Geladenes Teilchen in starken elektromagnetischen Feldern

Bachelor-Arbeit

vorgelegt am: 27. November 2010

am Theoretisch-Physikalischen Institut der Physikalisch-Astronomischen Fakultät der Friedrich Schiller Universität Jena

Name:	Tobias Hellwig
Matrikelnummer:	97069
Studienjahrgang:	2007
Erstgutachter:	Prof. Dr. Andreas Wipf
Zweitgutachter:	Prof. Dr. Gerhard Schäfer

Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Jena, 23. September 2010 Tobias Hellwig

Seitens des Verfassers/Verfasserin bestehen keine Einwände, die vorliegende Bachelorarbeit für die öffentliche Nutzung in der Thüringer Universitäts- und Landesbibliothek zur Verfügung zu stellen.

Jena, 23. September 2010 Tobias Hellwig

Inhaltsverzeichnis

1 Einführung

Teilchenbeschleuniger nehmen eine große Rolle in der Verifizierung von modernen physiklaischen Theorien, insbesondere in der Elementarteilchenphysik ein. Deshalb werden immer höhere Energien und somit auch größere Ringbeschleuniger benötigt. Zwar konnte mit dem LHC im CERN ein großer Beschleuniger realisiert werden, aber die Ausmaße sind schon jetzt riesig und das System sowohl teuer als auch anfällig für Fehler. Aus diesem Grund besteht ein Interesse an neuen Möglichkeiten um geladene Teilchen auf hohe Energien zu beschleunigen. Die folgende Arbeit soll sich mit einem vielversprechenden Ansatz beschäftigen. Statt mit homogenen Felder zu beschleunigen, sollen elektromagnetische Wellen verwendet werden, in die ein Teilchen eingebracht wird, und dass dadurch einen effektiven Energiegewinn erfährt. Das Auskoppeln eines beschleunigten Teilchens kann dann mittels eines homogenen Magnetfeldes realisiert werden. Der einfachen Idee stehen gewisse Probleme entgegen, die die Suche nach geeigneten Feldern nicht trivial werden lassen. Zum einen das Lawson-Woodward Theorem, das uns sagt, dass wir bis auf Effekte durch die pondomotorische Kraft keinen Energiegewinn verzeichnen können, wenn das Vektorpotenzial am Anfangs- und Endpunkt den gleichen Wert hat. Hier ist zu beachten, dass der gleiche Wert gefordert wird und nicht etwa ein Unterschied um eine additeve Konstante erlaubt ist. Des weiteren ist für die praktische Anwendung die Realisierbarkeit zu beachten. Das heißt, es muss neben der Erfüllung der Maxwell Gleichung auch untersucht werden, ob eine solche Welle Forderungen nach einer begrenzten Menge an transportierter Leistung erfüllt. Wir wollen uns vom Punkt der praktischen Realisierung aber nur bedingt leiten lassen. So werden wir zunächst den nur in guter Approximation praktisch realisierbaren Fall einer ebenen Welle untersuchen. Wir werden sehen, dass in Übereinstimmung mit obigen Theorem kein Energiegewinn möglich ist. Danach betrachten wir einen neuen Ansatz zur Beschreibung eines Gaußstrahls und untersuchen die Möglichkeit, ein Teilchen zu beschleunigen. Wir werden sehen, dass hier die Möglichkeit zur Beschleunigung gegeben ist, sofern man die richtigen Parameter wählt. Abschließend sollen Besselwellen im Allgemeinen konstruiert werden, um dann in paraxialer Approximation zwei Spezialfälle zu untersuchen. Besselwellen konnten schon erzeugt werden und besitzen interessante Eigenschaften. Es wird sich zeigen, dass ein spezieller paraxialer Grenzfall existiert, bei dem die Komponente der elektrischen und magnetischen Feldstärke in Ausbreitungsrichtung der Welle nicht verschwindet, was uns Hoffnung auf eine effektive Beschleunigung über mehrere Perioden macht.

