.

In der Realität misst man diese Größe aus Praktikabilitätsgründen für aufeinanderfolgende Intervalle.. Hierzu wird der zu betrachtende Zeitraum des Wellenzuges in äquidistante Intervalle halber mittlerer Wellenlängenäquivalente  $J = [\tau_n, \tau_{n+1}[$  unterteilt. Innerhalb eines solchen Intervall ist die Kontrastfunktion definiert als:

$$K(\tau \in J) := \frac{I_{\max}(J) - I_{\min}(J)}{I_{\max}(J) + I_{\min}(J)}$$

Befinden sich die betrachteten Zeitabstände im Bereich einer halben Wellenlänge, so gilt, dass die Kontrastfunktion gleich dem Betrag der Selbstkohärenz ist.

$$K(\tau) = |\gamma(\tau)|$$

Da die meisten Lichtquellen eine monoton fallende Kontrastfunktion besitzen, wird diese nun zur Definition einiger zentraler Begriffe verwendet. Eine Lichtquelle heißt:[11]

- vollkommen kohärent, falls  $|\gamma(\tau)| = 1$

- partiell kohärent, falls  $|\gamma(\tau)| < 1$
- vollkommen inkohärent, falls  $|\gamma(\tau)| = 0$

Ferner ist die sogenannte **Kohärenzzeit**  $\tau_c$  einer partiell kohärenten Lichtquelle als diejenige Zeit definiert, bei der  $|\gamma(\tau)|$  auf  $\frac{1}{e}$  abgefallen ist.

Im Falle einer Selbstkohärenz, wie es beim Michelson-Interferometer der Fall ist, lässt sich dies mithilfe der Lichtgeschwindigkeit auch als Länge  $l_c = c\tau_c$  ausdrücken, diese Länge wird dann **Kohärenzlänge** genannt.

Angesichts der Tatsache, dass auch räumliche Phänomene einen Verlust der Interferenzfähigkeit hervorufen können, sei an dieser Stelle die **räumliche Kohärenz** anhand des bereits erwähnten Young'schen Doppelspaltversuches näher erklärt.

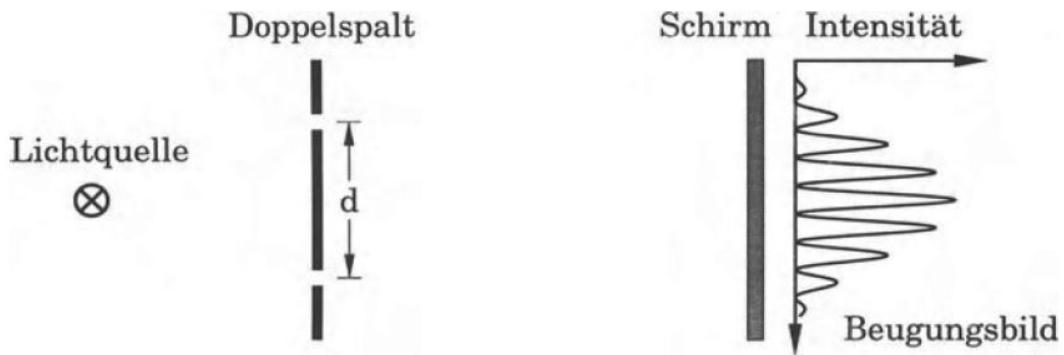


Abbildung 9: Der Youngsche Doppelspaltversuch [11]

Das elektrische Feld einer Huygens'schen Elementarwelle ist als Funktion des Abstands von ihrem Ursprung und der Zeit gegeben.

$$E(s, t) = E_0 e^{-i(k_s \omega t)}$$

Drückt man die komplexe Amplitude  $E_0$  in Polarkoordinaten aus, so lässt sich der Term umschreiben

$$E(s, t) = A e^{-i(k_s \omega t + \phi)}$$

Es seien die Abstände eines Punktes auf dem Schirm zu den Spalten gegeben durch  $s_1, s_2$ , so ergibt sich das elektrische Feld an diesem Punkt durch Überlagerung der Teilfelder

$$E(x(s_1, s_2), y(s_1, s_2), 0, t) = E_1(s_1, t) + E_2(s_2, t)$$

Die Intensität ergibt aus dem Skalarprodukt

$$\begin{aligned} I &= (E_1 + E_2) \cdot (E_1 + E_2)^* \\ &= A_1^2 + A_2^2 + 2A_1 A_2 \cdot \cos(k(s_2 - s_1)\phi_2 - \phi_1) \end{aligned}$$

Wie bei der zeitlichen Kohärenz ergibt sich die Intensität also aus der Summe der Einzelintensitäten sowie einem Interferenzterm.

Da sich der Schirm im Fernfeld befindet, können  $s_1$  und  $s_2$  pythagoräisch durch  $x, y$  und dem Schirmabstands  $z_0$  genähert werden

$$\begin{aligned} z_{1,2} &= \sqrt{\left[\left(x \mp \frac{d}{2}\right)^2 + y^2 + z_0^2\right]} \\ &= z_l \sqrt{1 + \frac{y^2}{z_l^2} + \frac{(x \mp \frac{d}{2})^2}{z_l^2}} \end{aligned}$$

Da sich der Schirm im Fernfeld befindet, sind  $x, y, d$  alle deutlich kleiner als  $z_0$ , aus diesem Grund kann der Wurzelterm in erster Ordnung genähert werden

$$z_{1,2} = z_l \left( 1 + \frac{y^2}{2} + \frac{(x \mp \frac{d}{2})^2}{2z_l^2} \right)$$

Unter der Annahme gleicher Amplituden der erzeugenden Wellen im Spalt lassen sich diese im Falle der Abnahme um  $\frac{1}{2}$  als  $\frac{A_0}{z_l}$  nähern.

Durch Einsetzen dieser Abschätzungen in die Funktion der Intensität, lässt sich diese folgendermaßen ausdrücken

$$I(x, y, 0) = \frac{2A_0^2}{z_l^2} \left[ 1 + \cos\left(\frac{2\pi d}{\lambda z_l}\right) + \phi_2 - \phi_1 \right]$$

Dadurch wird ein in x moduliertes und in y ausgedehntes Streifenmuster erzeugt. Hierbei beträgt der Abstand der Streifen

$$a = \frac{\lambda z_l}{d}$$

Zudem ist das erste Maximum aufgrund der Phasenverschiebung verschoben

$$x = a \frac{\phi_1 - \phi_2}{2\pi}$$

Diese Phasenverschiebung lässt sich aus den Abständen  $r_1, r_2$  der Spalte zur Lichtquelle und der Wellenlänge  $\lambda$  berechnen

$$\phi_1 - \phi_2 = (r_1 - r_2) \frac{2\pi}{\lambda}$$

Damit ist die Verschiebung der Interferenzmuster gegeben

$$x = \frac{a}{\lambda} (r_1 - r_2)$$

Betrachtet man 2 Lichtquellen, so überlagern sich deren Interferenzmuster. In diesem Fall wird schnell ersichtlich, dass, im Hinblick auf die Kontrastfunktion, deren erste Maxima nicht weiter als eine halbe Streifenbreite verschoben sein sollten. Das ist dann der Fall, wenn der Gangunterschied  $|(r_1 - r_2)|$  kleiner als eine halbe Wellenlänge ist.

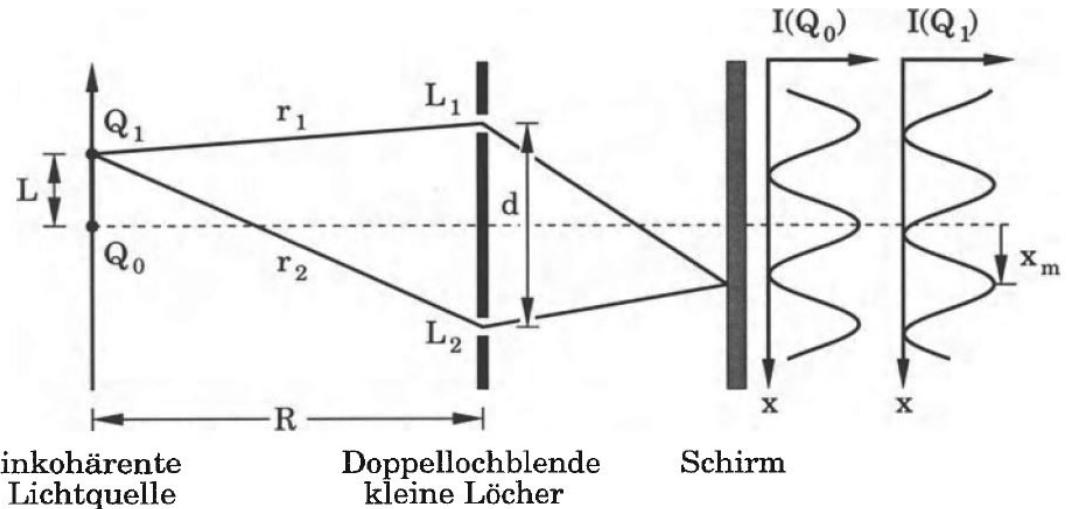


Abbildung 10: Der Youngsche Doppelspaltversuch mit inkohärenter Lichtquelle [11]

Analog zur vorherigen Betrachtung von  $z_1, z_2$  werden nun die Abstände  $r_1, r_2$  durch Vernachlässigen der y-Komponente pythagoräisch und in erster Ordnung genähert. Dadurch lässt sich der Gangunterschied mit den Materialkonstanten  $d, L, R$  ausdrücken

$$|r_1 - r_2| = \frac{dL}{R}$$

Daher gilt als Bedingung für die räumliche Kohärenz eines Aufbaus

$$\frac{dL}{R} < \frac{\lambda}{2}$$

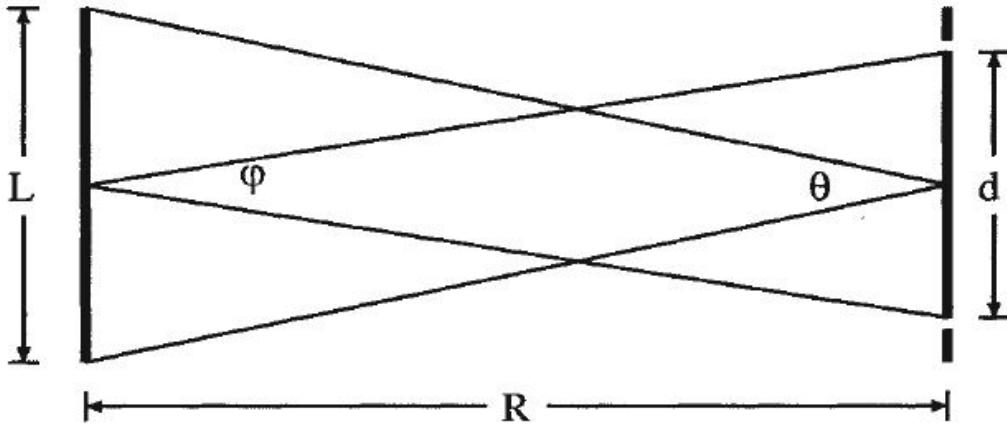


Abbildung 11: Die Winkelbezeichnungen räumlicher Kohärenz [11]

Für ausgedehnte Lichtquellen werden statt dieser Bedingung oft auch auf den Öffnungswinkeln basierende Näherungen verwendet

$$L \cdot \sin(\phi) < \frac{\lambda}{2}$$

$$d \cdot \sin(\Theta) < \frac{\lambda}{2}$$

Es zeigt sich also, dass die Interferenz von der Phasenverschiebung erzeugt wird, womit sich die räumliche Kohärenzfunktion achsennaher Interferenzmuster allein aus den Feldeigenschaften bei  $r_1, r_2$  ergibt. Die Kohärenzfunktion lässt sich dementsprechend als Kreuzkorrelation angeben

$$\Gamma(r_1, r_2, 0) = E(r_1, 0)E(r_2, 0)^*$$

Die **raumzeitliche Kohärenz** stellt eine Kombination aus der zeitlichen und der räumlichen Kohärenz dar. Für diesen verallgemeinerten Fall lautet die Kohärenzfunktion

$$\Gamma(r_1, r_2, t_1, t_2) = E(r_1, t_1)E(r_2, t_2)^* = E(r_1, t_1 + \tau)E(r_2, t_1)^*$$

### 2.5.2 Leuchtdioden und Weißlicht

Bei einer typischen Leuchtdiode handelt es sich aufgrund ihrer Halbleitereigenschaften um näherungsweise monochromatisches Licht einer bestimmten Wellenlänge, welche sich wie folgt berechnet [8]

$$\lambda(\Delta W) = \frac{hc_0}{\Delta W}$$

Hierbei handelt es sich bei  $\Delta W$  um Energiedifferenz der beim lichterzeugenden Elektronenübergang beteiligten Niveaus (Valenzelektronenpotentiale), typischerweise in eV (Elektronvolt) angegeben, und ist damit eine Materialeigenschaft der Leuchtdiode. Bei  $h$  handelt es sich um das Planck'sche Wirkungsquantum und bei  $c_0$  um die Vakuumlichtgeschwindigkeit.

Das Tageslicht hat hingegen ein breites Spektrum an Wellenlängen. Im Zuge des Trends, für den Menschen schonende Leuchtlösungen zu entwickeln, ist es daher nicht verwunderlich, dass Weißlicht LED entwickelt wurden. Eine vergleichsweise einfache Möglichkeit dies zu erreichen, ist mehrere verschiedenfarbige LED zu verwenden. Die wohl bekannteste ist die RGB-LED, welche aus einer roten, grünen und einer blauen LED besteht, dies bietet gerade in Partybeleuchtungen den Vorteil verschiedener Farbeffekte. Sind alle drei LED eingeschaltet, so wirkt die Gesamtfarbe durch additive Farbmischung weiß, weist allerdings große Lücken im Spektrum auf. Eine andere Methode ist die Ausnutzung von Photolumineszenz, hierbei wird ein Material durch Primärphotonen einer LED angeregt und emittiert Sekundärphotonen in einer gewünschten Verteilung. Dadurch ist es mittlerweile auch möglich, polychromatisches Licht ohne Lücken im Spektrum zu erzeugen.