2 Grundlegende Formeln

Da es sich bei Laserstrahlen um elektromagnetische Wellen handelt, haben sie den Maxwellgleichungen zu folgen. Diese wollen wir uns in einem für unsere Zwecke geeignetem Einheitensystem, den Heaviside-Lorentz-Einheiten, notieren.

$$\nabla \times \vec{H} = \frac{1}{c}\vec{j} + \frac{1}{c}\frac{\partial \vec{D}}{\partial t} \tag{2.1a}$$

$$\nabla \times \vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t} \tag{2.1b}$$

$$\nabla \cdot \vec{D} = \rho \tag{2.1c}$$

$$\nabla \cdot \vec{B} = 0 \tag{2.1d}$$

Dabei werden wir für unsere Untersuchungen im Vakuum sowohl den Strom $\vec{j}$ als auch die Quelle ρ null setzen. Das heißt, wir betrachten das im Laserfeld vorhandene Teilchen als reines Testteilchen, welches keine Rückkopplung mit der elektromagnetischen Welle zeigt. Im Vakkuum vereinfacht sich der Satz von Materialgleichungen, in unseren Einheiten, zu folgender einfacher

Form, die es uns zukünftig erlauben unsere Betrachtungen auf $\vec{E}$ und $\vec{B}$ zu beschränken.

$$\vec{D} = \vec{E} \tag{2.2a}$$

$$\vec{H} = \vec{B} \tag{2.2b}$$

Im Folgendem werden wir uns verschiedener Lösungsansätze bedienen, um eine analytische Beschreibung für $\vec{E}$ und $\vec{B}$ zu erhalten. Unsere Bewegungsgleichung werden wir aber konsequent in einer speziell-relativistischen Beschreibung lösen und dabei folgende Metrik bei kartesischen Raumkoordinaten verwenden.

$$g = diag(1, -1, -1, -1)$$

Da die Bewegung eines geladenen Teilchens untersucht werden soll, brauchen wir noch die Bewegungsgleichung. Diese ergibt sich unter Einführung der Eigenzeit $\tau = \int_0^t \sqrt{1 - \frac{v(t')^2}{c^2}} dt'$ zu:

$$\frac{du^\mu}{d\tau} = \frac{e}{mc} F^\mu{}_\nu u^\nu \tag{2.3}$$

Wobei sich die Komponenten von $F^\mu{}_\nu$ auf folgende Weise ergeben

$$F^\mu{}_\nu = \partial^\mu A_\nu - \partial_\nu A^\mu = \begin{pmatrix} 0 & E_x & E_y & E_z \\ E_x & 0 & B_z & -B_y \\ E_y & -B_z & 0 & B_x \\ E_z & B_y & -B_x & 0 \end{pmatrix} \tag{2.4}$$

Des weiteren gilt $\frac{dx}{d\tau} = cu$ mit $x = (ct, x, y, z)^t$. Die Komponenten von u sind nicht unabhängig, sondern es gilt folgender Zusammenhang zwischen u^0 und den anderen u^i, wobei in unserer Notation lateinische Buchstabe von eins bis drei, griechische Buchstaben von null bis drei laufen:

$$u^0 = \sqrt{1 + (u^1)^2 + (u^2)^2 + (u^3)^2} \tag{2.5}$$

Wie bekannt unterliegt A^μ einer gewissen Eichfreiheit, da die physikalisch relevanten Felder $\vec{E}$ und $\vec{B}$ in folgender Weise aus A^μ hervorgehen:

$$E_i = -\partial_i A_0 - \frac{1}{c} \partial_t A_i \tag{2.6a}$$

$$B_i = \epsilon_{ijk} \partial_j A_k \tag{2.6b}$$

Aufgrund dieser Eichfreiheit wurden verschiedene Arten der Eichung mit speziellen Namen versehen. Im folgenden werden wir zwei verschiedene Eichungen verwenden. Zum einen die Lorenz Eichung mit $\partial_\mu A^\mu = 0$ als auch die kovariante Eichung für ebene Wellen mit $k_\mu A^\mu = 0$, die wir aus der Lorenz Eichung durch Addition eines konstanten Vektors zum Vierer-Potenzial A erhalten, wobei wir die weitere Eichfreiheit innerhalb der Lorenz Eichung ausnutzen, da die Vierer-Divergenz dieses Vektors naturgemäß verschwindet.

3 Ebene Wellen

Die ebene Welle nimmt eine Sonderrolle ein. Eine ebene Welle kann als Strom von Photonen aufgefasst werden, die einen scharf definierten Vierervektor besitzen, der über die komplette Ausbreitung konstant ist. Das heißt, wenn das Elektron nach der Interaktion mit der ebenen Welle eine andere Energie hat, als vor der Interaktion, müsste dies als eine Absorption oder Emission eines Photons mit obiger Vierergeschwindigkeit aufgefasst werden. Wir wissen aber, dass es für ein freies Elektron nicht möglich ist, ein Photon zu absorbieren, da im Schwerpunktsystem des Elektrons und Photons die Energie-Impuls-Erhaltung ein Photon mit der Energie null verlangt

und somit den Prozess nicht erlaubt.
Für die Lösung der Bewegungsgleichung im Fall einer ebenen Welle werden Erhaltungssätze ausgenutzt. Die Herleitung orientiert sich dabei an dem Skript zu einem Vortrag von Professor A. Wipf[1].

Eine ebene Welle ergibt sich, wenn gilt $A^\mu(x)=A^\mu(k\cdot x)$. Dabei gilt $x_\mu=(ct,x,y,z)^t$ und k ist der lichtartige Wellenzahlvektor. Wir nutzen noch die Möglichkeit der kovarianten Eichung, so dass

$$k_\mu A^\mu = 0 \tag{3.1}$$

gilt. Die Bewegungsgleichung hat folgende Form:

$$\frac{du^\mu}{d\tau} = \frac{e}{mc}F^\mu{}_\nu u^\nu = \frac{e}{mc}\left(u^\nu\partial^\mu A_\nu - u^\nu\partial_\nu A^\mu\right) \tag{3.2}$$

Führen wir jetzt das dimensionslose Potenzial $a^\mu = \frac{eA^\mu}{mc^2}$ ein und bringen den Subtrahend der rechten Seite auf die linke Seite können wir die Gleichung auf folgende Weise schreiben:

$$\frac{d}{d\tau}(u^\mu + a^\mu) = cu^\nu\partial^\mu a_\nu = cu^\nu k^\mu \frac{da_\nu}{d(k\cdot x)} = cu^\nu k^\mu a'_\nu \tag{3.3}$$

Überschiebt man nun Gleichung 3.3 mit k_μ ergibt sich aufgrund unserer Eichung $k_\mu A^\mu = 0$ und der Eigenschaft, dass k ein lichtartiger Vierervektor ist:

$$\frac{d(k_\mu u^\mu)}{d\tau} = 0 \quad \Rightarrow \quad k\cdot u = \frac{\Omega}{c} = const \tag{3.4}$$

Überschieben wir 3.3 mit einem Vektor ξ senkrecht auf k, d.h. $\xi\cdot k = 0$, so finden wir die Erhaltungsgrösse:

$$\xi\cdot(u+a) = \xi\cdot(u_0+a_0) = const \tag{3.5}$$

Des weiteren werden zwei lichtartige Vektoren n_1, n_2 sowie zwei raumartige Vektoren ϵ_1 und ϵ_2 mit folgenden Eigenschaften definiert:

$$n_i^2 = 0, \quad n_1\cdot n_2 = 2, \quad k = \frac{\omega}{c}n_1 \quad \epsilon_i^2 = -1, \quad \epsilon_1\cdot\epsilon_2 = \epsilon_i\cdot n_j = 0 \tag{3.6}$$

Aufgrund der Definition bilden die Vektoren eine Basis in der Lorentz-Raumzeit. Somit können wir die Vierer-Vektor u und a auf folgende Art schreiben:

$$u = \frac{1}{2}(u\cdot n_1)n_2 + \frac{1}{2}(u\cdot n_2)n_1 - (\epsilon_1\cdot u)\epsilon_1 - (\epsilon_2\cdot u)\epsilon_2 \tag{3.7a}$$

$$a = \frac{1}{2}(a\cdot n_2)n_1 - (\epsilon_1\cdot a)\epsilon_1 - (\epsilon_2\cdot a)\epsilon_2 \ , \text{ da } n_1\cdot a = 0 \tag{3.7b}$$

für die Skalarprodukte mit sich selbst ergibt sich somit:

$$1 = u\cdot u = \frac{\Omega}{\omega}n_2\cdot u - (\epsilon_1\cdot u)^2 - (\epsilon_2\cdot u)^2, \ n_1\cdot u = \frac{c}{\omega}k\cdot u = \frac{\Omega}{\omega} \tag{3.8a}$$