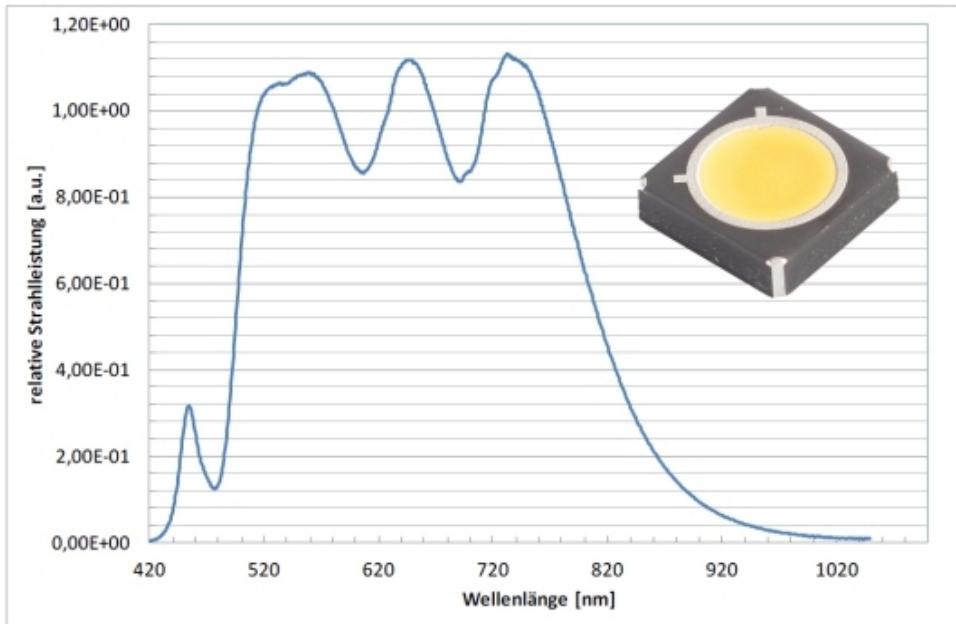


Abbildung 12: Beispiel einer kommerziellen Leuchtdiode mit kontinuierlichem Spektrum

### 2.5.3 Gaußstrahl

Um der Realität einer Lichtquelle etwas näher zu kommen, sei nun der sogenannte Gaußstrahl betrachtet, welcher eine rotationssymmetrische Auffächerung der Amplitudenverteilung vor und nach dem Fokuspunkt beschreibt. Im Fokus hat der Strahl die geringste Auffächerung und damit den kleinsten Radius der Einhüllenden. Der Ursprung der z-Achse wird zudem aus Symmetriegründen in den Fokus gelegt, dieser Radius  $\omega(0) = \omega_0$  wird **Waist-Radius** genannt.

Der Gaußstrahl ist eine Lösung der paraxialen Approximation der Wellengleichung einer sich in z-Richtung ausbreitenden Welle und erfüllt deshalb die Maxwell-Gleichungen nur näherungsweise, stellt aber bei hinreichend kleinem Waist-Radius  $\omega_0$  eine gute Approximation dar [10].

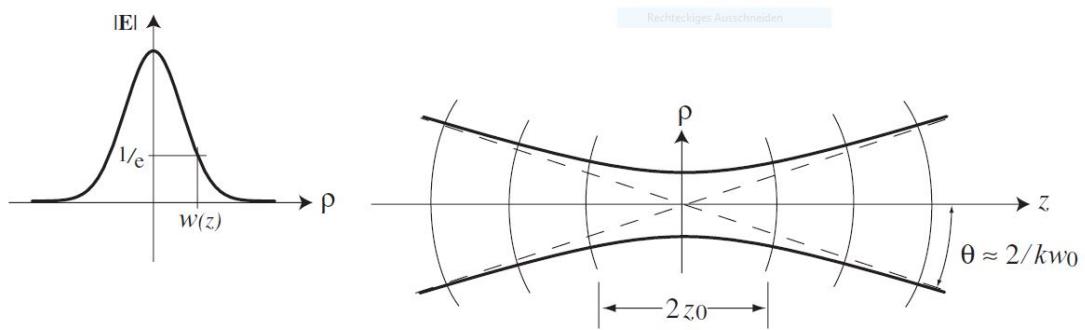


Abbildung 13: Visualisierung eines Gaußstrahls und seiner Amplitudenspektren

Das Amplitudenspektrum ist gaußförmig mit dem lokalen Strahlradius  $\omega(z)$ , welches als diejenige Entfernung definiert ist, bei der die Amplitude im Vergleich zu ihrem Maximum auf  $\frac{1}{e}$  abge-

sunken ist. Es gilt

$$E(\omega, z) = \frac{1}{e} E(0, z)$$

Dieser Radius folgt in Ausbreitungsrichtung einer Hyperbelform mit der als **Rayleigh-Länge** bezeichneten charakteristischen Länge  $z_0$ , bei der sich die transversale Amplitudenverteilung um den Faktor  $\sqrt{2}$  aufgeweitet hat.

$$\omega(z + z_0) = \sqrt{2}\omega(z)$$

Der Winkel  $\Theta$ , den die Asymptote der einhüllenden Hyperbel mit der z-Achse bildet, heißt **Divergenzwinkel** und berechnet sich aus [5]

$$\Theta = \frac{2}{k\omega_0}$$

Aufgrund der Rotationssymmetrie ist es sinnvoll, die Wellengleichung in Abhängigkeit von  $z$  und der Entfernung  $\rho = \sqrt{x^2 + y^2}$  zur z-Achse anzugeben.[5]

$$E(\rho, z) = E_0 \frac{\omega_0}{\omega(z)} e^{-\frac{\rho^2}{\omega^2(z)}} e^{i(kz - \eta(z) + \frac{k\rho^2}{2R(z)})}$$

Dies entspricht einer Kugelwelle mit den Korrekturtermen

- **Strahlradius**

$$\omega(z) = \omega_0 \sqrt{1 + \frac{z^2}{z_0^2}}$$

- **Radius der Wellenfront**

$$R(z) = z \left( 1 + \frac{z^2}{z_0^2} \right)$$

- **Phasenkorrektur(Gouy-Phase)**

$$\eta(z) = \arctan\left(\frac{z}{z_0}\right)$$

## 3 Konzeptionierung und Implementierung

### 3.1 Vorüberlegungen

Bevor eine Umsetzung erfolgen kann, ist eine Wahl über die zu verwendenden (Programmier-) Werkzeuge sowie über ein passendes Modell zu treffen. An dieser Stelle seien die Vorüberlegungen zu diesen Punkten näher erklärt. Auf die aufgetretenen Probleme, sowie deren Lösungen, wird zu einem späteren Zeitpunkt eingegangen.

#### 3.1.1 Analyse der Ausgangssituation

Wie bereits in der Aufgabenstellung beschrieben, ist das Ziel dieser Arbeit die Erstellung einer mathematischen Beschreibung und die Programmierung der Strahlpropagation und Superposition teilkohärenter Wellen für den Einsatz in einem Projekt zur Erstellung diffraktiver Optiken. Dies bedeutet im Bezug auf eine mögliche spätere Verwendung innerhalb des Projektes, eine möglichst praxistaugliche und an der Realität experimenteller Projekte orientierte Umsetzung anzustreben. Es ergeben sich daher zunächst folgende Anforderungen:

Das Programm soll in der Lage sein, eine reale Lichtquelle im Sinne einer vorgegebenen Spektralfunktion in vorgegebenen, äquidistanten Schritten abzutasten und aus weiteren vorgegebenen Materialparametern, die Verteilung des elektrischen Feldes in der optischen Ebene mittels Gauß-Propagation zu berechnen. An dieser Stelle soll das Programm aus der Transmissionsfunktion des optischen Elements und den vorgegebenen Entfernungslängen die Feldverteilungen in der Projektionsebene berechnen.

Zudem soll das Programm für jede abgetastete Wellenlänge den Strahlverlauf, sowie die Intensitätsverteilungen in der optischen- und sowie der Projektionsebene plotten.

Im Zuge einer praxisorientierten Anwendung soll das Programm des Weiteren mithilfe einer graphischen Benutzeroberfläche die Möglichkeit bieten, alle Berechnungsparameter individuell anzupassen, um schnelle Iterationen zu ermöglichen.

In Anbetracht der Tatsache, dass das Programm mit hoher Wahrscheinlichkeit im Nachgang keine Wartung und Updates durch den ursprünglichen Autor erhält, sind die Programmiersprache und die dafür nötigen Tools so zu wählen, dass eine Einarbeitung recht zügig und kostenfrei zu realisieren ist. In den folgenden Absätzen wird nun näher auf die aus diesen Überlegungen entstandenen Entscheidungen eingegangen.

#### 3.1.2 Wahl der Programmiersprache

Als Programmiersprache dieser Arbeit wurde Python gewählt, da diese Sprache gratis verfügbar und in der wissenschaftlichen Praxis weit verbreitet ist. Aus diesem Grund sind für diese Anwendungen bereits viele Bibliotheken mit passenden Funktionen verfügbar, von denen einige in dieser Arbeit verwendet wurden. Des Weiteren ist die Sprache vergleichsweise leicht zu erlernen und besitzt eine auch für Laien hohe Lesbarkeit des Codes.

Als Entwicklungsumgebung wurde die Communityversion von PyCharm verwendet, welche der Entwickler JetBrains (Stand 11.2020) kostenlos zu Verfügung stellt. Für das akademische Umfeld steht zudem eine umfangreichere Academy-Version gratis zur Verfügung (Stand 11.2020). Diese Kombination aus Programmiersprache und Entwicklungsumgebung erlaubt eine kosten-

freie Implementierung und Wartung des Programms bei Beibehaltung des Funktionsumfanges und der einfachen Erlernbarkeit.

### 3.1.3 Beschreibung des Modells anhand des Programmablaufes

Zur Berechnung der Strahlpropagation müssen zunächst einige Modellannahmen gemacht werden. Diese Annahmen werden nun, anhand einer beispielhaften Ausgabe des Strahlverlaufes aus dem Programm näher erläutert. Dabei beziehen sich die Abbildungen 14-18 auf die in den Abbildungen 20-22 definierten Eingabeparameter, sowie die Abbildungen 14, 16 und 17 auf die fünfte gesampelte Wellenlänge bei etwa 647,27nm.

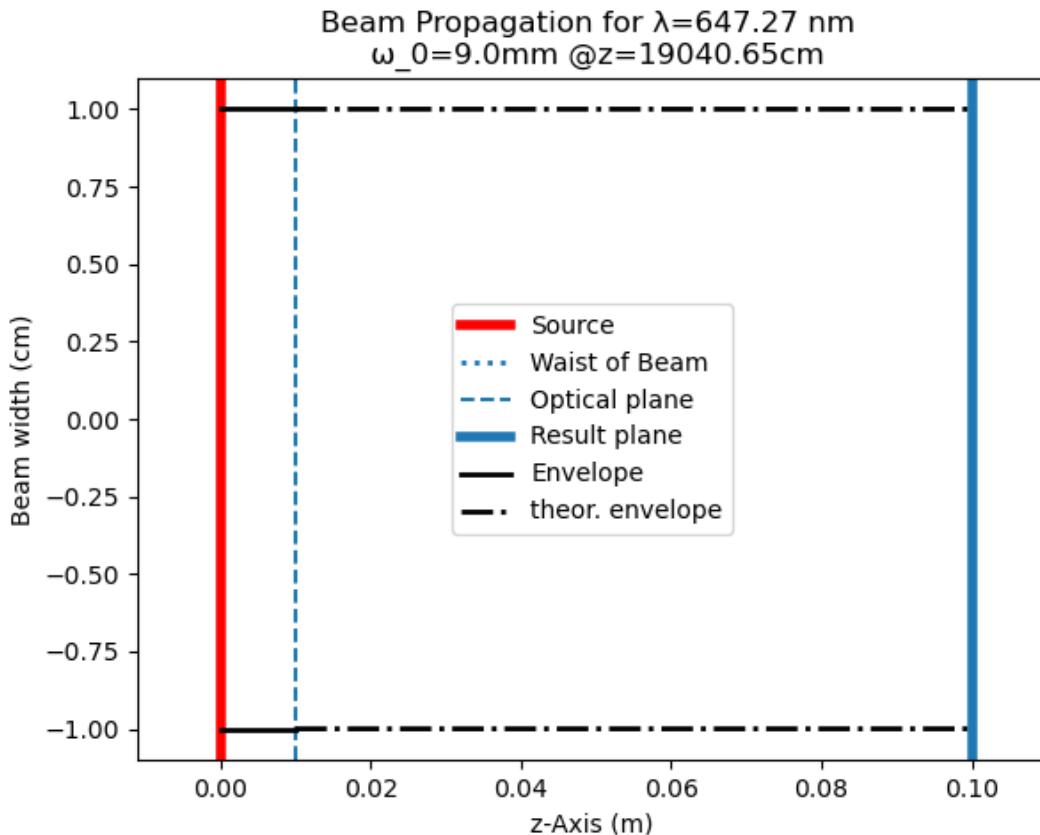


Abbildung 14: Programmausgabe der Strahlpropagation

Als erste geometrische Annahme sei die Lichtquelle als Ausgangspunkt eines Gaußstrahles approximiert. Die Ausbreitungsrichtung des Lichtes dient als z-Achse eines dreidimensionalen kartesischen Koordinatensystems, während die Position des Mittelpunktes der radialsymmetrischen Lichtquelle als Ursprung dessen definiert ist. Zudem seien sowohl das optische Element als auch der Schirm als an festgelegten z-Koordinaten zentriert angenommen. Sollte dies nicht der Fall sein, so müssen die Koordinaten vorher ggf. manuell transformiert werden.

Die Feldstärke  $E_0$  einer realen Lichtquelle folgt einer spektralen Verteilungsfunktion  $E_0(\lambda)$ , welche durch eine im Datenblatt eines Bauteils angegebene Intensitätsverteilung  $I_0(\lambda)$  approximiert werden kann (siehe Abbildung 12). Durch diese Approximation gehen freilich die Phaseninformationen verloren, weshalb in der Anwendung des Programms eine komplexwertige

ge Feldstärkeverteilung angenommen wird. Die Feldstärkeverteilung wird nun in einem vorher festgelegten Intervall äquidistant abgetastet und die daraus berechnete Intensität zum Vergleich mit einem eventuell vorhandenen Datenenblatt der Lichtquelle ausgegeben.

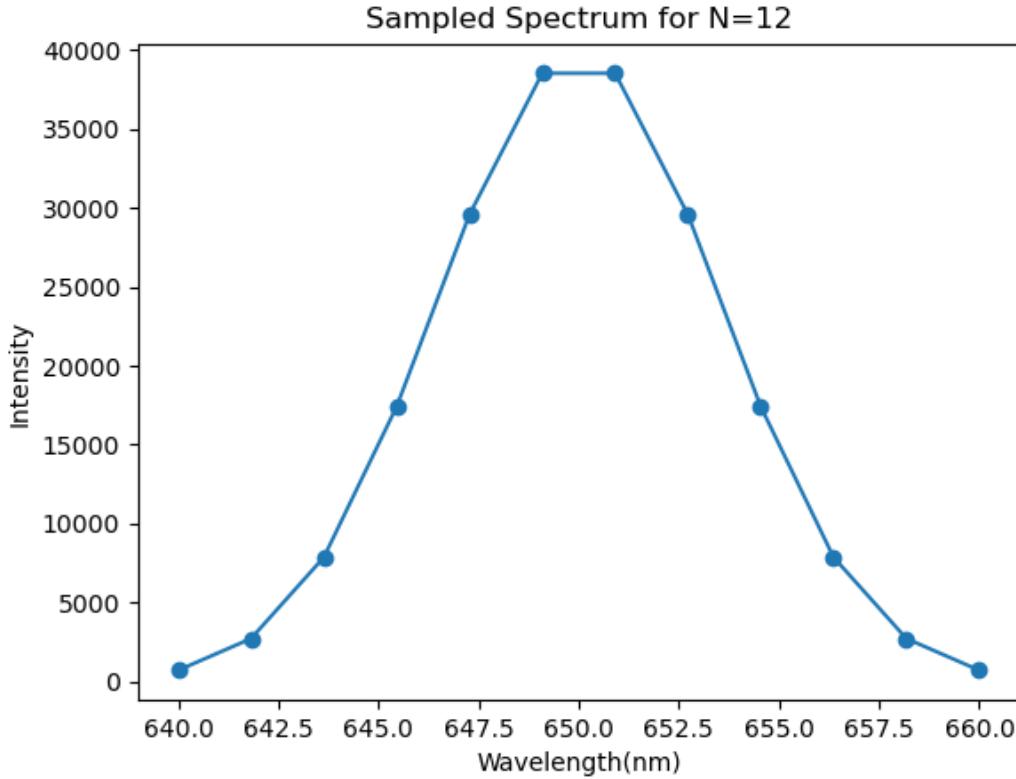


Abbildung 15: Programmausgabe des abgetasteten Spektrums

Durch Umstellen der Formel für das transversale Profil eines Gaußstrahles

$$\omega(z) = \omega_0 \sqrt{1 + \frac{z^2}{z_r^2}}$$

nach  $z$  lässt sich aus den Parametern der Quelle die Position des Fokuspunktes nach folgender Formel berechnen

$$z_0 = \frac{\pi \omega_0^2}{\lambda} \sqrt{\frac{\omega_s^2}{\omega_0^2} - 1}$$

Dabei handelt es sich bei  $z_r = \frac{\pi \omega_0^2}{\lambda}$  um die Rayleigh-Länge (siehe 2.5.3), bei  $\omega_s$  um den in den Quellparametern festgelegten Radius der Lichtquellenausdehnung und bei  $\omega_0$  um den Waistradius d.h. die Ausdehnung des Strahls im Fokuspunkt.

Mit diesen Informationen wird zunächst für jede Wellenlänge ein Profil der Strahlpropagation (siehe Abbildung 14) erstellt. Als nächstes wird anhand festgelegter Parameter  $n_x$ ,  $n_y$  sowie Intervallen  $I_x$  und  $I_y$  die optische Ebene an  $2^{n_x} \cdot 2^{n_y}$  Punkten äquidistant diskretisiert und mithilfe der in 2.5.3 erarbeiteten Feldstärkeformel des Gaußstrahls die jeweilige Feldstärke berechnet

$$E(\rho, z) = E_0 \frac{\omega_0}{\omega(z)} e^{-\frac{\rho^2}{\omega^2(z)}} e^{i(kz - \eta(z) + \frac{k\rho^2}{2R(z)})}$$

- Entfernung zur z-Achse

$$\rho = \sqrt{x^2 + y^2}$$

- Strahlradius

$$\omega(z) = \omega_0 \sqrt{1 + \frac{z^2}{z_0^2}}$$

- Radius der Wellenfront

$$R(z) = z \left( 1 + \frac{z^2}{z_0^2} \right)$$

- Phasenkorrektur(Gouy-Phase)

$$\eta(z) = \arctan\left(\frac{z}{z_0}\right)$$

Für jede Wellenlänge wird nun die Intensität in der optischen Ebene geplottet

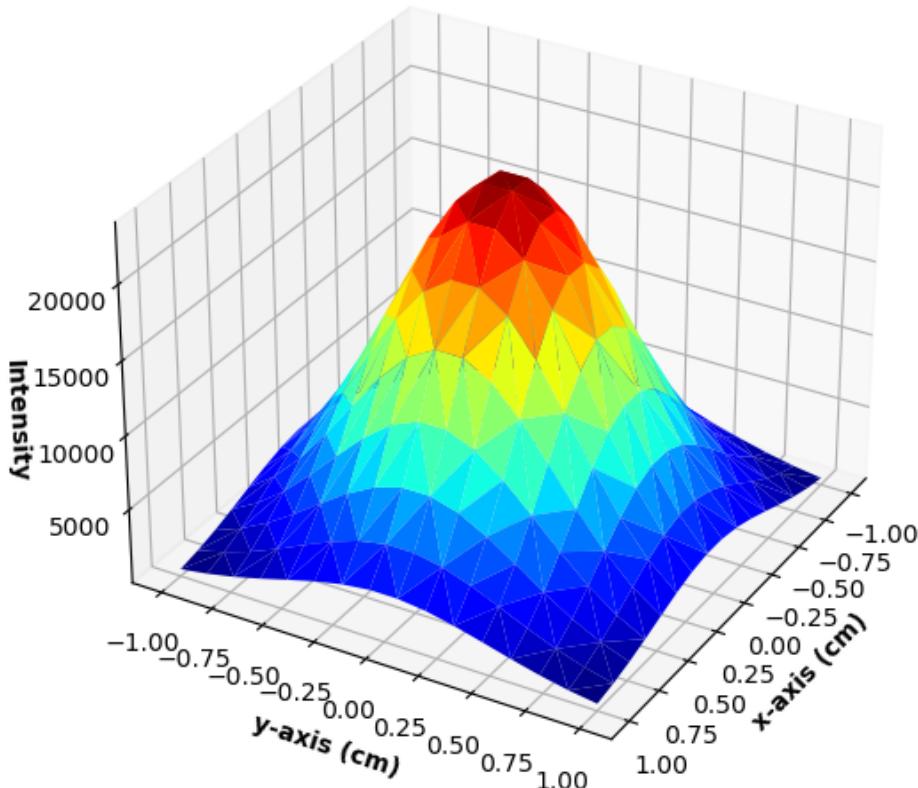


Abbildung 16: Programmausgabe der Intensität in der optischen Ebene

Nachdem nun für jeden gerasterten Punkt in der optischen Ebene die elektrische Feldstärke berechnet wurde, kalkuliert das Programm anhand festgelegter Parameter die Feldstärke für die Schirmebene. Hierzu wird diese zunächst, analog zur Rasterung der optischen Ebene, anhand festgelegter Parameter  $n_x$ ,  $n_y$  sowie Intervallen  $J_x$  und  $J_y$  an  $2^{n_x} \cdot 2^{n_y}$  Punkten äquidistant diskretisiert.

Ferner seien die x-, respektive y-Werte der Schirmebene mit u bzw. v bezeichnet. Zu jedem

Punkt  $P_{u,v}$  und jeder Wellenlänge  $\lambda$  wird nun die elektrische Feldstärke wie folgt berechnet

$$E_{Schirm}(u, v) = \sum_{x,y} C \cdot FFT \left( E_{opt}(x, y) \cdot \tau(x, y, \lambda) \cdot e^{\frac{2\pi i}{\lambda} \sqrt{(x-u)^2, (y-v)^2}} \cdot e^{-\frac{(x-u)^2 + (y-v)^2}{L_c^2}} \right)$$

Dabei handelt es sich bei den einzelnen Faktoren um

- $C$   
festgelegter Skalierungsfaktor
- $E_{opt}(x, y)$   
vorher berechnete el. Feldstärke in einem gerasterten Punkt der optischen Ebene
- $\tau(x, y, \lambda)$   
Wert der Transmissionsfunktion des optischen Elements zu diesem Punkt
- $e^{\frac{2\pi i}{\lambda} \sqrt{(x-u)^2, (y-v)^2}}$   
Phasenkorrektur aufgrund unterschiedlicher Laufzeiten
- $e^{-\frac{(x-u)^2 + (y-v)^2}{L_c^2}}$   
Raumzeitlicher Kohärenzfaktor mit der Kohärenzlänge  $L_c$  der Lichtquelle

Sind diese Werte berechnet, so wird die jeweilige Intensitätsverteilung in der Schirmebene, analog zur Intensitätsverteilung der optischen Ebene ausgegeben.

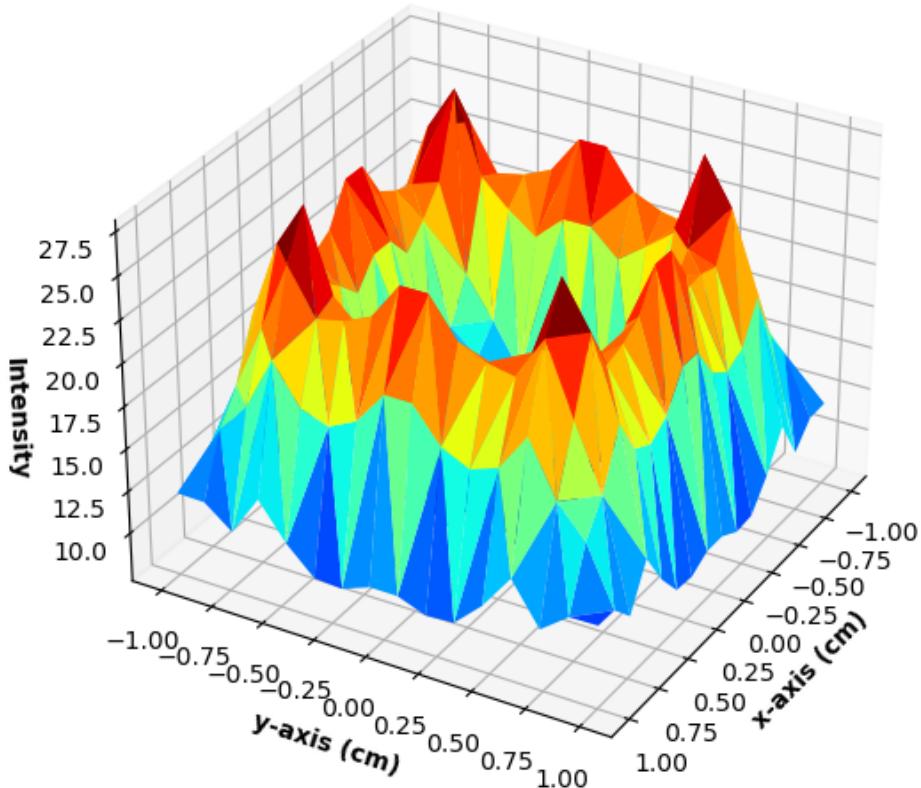


Abbildung 17: Programmausgabe der Intensität in der Schirmebene

Für die Ausgabe der Gesamtintensitätsverteilung (und damit dem sichtbaren Bild) werden nun die Intensitäten aller Wellenlängen addiert. Zusätzlich werden für alle Intensitätsplots die Rohdaten als txt-Datei ausgegeben.

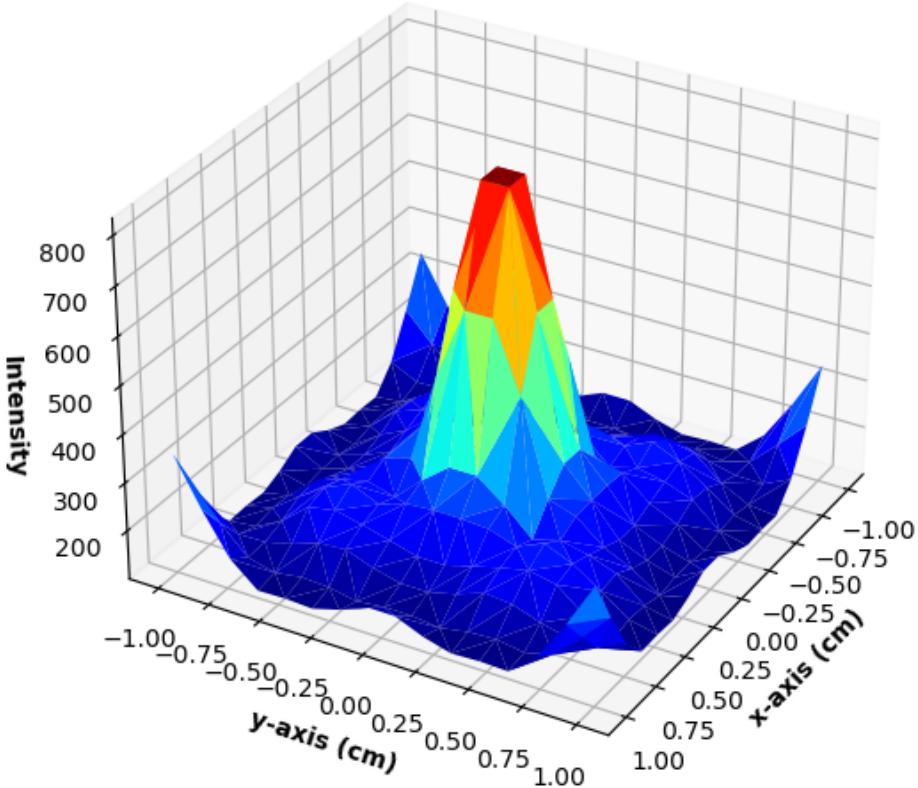


Abbildung 18: Programmausgabe der Gesamtintensität in der Schirmebene

### 3.2 Umsetzung

An dieser Stelle soll nun genauer auf die konkrete Umsetzung und Bedienung des Programmmoduls eingegangen werden. Zunächst wird die Bedienung der graphischen Benutzeroberfläche näher vorgestellt, im nächsten Schritt werden die einzelnen Ein- und Ausgabeparameter näher erklärt und als letztes die Programmfunctionen selbst im Detail beschrieben und Verbesserungspotentiale analysiert.

#### 3.2.1 GUI

Eine graphische Benutzeroberfläche, im Folgenden GUI (aus dem Englischen *graphical user interface*) genannt, stellt eine Visualisierung eines Programms dar und erlaubt schnellere und intuitivere Eingaben von Parametern, sowie direkte Rückmeldungen. Im Rahmen dieser Arbeit wurde das Programmmodul mit einer GUI ausgestattet, um schnell Iterationen und Parameteränderungen zu ermöglichen. Da eine GUI sehr intuitiv zu bedienen ist, setzt diese keine Programmierkenntnisse voraus, was in der Anwendung von Vorteil ist. Aufgrund der Internationalität der (natur-)wissenschaftlichen Forschung sind sowohl die GUI, als auch der Programmcode selbst in englischer Sprache verfasst. Ferner sind alle Einheiten, sofern nicht anders

erwähnt, in SI-Einheiten angegeben.

Die GUI wurde mit der Python Bibliothek tkinter erstellt, welche frei unter der Python Lizenz verfügbar ist und damit de facto Standard für GUI in Python ist.

Nach Start des Programms öffnet sich zunächst die Hauptseite des Programms mit den Buttons zur Berechnung, der Ausgabe als Plot der Berechnungen sowie der Möglichkeit, Einstellungen zu speichern bzw. zu laden. Um die Einstellungen zu ändern, kann mithilfe der oberen Reiter eine Unterseite ausgewählt werden. Dabei gilt folgende Unterteilung:

- **Calculation**

Hier befinden sich die Funktionalitäten zur Berechnung, Ausgabe und Speicher- bzw. Ladefunktion

- **Settings**

An dieser Stelle sind allgemeine Einstellungen wie die Anzahl der Rasterungen für Wellenlängen und Geometrien, Speicherort, Winkelangaben der plotting Funktion sowie die Geometrie der Schirmebene zu finden.

- **Optical Element**

Auf dieser Seite befinden sich die Parameter des optischen Elements.

- **Optical Element**

Hier sind die Parameter der Lichtquelle definiert.

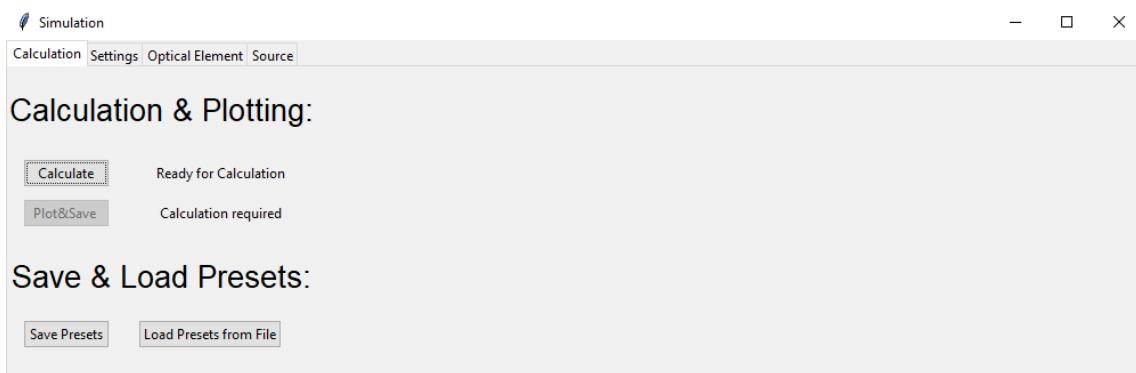


Abbildung 19: Berechnungsseite der graphischen Benutzeroberfläche

Dies ist die sich beim Start des Programms automatisch öffnende Seite.

Im oberen Abschnitt lassen sich die Funktionen der Berechnung und der Ausgabe durch Klicken der Buttons Calculate, respektive Plot&Save starten. Der Button Plot&Save lässt sich jedoch erst nach einer erfolgten Berechnung anklicken. Wird eine Ausgabe mittels Plot&Save gestartet, so werden die Einstellungen zusätzlich gespeichert und damit einer Datei zum späteren Einlesen ausgegeben. Jeweils rechts der entsprechenden Buttons befindet sich ein Statustext, der je nach momentanem Stand angepasst wird und einen Anhaltspunkt liefert, in welchem Funktionsschritt sich das Programm gerade befindet.

Im unteren Abschnitt befinden sich die Funktionen zum Speichern und Laden der Einstellungen. Durch einen Klick auf den Button Save Presets lassen sich die Einstellungen manuell

abspeichern, während ein Klick auf Load Presets from File ein Dateiauswahlfenster öffnet, um eine vorher gespeicherte Datei auszuwählen und ihre Einstellungen einzulesen.

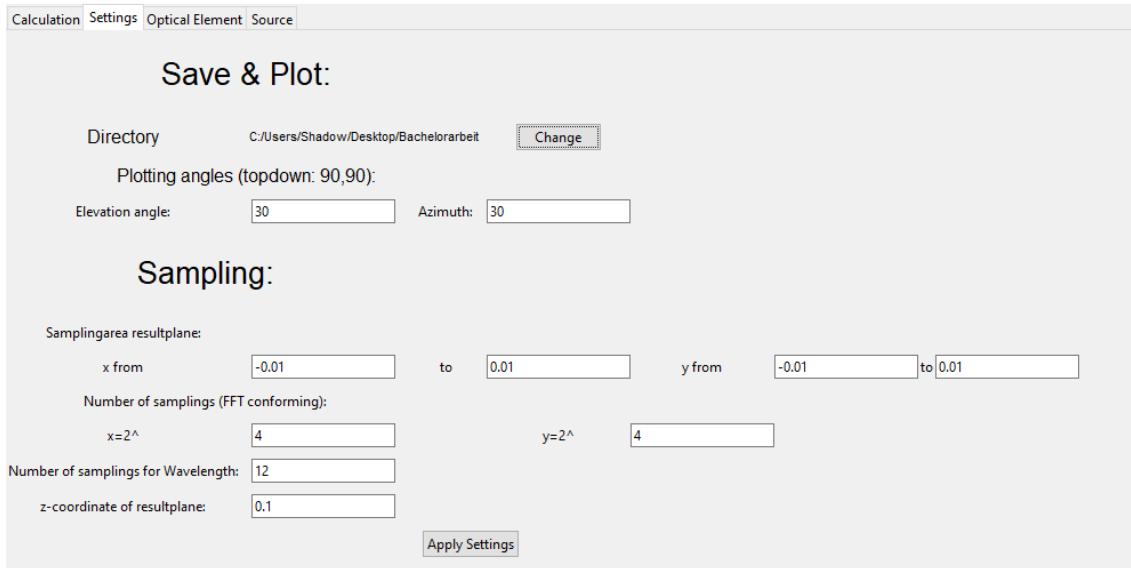


Abbildung 20: Einstellungsseite der graphischen Benutzeroberfläche

Die Settings Seite enthält generische Einstellungen (oberer Part) sowie die Geometriedaten der Schirmebene und Rastereinstellungen (unterer Part).

Von oben nach unten befinden sich auf dieser Seite folgende **Einstellungsmöglichkeiten**. Hinter **Directory** ist der **Speicherpfad der Ausgaben** angegeben. Mit einem Klick auf den Button Change öffnet sich ein Auswahlfenster, mithilfe dessen ein anderer Pfad gewählt werden kann.

**Plotting Angles** legt die **Winkel** der Sicht auf die **Intensitätsplots** fest, wobei 90-90 einer Sicht von oben entspricht.

Unter **Samplingarea resultplane** sind die abzutastenden **Intervalle der Schirmebene** eingegeben. Dies entspricht einem rechteckförmigen Auschnitt der Schirmebene, wobei sich der Punkt (0,0) im Mittelpunkt des Gaußstrahls befindet.

**Number of Samplings** (FFT conforming) erwartet die Eingabe einer **natürlichen Zahl als Exponenten**, jeweils **für die Anzahl der Rasterungen** in x -bzw y. So entspricht eine Angabe von 4,4 jeweils  $2^4 = 16$  Rasterungen in x bzw y und somit 256 diskreten Punkten.

Das Eingabefeld **Number of samplings for Wavelength** legt die **Anzahl der Wellenlängen** fest, welche äquidistant abgetastet werden. Die dazugehörigen Einstellungen der Spektralfunktion und des Abtastintervalls befinden sich im Reiter Source bei den Einstellungen der Lichtquelle.

**z-coordinate of resultplane** gibt die **z-Koordinate der Schirmebene** an.

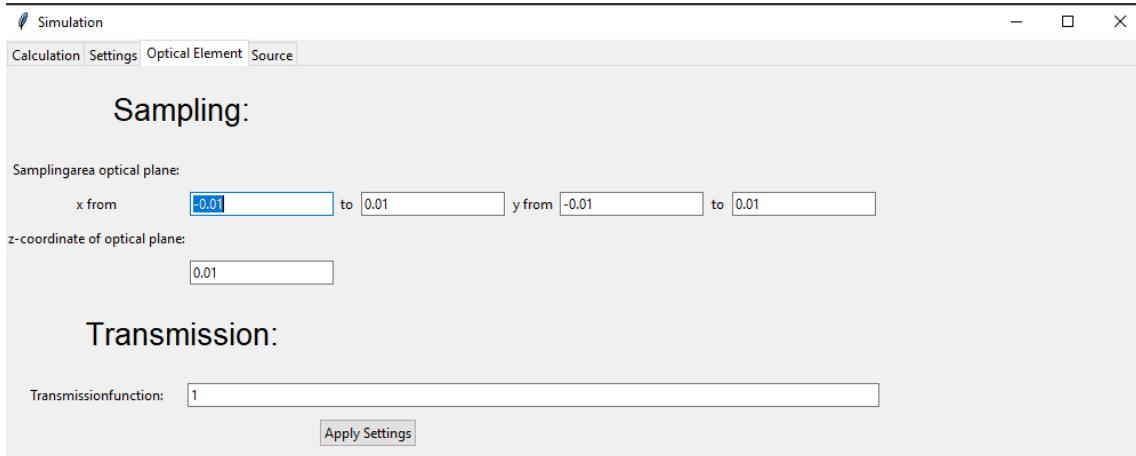


Abbildung 21: Einstellungen des optischen Elements in der graphischen Benutzeroberfläche

Auf der Optical Element Seite sind die Einstellungen des optischen Elements zu finden. Zunächst sind die Geometriedaten des Elements, in analoger Notation zu den Geometriedaten der Schirmebene auf Settings Seite, im oberen Teil einstellbar. Im unteren Teil ist die **Transmissionsfunktion** des optischen Elements einstellbar. Hier kann eine (nahezu) beliebige mathematische Funktion der Variablen  $x, y, l$  in Python-Notation angegeben werden. Diese Funktion wird dann in der Berechnung als  $\tau(x, y, \lambda)$  interpretiert.

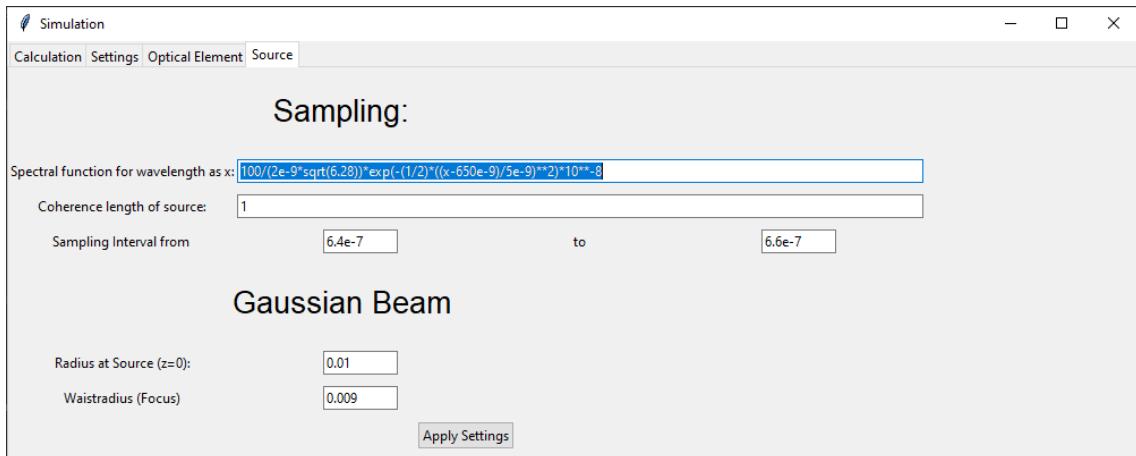


Abbildung 22: Einstellungen der Lichtquelle in der graphischen Benutzeroberfläche

An dieser Stelle sind die Parameter der Lichtquelle, engl. **Source** zu finden. Von oben nach unten sind hier folgende Einstellungen möglich:

**Spectral function for wavelength as x** erlaubt es, analog zur Transmissionsfunktion  $\tau$  des optischen Elements, die **Spektralfunktion der Lichtquelle** als Funktion von  $\lambda$  anzugeben. Hier ist allerdings zu beachten, dass die Spektralfunktion explizit im Sinne des elektrischen Feldes und nicht der Intensität erwartet wird (ggf. ist hier also eine Transformation der Daten aus dem Datenblatt der Quelle nötig) und zudem die Variable  $\lambda$  als  $x$  angegeben werden muss. Aus dieser Funktion werden im Berechnungsschritt die Werte  $E_0$  für jede Wellenlänge berechnet.

Das Eingabefeld **Coherence length of Source** erlaubt es die **Kohärenzlänge der Quelle** anzugeben. Die in der Graphik angegebene Kohärenzlänge entspricht 1m und ist damit als

eher unrealistisch einzuordnen. Eine typische Kohärenzlänge für eine Leuchtdiode entspricht 4 Mikrometer, in Python-Notation 4e-6.

### 3.2.2 Ausgabeschema der Dateien

Durch das Starten einer Berechnung wird in dem auf der Settings Seite festgelegten Ordner ein Unterordner nach folgendem Schema erzeugt:

DD\_MM\_YYYY\_hhmm

Dementsprechend wurde der Ordner 22\_11\_2020\_1814 am 22. November 2020 um 18:14 angelegt. Diese Struktur macht es möglich, seriell durchgeführte Berechnungen voneinander zu unterscheiden. Aufgrund der Unterscheidbarkeit ist es daher notwendig, dass das Programm sich nach einem Plot selbst schließt und erneut gestartet werden muss. Innerhalb dieses Ordners befindet sich nun folgende Struktur

Dieser PC > Desktop > Bachelorarbeit > 22_11_2020_1814			
Name	Änderungsdatum	Typ	Größe
Beam Propagation	22.11.2020 18:14	Dateiordner	
Data	22.11.2020 18:14	Dateiordner	
Intensity_OPT	22.11.2020 18:14	Dateiordner	
Intensity_RES	22.11.2020 18:14	Dateiordner	
InputData	22.11.2020 18:14	Textdokument	1 KB
Result	22.11.2020 18:14	PNG-Datei	130 KB
Result	22.11.2020 18:14	Textdokument	14 KB
Spectrum	22.11.2020 18:14	PNG-Datei	34 KB

Abbildung 23: Ordnerstruktur der Dateiausgabe

Hierbei handelt es sich um folgende Ordner bzw. Dateien:

- **Beam Propagation**

Hier befinden sich die Plots der Strahlpropagation für jede Wellenlänge (vgl. Abb. 14).

- **Data**

In diesem Ordner werden die gespeicherten Daten für die Ladefunktion abgelegt

- **Intensity\_OPT**

Dies ist der Speicherort für die Plots in der optischen Ebene (vgl. Abb. 16), sowie die dazugehörige Ausgabe der Rohdaten als .txt Datei.

- **Intensity\_RES**

Analog zur Ausgabe der Plots in der optischen Ebene, werden hier die Plots in der Schirmebene (vgl. Abb. 17) und die dazugehörigen Daten ausgegeben.

- **Input Data**

In dieser Datei werden die Eingabeparameter in einer benutzerfreundlichen Form ausgeben.

- **Result**

Diese zwei Dateien enthalten den Plot der Gesamtintensität (vgl. Abb 18) sowie die dazugehörigen Rohdaten.

- **Spectrum**

In dieser Datei werden die Grundintensitäten für jede abgetastete Wellenlänge geplottet. (vgl. Abb. 15)

### 3.2.3 Ausgewählte Funktionen und Parallelisierung

Das Programm besteht aus 6 Python Dateien

- **main.py**

Der hier enthaltene Code wird bei Start des Programms ausgeführt und hat lediglich die Funktion, die GUI zu starten bzw. eventuelles Debugging in der Konsole zu erlauben.

- **GUI.py**

Hier befindet sich die GUI-Logik und damit das Frontend des Programms. An dieser Stelle sind die Geometrien der einzelnen Buttons und Eingabefelder sowie die Anbindung an das Backend definiert.

- **SourceData.py, OpticalElementData.py, Settings.py**

Diese 3 Klassen sind zur Speicherung von Daten vorgesehen und entsprechen somit den Eingabeparametern der Abbildungen 20-22.

- **Calculation.py**

Hier befindet sich die eigentliche Berechnungslogik. Auf ausgewählte Funktionen dieser wird nun im Folgenden eingegangen.

Im Folgenden sei auf einige Abschnitte der Calculation.py Datei näher eingegangen. Aufgrund der Tatsachen, dass Python stark auf Einrückungen basiert und eine Darstellbarkeit auf Papier kein scrolling erlaubt, musste hier des Öfteren von einem Zeilenumbruch innerhalb einer Codezeile Gebrauch gemacht werden. In Python steht ein Backslash am Ende einer Zeile für eine Fortsetzung dieser Codezeile in der nächsten Zeile.

## Codeabschnitt Bibliotheken

```
1 import sympy
2 import numpy
3 import os #operating system
4 import time#time
5 from time import strftime
6 import matplotlib.pyplot as plt#plotting
7 from matplotlib.ticker import LinearLocator, FixedLocator
8 from matplotlib.ticker import FormatStrFormatter
9 from mpl_toolkits.mplot3d import Axes3D
10 from matplotlib import cm
11 import threading #threading
12 import pickle#saving data
13 #my classes
14 from SourceData import SourceData
15 from OpticalElementData import OpticalElementData
16 from Settings import Settings
```

In diesem Codeabschnitt sind die für die Berechnung notwendigen Bibliotheken definiert. Besonders erwähnenswert sind hierbei sympy, numpy und matplotlib.

sympy erlaubt unter Anderem das Parsen von Strings als Funktion und stellt daher eine der Hauptfunktionalitäten zur Verfügung. Die als String angegebenen Transmissions- und Spektralverteilungsfunktionen werden mithilfe dieser Bibliothek durch Abtasten in passende numerische Werte konvertiert.

numpy ist eine numerische Bibliothek für Python und stellt unter Anderem einen einfacheren Datentyp für mehrdimensionale Objekte zur Verfügung.

matplotlib ist eine Bibliothek für Graphikausgaben und erlaubt es, aus dem Programm heraus Daten zu visualisieren.

## Codeabschnitt Hauptberechnungsfunktion

```
1 def Calculate_for_Wavelength(self , wavelength):
2     print("Starting calculation")
3     "Calculates Fields for both planes and writes into Data"
4     #coherence length
5     L=self.SourceData.source_coherencelength
6     #scaling factor
7     C=1
8     i=wavelength
9     #distances
10    dist_source_opt=self.OpticalElementData.oe_coordinates[2]
11    dist_opt_im = self.Settings.image_coordinates[2]\ \
12      - self.OpticalElementData.oe_coordinates[2]
13    # Sample the Intensity function
14    # Get Lambda
15    la=self.SourceData.source_samplingarea[0]\ \
16      + self.d_lambda_step * i
17    self.calc_sampling_lambda[i]=la
18    # Waistrad
19    self.calc_sampling_waistrad[i]=\ \
20      self.SourceData.source_waistrad
```

```

21 # Offsets
22 self.calc_sampling_offsets[i]=\
23 [0, 0, self.Calculate_z_offset(i)]
24 # E_0(lambda) Field of Wavelengths
25 self.calc_sampling_E0[i]=\
26 self.SourceData.source_spectrum_sympify\
27 .subs("x", self.calc_sampling_lambda[i])
28 # loop x
29 for m in range(0, 2 ** self.Settings.sampling_FFT_N[0]):
30     # loop y
31     for n in range(0, 2 ** self.Settings.sampling_FFT_N[1]):
32         # Calculate coordinates in optical and result plane
33         self.calc_CoordsOPT_lambda_xy[i][m][n][0]=\
34             self.OpticalElementData.oe_samplingarea[0][0] +\
35             self.d_x_step_OPT * m #
36         self.calc_CoordsOPT_lambda_xy[i][m][n][1]=\
37             self.OpticalElementData.oe_samplingarea[1][0] +\
38             self.d_y_step_OPT * n #
39         self.calc_CoordsRES_lambda_xy[i][m][n][0]=\
40             self.Settings.image_samplingarea[0][0] +\
41             self.d_x_step_RES * m #
42         self.calc_CoordsRES_lambda_xy[i][m][n][1]=\
43             self.Settings.image_samplingarea[1][0] +\
44             self.d_y_step_RES * n #
45         # Calculate E(x,y,z_optic)
46         # in optical plane for each wavelength
47         self.calc_Eopt_lambda_xy[i,m,n]=\
48             self.Calculate_Field(i, \
49             self.calc_CoordsOPT_lambda_xy[i][m][n][0], \
50             self.calc_CoordsOPT_lambda_xy[i][m][n][1])
51 #calculate E in resultplane
52 #Calculate resulting Field
53 print("calculating E_res")
54 phaseshift= lambda l,x,u,y,v:\
55 numpy.exp((2j*numpy.pi/l)*((x-u)**2+(y-v)**2)**0.5)
56 coherencefactor=lambda l,x,u,y,v:\
57 numpy.exp(-((x-u)**2+(y-v)**2)/L**2)
58 #Transmission Matrix and Intensitysum (opt)
59 for m in range(0, 2 ** self.Settings.sampling_FFT_N[0]):
60     for n in range(0, 2 ** self.Settings.sampling_FFT_N[1]):
61         x=self.calc_CoordsOPT_lambda_xy[i][m][n][0]
62         y=self.calc_CoordsOPT_lambda_xy[i][m][n][1]
63         self.calc_transmission_lambda_xy[i][m][n]=\
64             self.OpticalElementData.\
65             oe_transmissionfunction_sympify.subs\
66             ((["x",x],["1",la],[ "y",y]), evaluate=True)
67 #calculate Results for each xy and Intensitysum (res)
68 for u_i in range(0, 2 ** self.Settings.sampling_FFT_N[0]):
69
70     for v_i in range(0, 2 ** self.Settings.sampling_FFT_N[1]):
71         u = self.calc_CoordsRES_lambda_xy[i][u_i][v_i][0]
72         v = self.calc_CoordsRES_lambda_xy[i][u_i][v_i][1]
73         res=0

```

```

74     for x_i in range(\n
75         0, 2 ** self.Settings.sampling_FFT_N[0]):\n
76         # loop y\n
77         for y_i in range(\n
78             0, 2 ** self.Settings.sampling_FFT_N[1]):\n
79             x = self.calc_CoordsOPT_lambda_xy\\
80                 [i][x_i][y_i][0]\n
81             y = self.calc_CoordsOPT_lambda_xy\\
82                 [i][x_i][y_i][1]\n
83             ps = phaseshift(la, y, v, x, u)\n
84             cf=coherencefactor(la, y, v, x, u)\n
85             trami=self.calc_transmission_lambda_xy\\
86                 [i][x_i][y_i]\n
87             res+=C*numpy.fft.fftn(self.calc_Eopt_lambda_xy\\
88                 [i,x_i,y_i]*\\
89                 self.calc_transmission_lambda_xy\\
90                 [i][x_i][y_i]*ps*cf)\n
91             # maybe also put coherence constant here\n
92             self.calc_Eres_lambda_xy[i][u_i][v_i]=res\n
93             #Scaling factor\n
94             o_s=0\n
95             r_s=0\n
96             for x_i in range(0, 2 ** self.Settings.sampling_FFT_N[0]):\n
97                 for y_i in range(0, 2 ** self.Settings.sampling_FFT_N[1]):\n
98                     o_s += numpy.real(self.calc_Eopt_lambda_xy\\
99                         [i][x_i][y_i] * numpy.conjugate(\n
100                             self.calc_Eopt_lambda_xy[i][x_i][y_i]))\n
101             for u_i in range(0, 2 ** self.Settings.sampling_FFT_N[0]):\n
102                 for v_i in range(0, 2 ** self.Settings.sampling_FFT_N[1]):\n
103                     r_s += numpy.real(self.calc_Eres_lambda_xy\\
104                         [i][u_i][v_i] *numpy.conjugate(\n
105                             self.calc_Eres_lambda_xy[i][u_i][v_i]))\n
106             k = o_s / r_s\n
107             for u_i in range(0, 2 ** self.Settings.sampling_FFT_N[0]):\n
108                 for v_i in range(0, 2 ** self.Settings.sampling_FFT_N[1]):\n
109                     self.calc_Eres_lambda_xy[i][u_i][v_i]*=k

```

Diese Funktion wird parallelisiert ausgeführt und berechnet für jede Wellenlänge die Feldstärken an den jeweiligen Rasterpunkten der optischen bzw. der Schirmebene und speichert diese in der Calculation Klasse ab. Hierzu werden zunächst bis Zeile 20 die zur Berechnung nötigen Variablen belegt.

In Zeile 22 wird mithilfe der Calculate\_z\_offset()-Funktion die z-Verschiebung des Fokuspunktes berechnet (siehe entsprechender Codeabschnitt).

In Zeile 29ff werden die Koordinaten der Samplingpunkte berechnet und zwischengespeichert.

In Zeile 47ff findet dann die Berechnung der elektrischen Feldstärke in der optischen Ebene mithilfe der Calculate\_Field()-Funktion als Gaußstrahl statt (siehe entsprechender Abschnitt).

In den Zeilen 54-57 sind Hilfsfunktionen zur Berechnung in der Schirmebene definiert. Sie entsprechen den auf Seite 26 erwähnten Funktionen.

In Zeile 59ff werden die Werte der Transmissionsfunktion für alle Rasterpunkte der optischen Ebene berechnet und zwischengespeichert.

In Zeile 87ff werden die Werte der elektrischen Feldstärke in den Rasterpunkten der Schirmebene als Fouriertransformation berechnet.

In Zeile 94ff werden alle Intensitätswerte der optischen bzw. Schirmebene addiert und die beiden Ergebnisse miteinander verglichen. Hieraus wird eine Skalierung berechnet.

### Codeabschnitt z-Offset Berechnung

```

1  def Calculate_z_offset(self ,wavelength):#as iterator
2      "wavelength as iterator"
3      #get wavelength
4      l=self .calc_sampling_lambda[ wavelength ]
5      W_s=self .SourceData .source_beam_radius
6      W_0=self .SourceData .source_waistrad
7      res=(numpy .pi*W_0**2/ l )*((W_s**2/W_0**2) -1)**0.5
8      return  res

```

### Codeabschnitt Berechnung in optischer Ebene

```

1  def Calculate_Field(self ,wavelength ,x,y):#wavelength as iterator , x,y as val
2      "Calculate the Field in optical plane at x,y"
3      l=self .calc_sampling_lambda[ wavelength ]
4      E_0=self .calc_sampling_E0[ wavelength ]
5      W_0=self .calc_sampling_waistrad[ wavelength ]
6      z = OpticalElementData .oe_coordinates [2] \
7      - self .calc_sampling_offsets [ wavelength ][ 2 ]
8      #calculated stuff
9      r=(x**2+y**2)**0.5
10     k=(2*numpy .pi)/ l
11     z_r=numpy .pi*W_0**2/ l
12     W_z=W_0*(1+(z/z_r)**2)**0.5
13     R_z=z*(1+(z_r/z)**2)**0.5
14     gouy=numpy .arctan (z/z_r)
15     res=0
16     e5=E_0*(W_0/W_z)*numpy .exp(-(r/W_z)**2) \
17     *numpy .exp((-1j*k*r**2)/(2*R_z))*numpy .exp(-1j*(k*z-gouy))
18     res= complex(e5)
19     return  res

```

### 3.2.4 Verbesserungspotential

Das Programm benötigt für die Berechnung und das Plotten von 12 Wellenlängen mit je 256 Rasterpunkten in der optischen sowie Schirmebene etwa eine Minute. Während eine höhere Anzahl an Wellenlängen die Laufzeit lediglich linear erhöht, verlängert eine um den Faktor 2 erhöhte Auflösung die Laufzeit exponentiell. Insbesondere im Berechnungsschritt der Schirmebene erhöht sich der Rechenaufwand in diesem Fall um den Faktor 8. Dies lässt sich lediglich mit der Wahl eines anderen mathematischen Modells verhindern. Dennoch wäre es möglich, das Programm durch optimierte Berechnungen und weitere Parallelisierung etwas performanter zu gestalten.

So sind sogenannte list-comprehensions und Lambdafunktionen meist etwas ressourcenschonender als Schleifen und einzelne Funktionen. Zudem ist zwar die Berechnungsfunktion für jede Wellenlänge parallelisiert, um von der Leistungsfähigkeit moderner Mehrkernprozessoren zu profitieren, jedoch ist die Plot-Funktion aufgrund der Anforderung seitens matplotlib weiterhin single-threaded. Durch die Wahl einer anderen Plotting-Bibliothek oder einer mit matplotlib kompatiblen Taskingbibliothek ließe sich dieser Schritt auch parallel ausführen. Allerdings ist der Hauptrechenaufwand aufgrund der exponentiellen Abhängigkeit ( $\mathcal{O}(2^n)$ ) in der Berechnung zu finden, weshalb der Fokus der Optimierung auf performanteren Funktionen oder idealerweise auf einem mathematischen Modell ohne exponentieller Laufzeit ( $\mathcal{O}(n)$ ) liegen sollte.

Abseits der reinen Ressourcenoptimierung wurde in der Entwicklung des Programms auch Wert auf eine positive User Experience gelegt. Insbesondere auf eine leichte Bedienbarkeit mittels graphischer Benutzeroberfläche und strukturierter Ausgabe sowie der Möglichkeit, vorherige Eingaben wieder einzulesen. An dieser Stelle gibt es noch Erweiterungs- und Verbesserungspotential, um eine Verwendung in der Praxis möglichst störungsfrei zu ermöglichen. Derzeit ist die Save&Load Funktionalität noch nicht 100% zuverlässig und müsste auf Zuverlässigkeit geprüft werden. Eine weitere natürliche Erweiterung wäre es auch, bereits erledigte Berechnungen zu speichern, um Plotparameter ändern zu können, ohne eine neue Berechnung durchführen zu müssen. Diese Plots selbst könnten zudem mit verschiedenen optionalen Parametern erweitert werden, um aussagekräftige Graphiken direkt aus dem Programm zu erstellen.

Die Graphische Benutzeroberfläche gibt in dieser Ausbaustufe einfache Informationen zum Status des Programms heraus. Bei Berechnungen mit hoher Auflösung wirkt es jedoch so, als ob das Programm abgestürzt wäre. Lediglich in der Konsolenausgabe ist erkennbar, dass es weiterhin läuft. Dementsprechend wäre eine erweiterte Ausgabe des aktuellen Status der parallelisierten Berechnungen innerhalb der GUI wünschenswert. Zudem wäre eine Abbruchmöglichkeit für den Berechnungsschritt im Falle ungewünschter Eingaben von Vorteil.

Ferner wäre ein möglicher Ausbauschritt die GUI anzupassen, um ein direktes Begutachten der erzeugten Daten innerhalb dieser zu erlauben. Dieses Feedback würde dann eine schnellere Iteration verschiedener Parameteränderungen erlauben. Zudem wäre damit eine Verwendung in der Lehre möglich, da es die Effekte von Parameteränderungen anschaulich demonstriert und damit das Verständnis der einzelnen Parameter fördert.

## 4 Beispielberechnungen

Die folgenden Berechnungen zeigen den Einfluss eines Phasengitters auf die erzeugte Bildfigur. Um die Anwendungsmöglichkeiten des Programms zu demonstrieren, werden an diesem verschiedene Beispielberechnungen mit unterschiedlichen Parametern durchgeführt. Im Vergleich zu den bereits in den vorherigen Abschnitten durchgeführten Berechnungen wurden folgende Parameter angepasst:

- **Erhöhte Auflösung**

Die Auflösung der Ebenen wurde von 256 Rasterpunkten (16x16) auf 1024 Rasterpunkte (32x32) erhöht, um Interferenzen sichtbarer zu machen.

- **Transmissionsfunktion**

Als Transmissionsfunktion wurde ein in der x-Koordinate wirkendes, amplitudenerhaltenes Phasengitter gewählt  $e^{i \cdot \sin(\frac{3 \cdot 10^8 \cdot x}{\lambda_z})}$ , zudem wurde die Lichtgeschwindigkeit für diesen Fall mit  $3 \cdot 10^8 \text{ m/s}$  genähert. Bei  $\lambda_z$  handelt es sich um die zentrale Wellenlänge des Gitters und damit um eine Materialkonstante des Gitters. Es wurden drei Vergleichsrechnungen mit einer zentralen Wellenlänge von 325nm, 650nm und 975nm durchgeführt.

- **Kohärenzlängen**

Für die Berechnungen wurden unterschiedliche Kohärenzlängen verschiedener Größenordnungen zugrunde gelegt. Diese reichen von einer, für eine Leuchtdiode typischen, Kohärenzlänge von 4 Mikrometern bis zu einer Kohärenzlänge von 1km (Helium-Neon-Laser).

- **Spektralfunktion**

Für die in Abbildung 15 abgetastete Spektralfunktion wurden als Vergleichsrechnungen jeweils um  $\pm 325\text{nm}$  verschobene Spektralfunktionen zugrunde gelegt.

Für die Beispielberechnungen werden jeweils vier Graphiken ausgegeben, in der Reihenfolge von oben links nach unten rechts sind diese:

- Der **Intensitätsplot** der fünften gesampleten Wellenlänge in der **optischen Ebene**
- Der **Intensitätsplot** der fünften gesampleten Wellenlänge in der **Schirmebene**
- Die gesampte **Spektralfunktion**
- Das **Gesamtbild** in der Schirmebene

Aufbauend auf den bereits in vorherigen Abschnitten erwähnten Beispielberechnungen wurde nun zunächst ein Fall festgelegt, mit dem die jeweiligen Variationen verglichen werden sollen. Die zugrunde gelegten Vergleichsparameter sind:

- **zentrale Wellenlänge der Transmissionsfunktion**

$$\lambda_z = 650\text{nm}$$

- **Kohärenzlänge**

$$L_c = 1\text{cm}$$

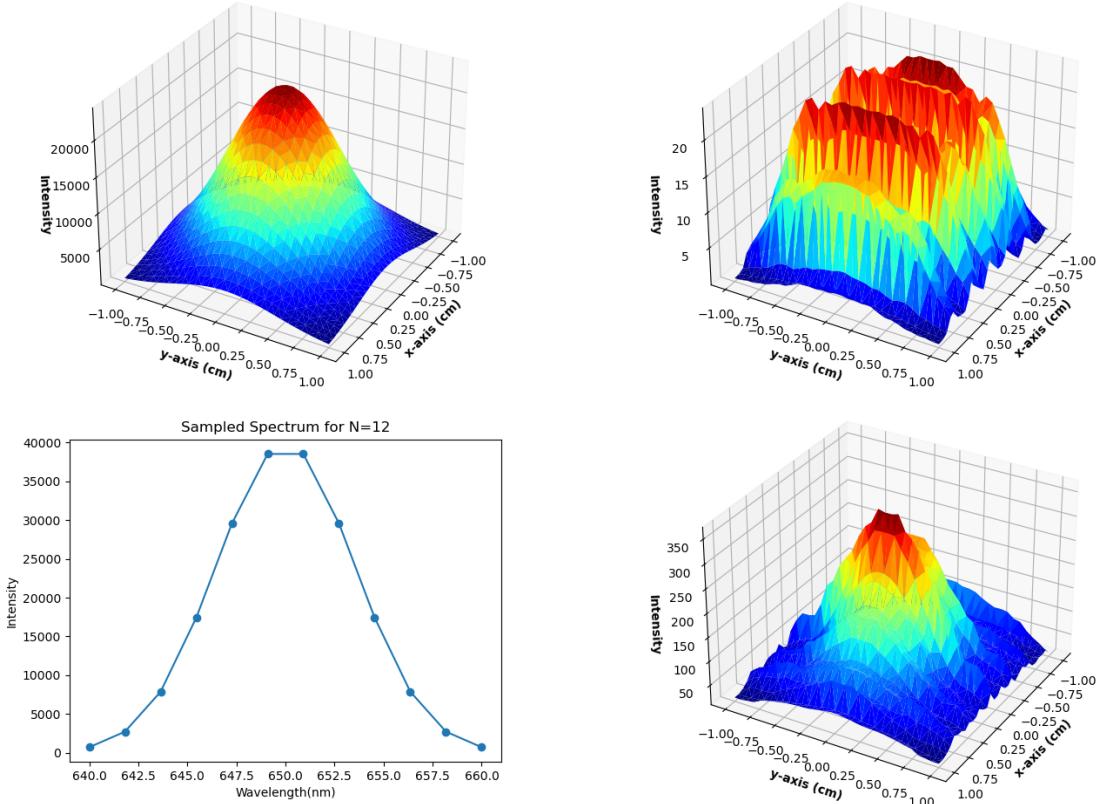


Abbildung 24: Plots der Vergleichsberechnung  $\lambda_z=650\text{nm}$ ,  $L_c=1\text{cm}$

Anhand dieser Plots ist gut zu erkennen, dass das Phasengitter ein in x moduliertes und in y ausgedehntes streifenförmiges Interferenzmuster erzeugt und das 0te Maximum um etwa 0.5 cm in negative x-Richtung verschoben ist. Ferner ist auch das gaußförmige Eingangssignal weiterhin gut im Ausgangssignal zu erkennen, ist aber aufgrund der räumlichen Kohärenz etwas verschwommen.

#### 4.1 Einfluss der Kohärenzlänge

In den folgenden Berechnungen wurde im Vergleich zur Beispielberechnung (Abb. 24) jeweils die Kohärenzlänge der Quelle variiert. Die betrachteten Kohärenzlängen erstrecken sich über mehrere Größenordnungen und sind aufsteigend sortiert:

4μm (typisch für eine LED), 0.1mm, 1mm, 1dm, 1m, 1km (Helium-Neon Laser)

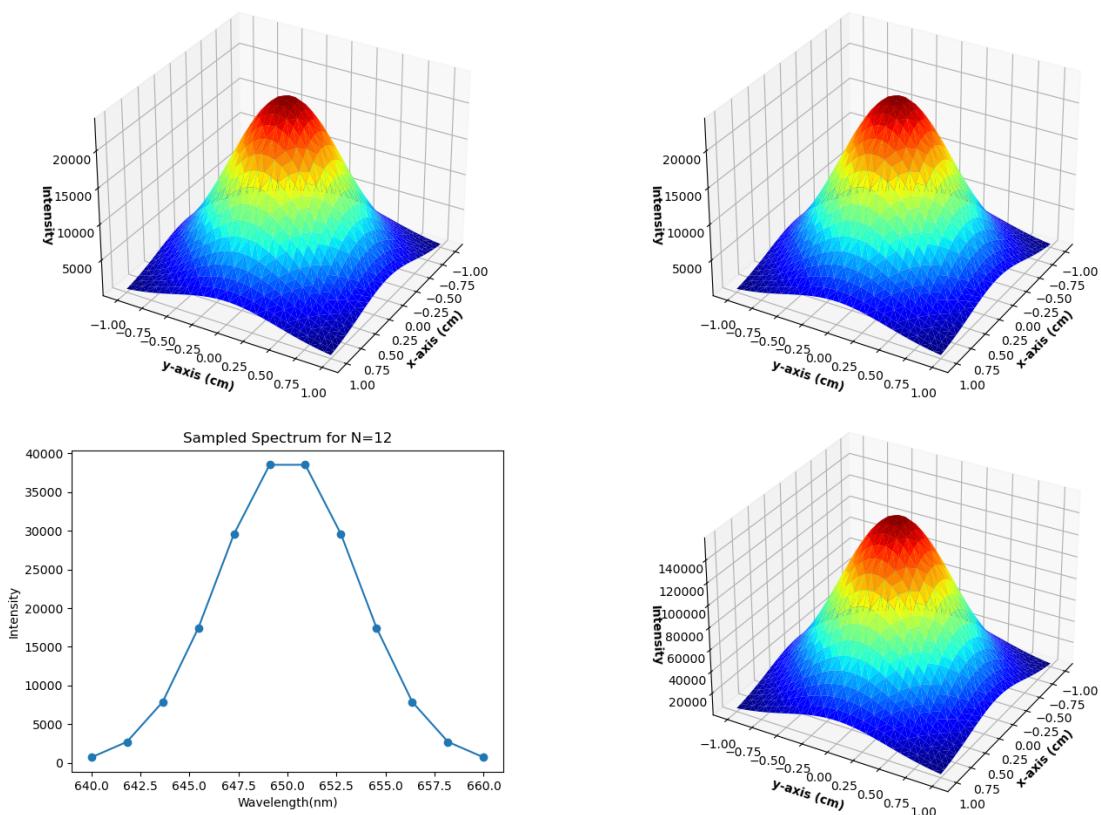


Abbildung 25: Plots der Vergleichsberechnung mit  $L_c=4\mu\text{m}$

Diese Graphiken machen deutlich, dass eine Koheränzlänge von 4 Mikrometern nicht ausreichend ist, um in dem betrachteten Experiment Interferenzphänomene hervorzurufen.

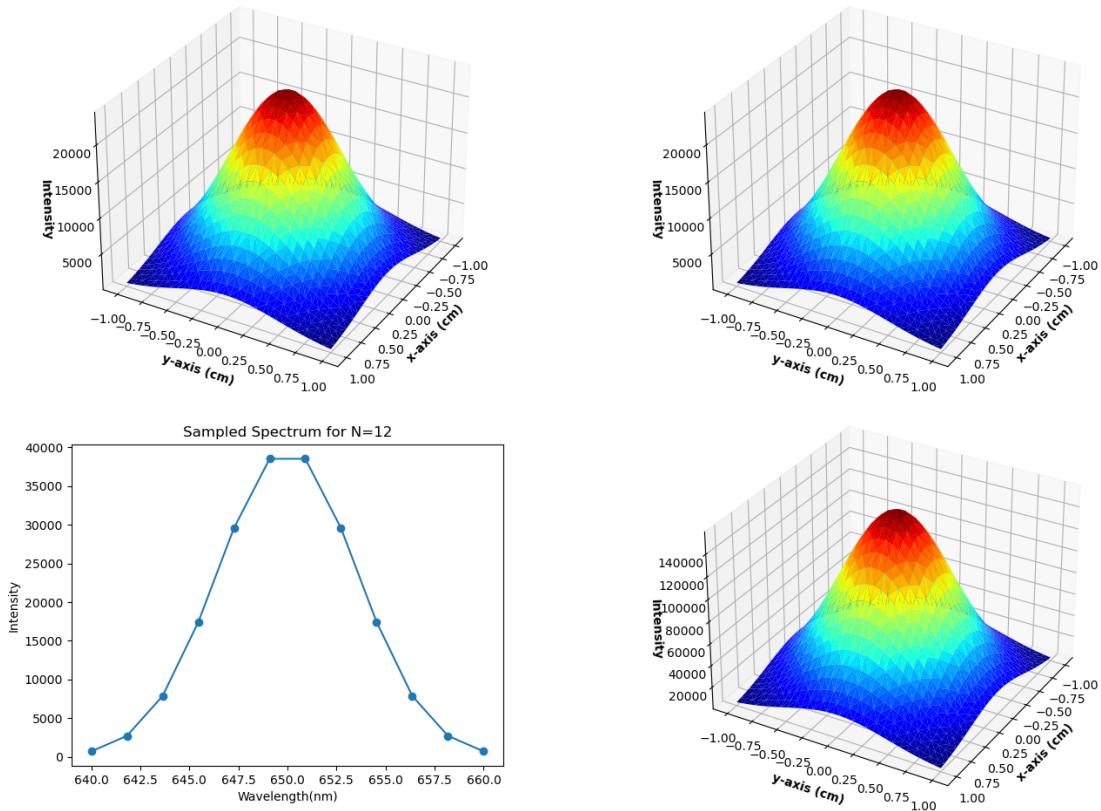


Abbildung 26: Plots der Vergleichsberechnung mit  $L_c=0.1\text{mm}$

Analog zu Abb. 25 ist auch eine Kohärenzlänge von 0.1mm nicht ausreichend, um betrachteten Aufbau Interferenzphänomene hervorzurufen.

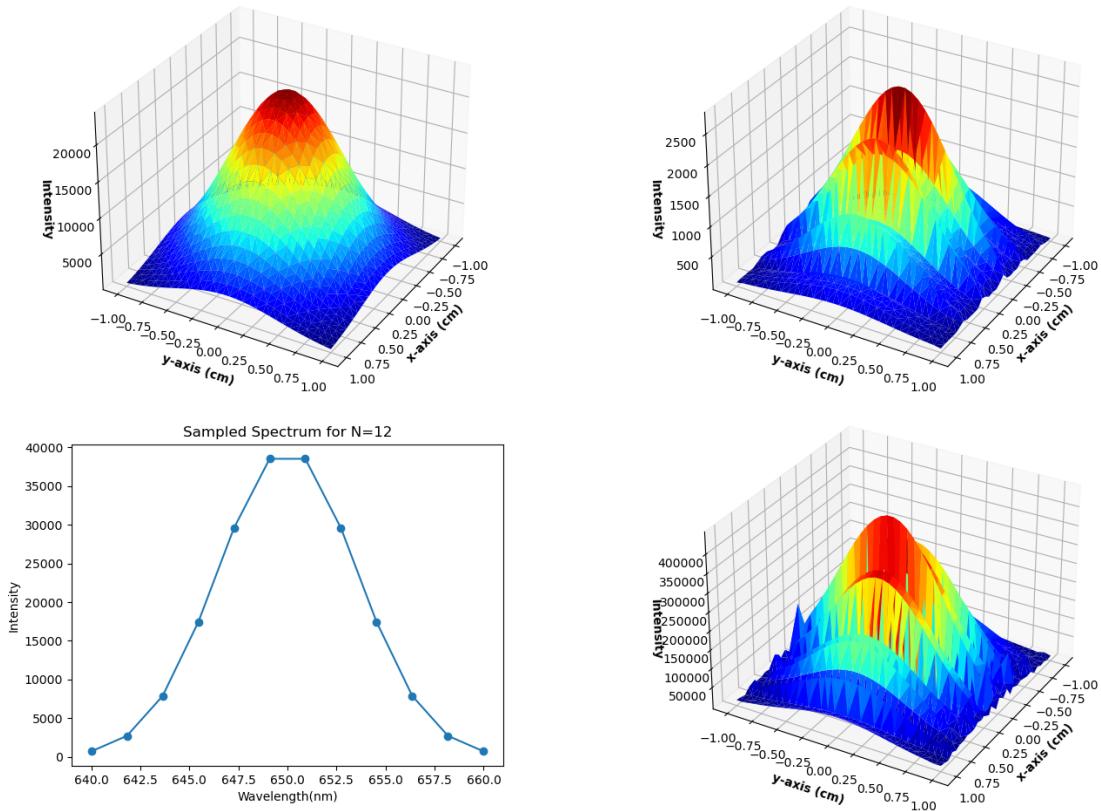


Abbildung 27: Plots der Vergleichsberechnung mit  $L_c=1\text{mm}$

Mit einer Kohärenzlänge von 1mm ist das erste Mal ein Interferenzmuster zu erkennen. Im Vergleich zur Referenz ist das Eingangssignal klarer zu erkennen, da aufgrund der kleinen Kohärenzlänge keine räumliche Kohärenz stattfindet, sondern lediglich Interferenz in x-Richtung auftritt.

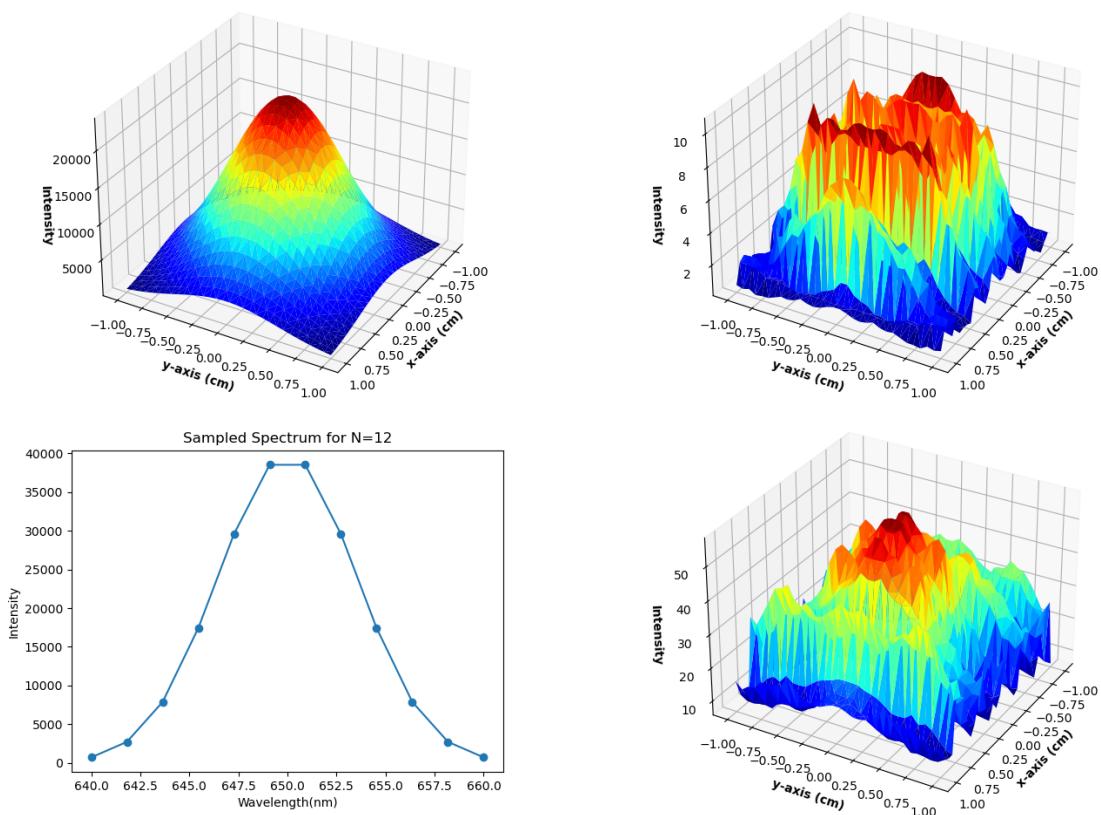


Abbildung 28: Plots der Vergleichsberechnung mit  $L_c=1\text{dm}$

Mit einer Kohärenzlänge von 1dm ist aufgrund der größeren Kohärenzlänge eine erhöhte räumliche Verwischung des Interferenzmusters zu erkennen.

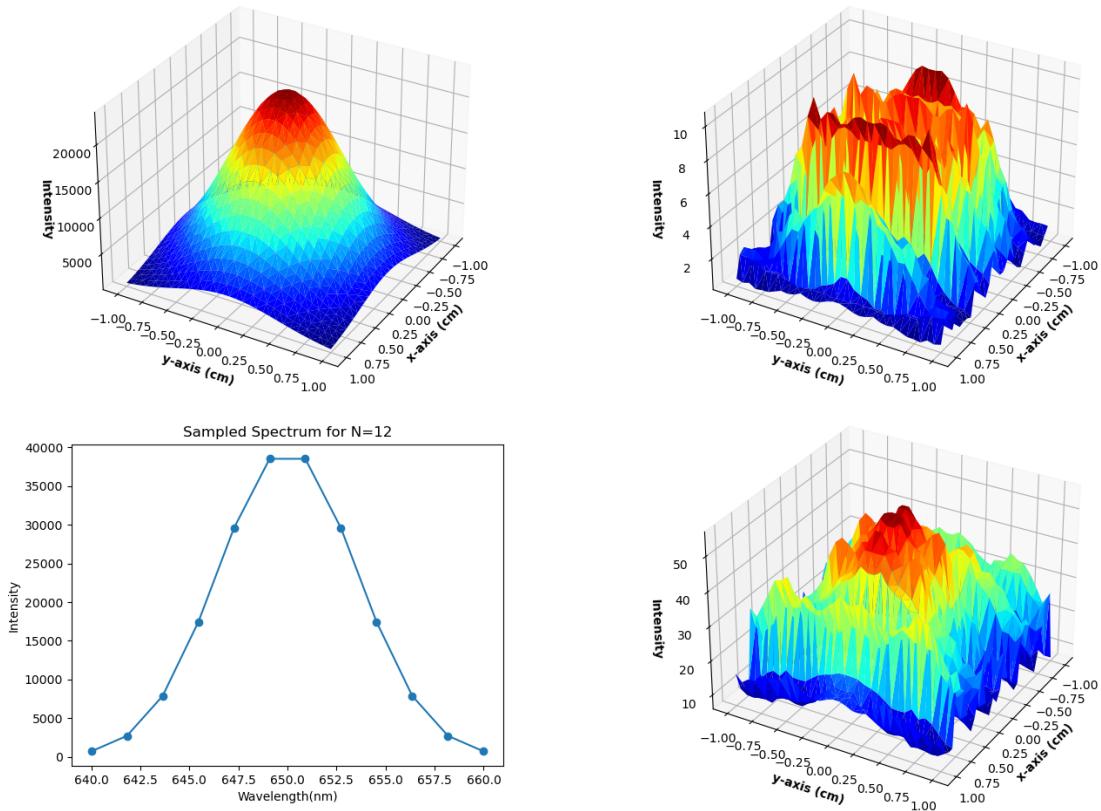


Abbildung 29: Plots der Vergleichsberechnung mit  $L_c=1\text{m}$

Im Vergleich zu Abbildung 28 ist bei einer weiteren Erhöhung der Kohärenzlänge kein Effekt mehr zu erkennen, da bereits die in Abbildung 28 betrachtete Kohärenzlänge von 1dm ausreichte, um alle Rasterpunkte miteinander interferenzfähig zu machen.

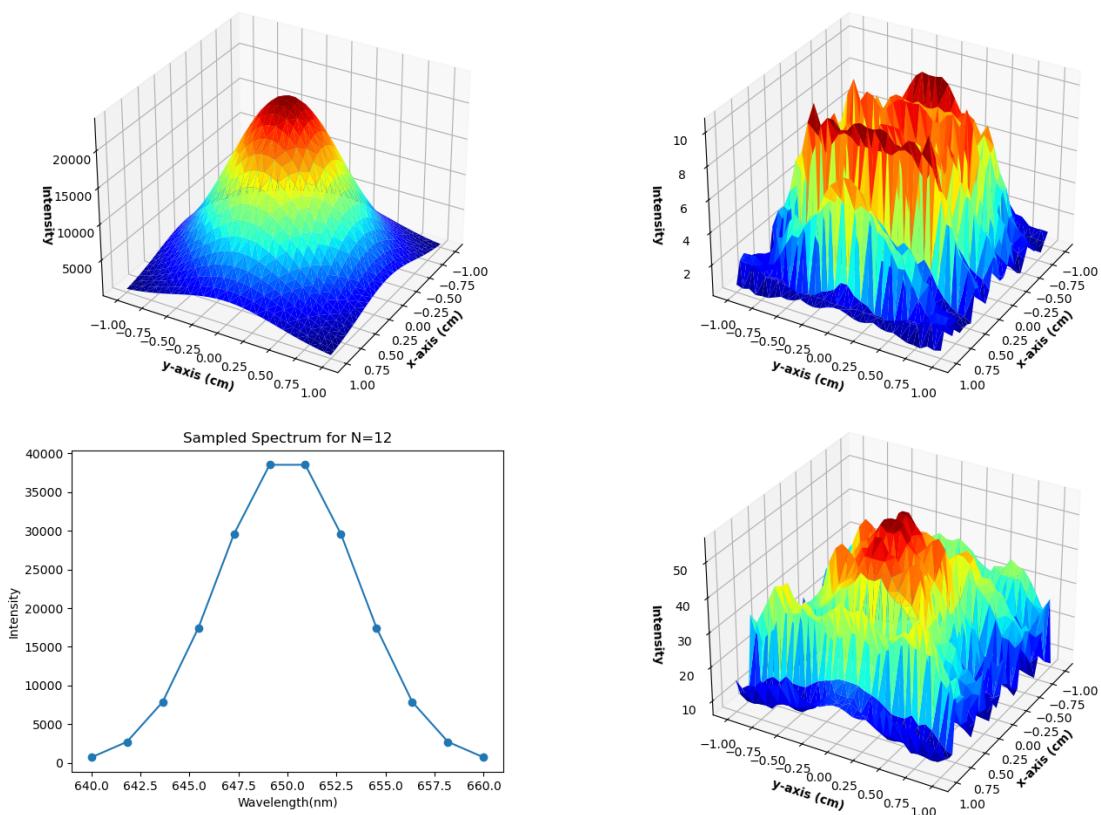


Abbildung 30: Plots der Vergleichsberechnung mit  $L_c=1\text{km}$

Wie bereits in Abbildung 29 erklärt, ist bei der für einen Helium-Neon Laser üblichen Kohärenzlänge von einem Kilometer keine Veränderung im Vergleich zu Abbildung 28 erkennbar.

## 4.2 Variation der Transmissionsfunktion

An dieser Stelle wird nun im Vergleich zu Abbildung 24 die zentrale Wellenlänge des Phasengitters variiert.

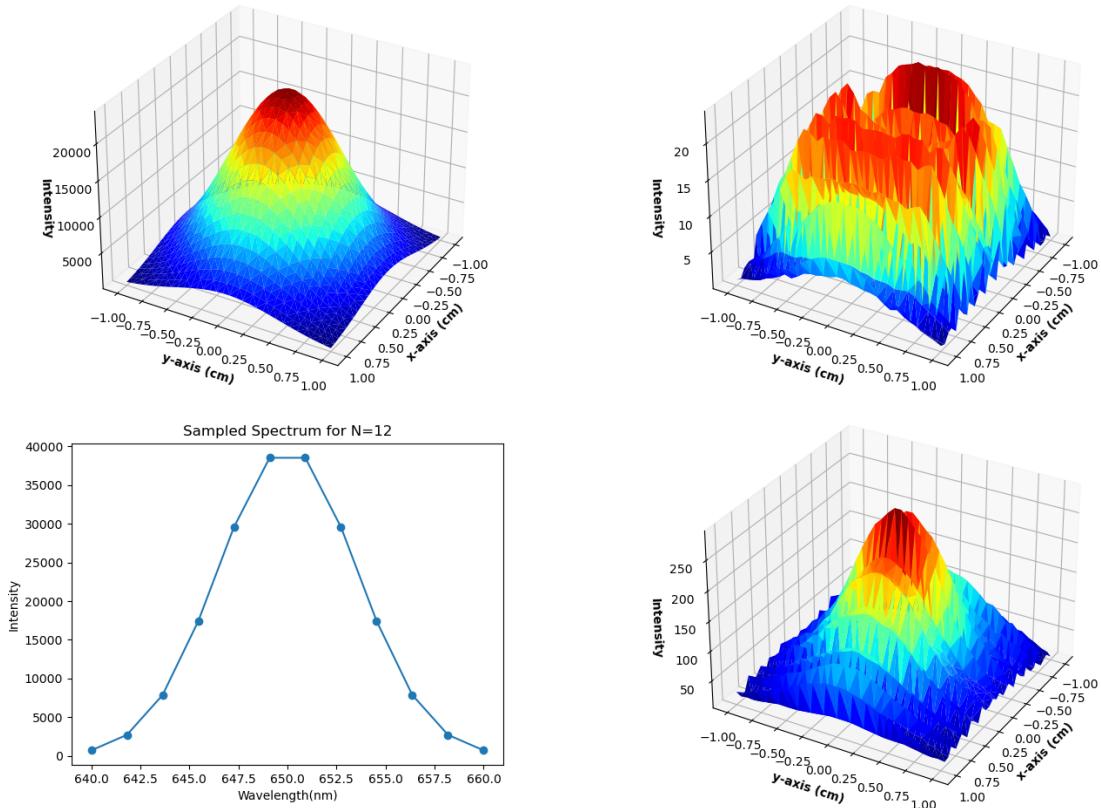


Abbildung 31: Plots der Vergleichsberechnung mit  $\lambda_z = 325\text{nm}$

Im Vergleich zu dem mit  $\lambda_z = 650\text{nm}$  gerechneten Vergleichsplot sind an dieser Stelle mehr Maxima zu erkennen. Dies deckt sich mit der Erwartung der Verdopplung, da die Vergleichstransmissionsfunktion  $e^{i \cdot \sin(\frac{3 \cdot 10^8 \cdot x}{\lambda_z})}$  zu  $e^{i \cdot \sin(\frac{3 \cdot 10^8 \cdot 2x}{\lambda_z})}$  umgeformt werden kann.

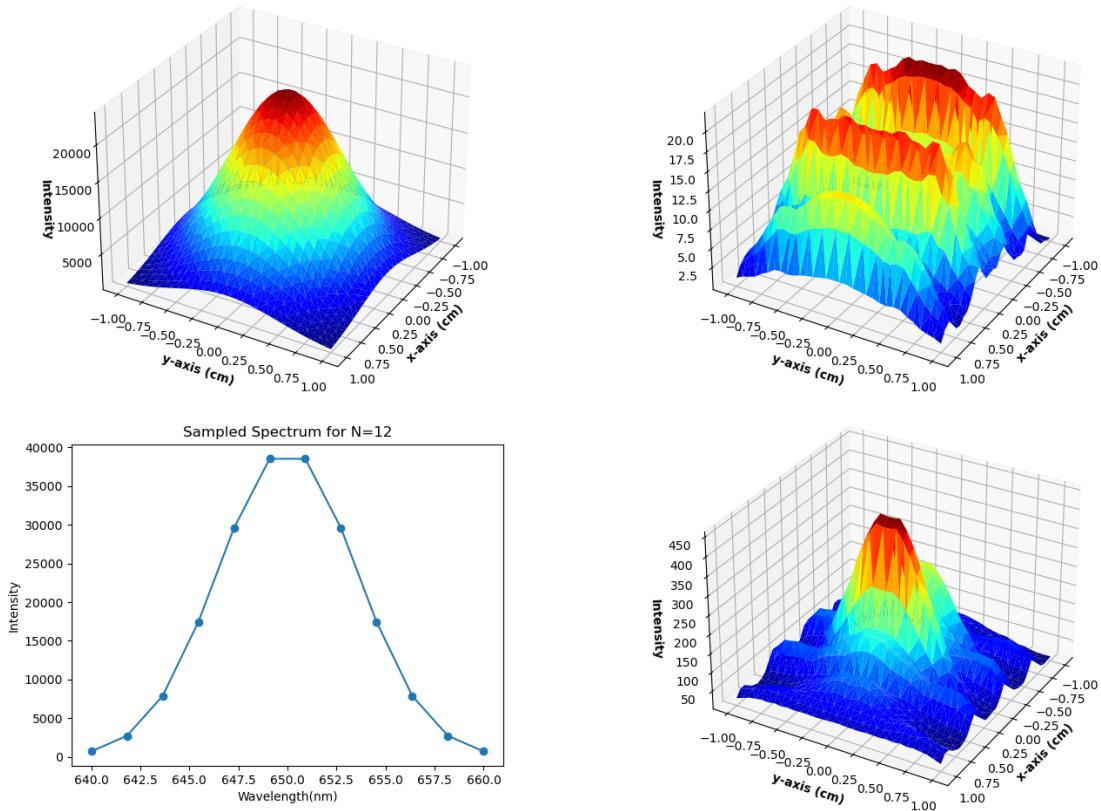


Abbildung 32: Plots der Vergleichsberechnung mit  $\lambda_z=975\text{nm}$

Analog zu Abbildung 31 sind an dieser Stelle entsprechend weniger Maxima zu erkennen.  
 Dies deckt sich mit der Erwartung, da die Vergleichstransmissionsfunktion  $e^{i \cdot \sin(\frac{3 \cdot 10^8 \cdot x}{3 \lambda_z})}$  zu  $e^{i \cdot \sin(\frac{3 \cdot 10^8 \cdot 2 \cdot \frac{3}{2} x}{\lambda_z})}$  umgeformt werden kann.

### 4.3 Variation des Lichtspektrums

An dieser Stelle wird im Vergleich zu den in Abbildung 24 betrachteten Plots das Spektrum des einfallenden Lichtes variiert. Hierbei ist zu beachten, dass die betrachteten fünften Wellenlängen, im Gegensatz zu den vorherigen Betrachtungen, nun nicht mehr etwa 647.27nm, sondern etwa 322.27nm respektive 922.27nm entsprechen.

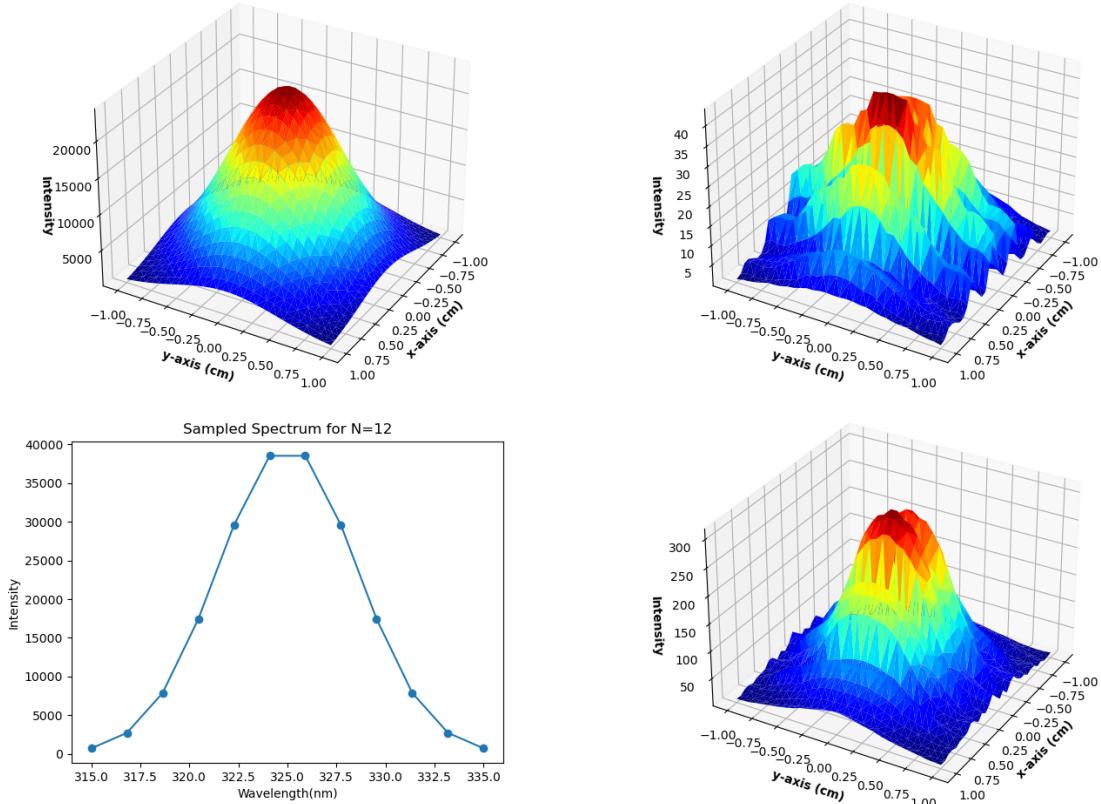


Abbildung 33: Plots der Vergleichsberechnung mit um -325nm verschobenem Lichtspektrum

Wird das Spektrum der Lichtquelle geändert, so ändern sich, aufgrund des unterschiedlichen z-offsets, zunächst einmal auch die Intensitäten der jeweiligen Wellenlängen in der optischen Ebene (oben links). Ein Blick auf den entsprechenden Plot zeigt allerdings, dass dieser Effekt zu vernachlässigen ist.

Wird die Feldstärke in der optischen Ebene als gleich angenommen, so ändert sich in der Berechnung der Intensitätsverteilung in der Schirmebene lediglich die Phasenkorrektur aufgrund unterschiedlicher Laufzeiten (siehe S. 26). Diese bewirkt eine Koordinatenverschiebung, was sich im Plot als Streckung bemerkbar macht.

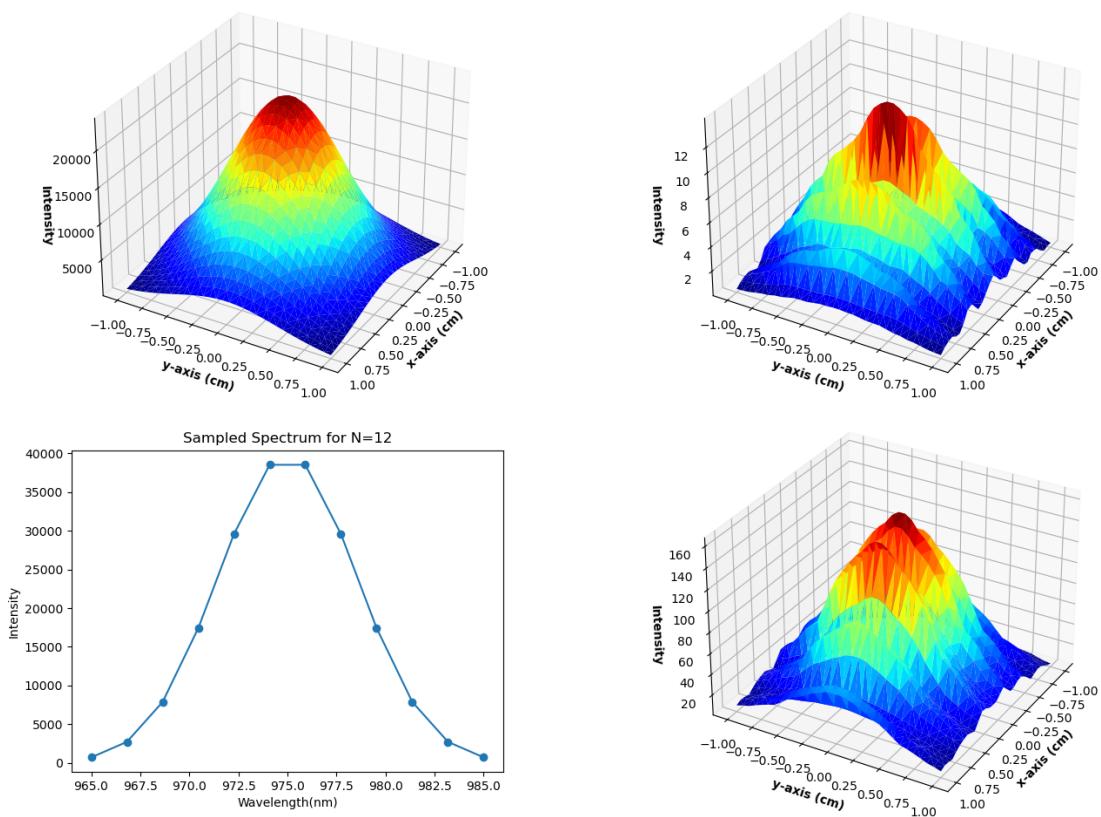


Abbildung 34: Plots der Vergleichsberechnung mit um +325nm verschobenem Lichtspektrum

Analog zu Abb. 33 macht sich die Änderung des Spektrums in diesem Fall als Stauchung bemerkbar.

## 5 Fazit

Diese Arbeit stellt dem Anwender ein mathematisch-physikalisches Modell zur Simulation von Strahlpropagation und Superposition teilkohärenter Wellen zur Verfügung. Auf Basis dieses Modells wurde ein praxisorientiertes und leistungsfähiges Programm entwickelt, welches in der Lage ist, den Effekt diverser diffraktiver optischer Elemente unter verschiedenen teilkohärenten Lichtquellen kontinuierlichen Spektrums zu simulieren. Gemäß des Anforderungsprofils wurde das Programm in Python 3 mit frei verfügbaren wissenschaftlichen Bibliotheken und damit unabhängig von Lizenzgebühren anwendbar und wartbar umgesetzt.

Das mathematisch-physikalische Modell wird zunächst anhand eines Beispielablaufes des Programms näher erklärt. Ein besonderer Augenmerk wurde, neben den nötigen Anforderungen, auf eine hohe Anwenderfreundlichkeit gelegt. So lassen sich die für die Berechnung notwendigen Parameter, wie die zugrundeliegende Transmissionsfunktion oder das Lichtspektrum der Quelle, in einer übersichtlichen graphischen Benutzeroberfläche einstellen. Die Bedienung dieser Benutzeroberfläche, sowie alle Dateiausgaben des Programms sind ausführlich und anhand von Beispielen erklärt.

Um den Nutzen dieses Programms in der Forschungsanwendung zu demonstrieren, wurde ein für die Anwendung repräsentativer Aufbau mit einem Phasengitter unter verschiedenen Kohärenzlängen, Lichtquellen und Transmissionsfunktionen berechnet.

Die aufbereiteten Beispielberechnungen illustrieren die Vorzüge des Programms sehr deutlich. Das Programm erlaubt es, die Effekte von (kleinen) Parameteränderungen durch eine schnelle Abfolge von Simulationen zu verdeutlichen und bietet damit eine Möglichkeit, passende Einstellungen bereits vor der Durchführung eines realen Experiments zu eruieren.

Ausgehend von der jetzigen Ausbaustufe des Programms sind mehrere Erweiterungsmöglichkeiten realisierbar. Verbesserungen hinsichtlich der Performance des Programms sind allerdings in Anbetracht der Tatsache, dass das Modell eine exponentielle asymptotische Laufzeit zugrunde legt, nur beschränkt möglich. Es sollten daher die User-Experience, Zuverlässigkeit und Flexibilität im Fokus zukünftiger Erweiterungen liegen.

Ein vielversprechender Ansatz ist es, die GUI um direkte Ausgaben der erzeugten Plots sowie weitere Parameter für die Erstellung aussagekräftiger Graphiken zu erweitern. Dies würde sowohl schnellere Iterationen in der Anwendung als auch eine Anschlussverwendung in der Lehre ermöglichen.

## 6 Quellen

### Auflistung der Quellen

- [1] Okan K. Ersoy. *Diffraction, Fourier optics and Imaging*. Verlag John Wiley & Sons, Inc., 2007.
- [2] Miles Klein Thomas E. Furtak. *Optik*. Springer-Verlag Berlin Heidelberg, 1988.
- [3] C. Huygens. *Traite de la lumiere: Où sont expliquées les causes de ce qui luy arrive dans la reflexion, & dans la refraction. Et particulierement dans l'étrange refraction du cristal d'Islande*. Chez Pierre vander Aa, 1690.
- [4] Othmar Marti. „Vorlesungsskript PHYS2000.0 Optik“. In: (2015).
- [5] Prof. Dr. Lukas Novotny. „Lecture Notes on ELECTROMAGNETIC FIELDS AND WAVES“. In: (2013).
- [6] Donald C. O’Shea. *Diffractive Optics Design, Fabrication, and Test*. Tutorial Texts in Optical Engineering, Volume TT62, SPIE PRESS, 2004.
- [7] O. Benson & A. Peters. *Maxwellsche Gleichungen*. 2010.
- [8] *Skript zum Versuch Organische Leuchtdioden*. 2009.
- [9] Dabid S. Tannhouse Stephen G. Lipson Henry S. Lipson. *Optik*. Springer, 1997.
- [10] *Versuch zu Gaußscher Strahlenoptik*. 2011.
- [11] W Lauterborn T Kurz M. Wiesenfeldt. *Kohärente Optik*. Springer, 1993.
- [12] T. Young. *II. The Bakerian Lecture. On the theory of light and colours*. Phil. Trans. R. Soc.9212–48, 1802.

## 7 Abbildungsverzeichnis

### Abbildungsverzeichnis

1	Huygens' Wellenfrontbeschreibung . . . . .	3
2	Elementarwellen: Young'sches Experiment . . . . .	6
3	reale Blende . . . . .	7
4	Prisma und Blazegitter . . . . .	8
5	Dammann-Gitter . . . . .	9
6	Hocheffizientes Dammann-Gitter . . . . .	9
7	Beugungsgeometrie . . . . .	13
8	Michelson Interferometer . . . . .	14
9	Youngscher Doppelspaltversuch . . . . .	16
10	Youngscher Doppelspaltversuch(inkohärent) . . . . .	18
11	Winkelbezeichnungen räuml. Kohärenz . . . . .	18
12	Spektrum einer Weißlicht-LED . . . . .	20
13	Gaußstrahl . . . . .	20
14	Strahlpropagation . . . . .	23
15	Intensitätsspektrum . . . . .	24
16	Intensitätsplot in opt. Ebene . . . . .	25
17	Intensitätsplot in Schirmebene . . . . .	26
18	Gesamtintensitätsplot in Schirmebene . . . . .	27
19	GUI-Calculation . . . . .	28
20	GUI-Settings . . . . .	29
21	GUI-Optical Element . . . . .	30
22	GUI-Source . . . . .	30
23	Ordnerstruktur Dateiausgabe . . . . .	31
24	Plots der Vergleichsberechnung . . . . .	39
25	Vergleichsplot $L_c=4\mu\text{m}$ . . . . .	40
26	Vergleichsplot $L_c=0.1\text{mm}$ . . . . .	41
27	Vergleichsplot $L_c=1\text{mm}$ . . . . .	42
28	Vergleichsplot $L_c=1\text{dm}$ . . . . .	43
29	Vergleichsplot $L_c=1\text{m}$ . . . . .	44
30	Vergleichsplot $L_c=1\text{km}$ . . . . .	45
31	Vergleichsplot $\lambda_z=325\text{nm}$ . . . . .	46
32	Vergleichsplot $\lambda_z=975\text{nm}$ . . . . .	47
33	Vergleichsplot Spektrum um 325nm . . . . .	48
34	Vergleichsplot Spektrum um 975nm . . . . .	49

## **Erklärung der Urheberschaft**

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum

Unterschrift