$$a\cdot a = -(\epsilon_1\cdot a)^2 - (\epsilon_2\cdot a)^2 \tag{3.8b}$$

Beachtet man nun die Erhaltungsgrösse aus Gleichung 3.5 und die Linearität des Skalarproduktes, so kann man noch drei weitere Skalarprodukte auswerten:

$$\epsilon_i\cdot u = \epsilon_i\cdot u_0 - (a-a_0) = \epsilon_i\cdot u_0 - \Delta a \tag{3.9}$$

$$n_2\cdot u = \frac{\omega}{\Omega}(1 + (\epsilon_1\cdot u)^2 + (\epsilon_2\cdot u)^2) = \frac{\omega}{\Omega}(1 + (\epsilon_1\cdot u_0 - \Delta a)^2 + (\epsilon_2\cdot u_0 - \Delta a)^2) \tag{3.10}$$

Nachdem nun alle Skalarprodukte aus 3.7a bestimmt wurden, ergibt sich u zu:

$$u = \frac{\Omega}{2\omega} n_2 + \frac{\Omega}{2\omega}\left(1 + (\epsilon_1 \cdot u_0 - \Delta a)^2 + (\epsilon_2 \cdot u_0 - \Delta a)^2\right) n - (\epsilon_1 \cdot u_0 - \Delta a)\epsilon_1 - (\epsilon_2 \cdot u_0 - \Delta a)\epsilon_2 \tag{3.11}$$

Zur Anfangszeit ist $u = u_0$ und $\Delta a = 0$, so dass sich u schreiben lässt als:

$$u = u_0 - \frac{\omega}{\Omega}\sum_i ((u_0 \cdot \epsilon_i)(\Delta a \cdot \epsilon_i))n - \frac{\omega}{2\Omega}(\Delta a \cdot \Delta a)n + \sum_i ((\epsilon_i \cdot \Delta a)\epsilon_i) \tag{3.12}$$

Nutzen wir nun noch folgenden Zusammenhang aus,

$$\begin{aligned}\Delta a - \frac{\omega}{\Omega}(\Delta a \cdot u_0)n &= \frac{1}{2}(\Delta a \cdot n_2)n_1 + \sum_i (\epsilon_i \cdot \Delta a)\epsilon_i - \frac{\omega}{\Omega}\sum ((u_0 \cdot \epsilon_i)(\epsilon_i \cdot \Delta a))n - \frac{1}{2}(\Delta a \cdot n_2)n_1 \\ &= \sum_i (\epsilon_i \cdot \Delta a)\epsilon_i - \frac{\omega}{\Omega}\sum ((u_0 \cdot \epsilon_i)(\epsilon_i \cdot \Delta a))n \end{aligned} \tag{3.13}$$

erhalten wir folgendes einfaches Ergebnis für die Vierergeschwindigkeit u in Abhängigkeit der Eigenzeit τ des Testteilchens, wobei wir uns noch den Zusammenhang von der Wellenfrequenz ω und dem Betrag des dreier Wellenzahlvektors $|\vec{k}| = \frac{\omega}{c}$ zu Nutze machten:

$$u(\tau) = u_0 - \left(\Delta a - k\frac{c\Delta a \cdot u_0}{\Omega}\right) - k\frac{c\Delta a \cdot \Delta a}{2\Omega} \tag{3.14}$$

Man erkennt, dass nach dem Abschalten des externen Feldes das Teilchen wieder die gleiche Geschwindigkeit besitzt, die es vor dem Einschalten des Feldes hatte. Wir konnten somit bestätigen, dass es nicht möglich ist mit einer ebenen Welle einem Testteilchen dauerhaft Energie zuzuführen. Dies steht im Einklang mit dem Theorem von Lawson-Woodward und unserer Auffassung der ebenen Welle als Photonenstrom.

Wir wollen uns im folgenden noch die Lösung der Bewegungsgleichung für einen Spezialfall anschauen. Dabei werden wir die Problematik einmal für ein anfangs ruhendes Teilchen und eines, welches im Mittel ruht, betrachten. Wir wählen folgendes Vierer-Potenzial für eine elliptisch polarisierte Welle:

$$a = \alpha_0(0, \delta\sin(\Omega\tau), -\sqrt{1-\delta^2}\cos(\Omega\tau), 0)^t \tag{3.15}$$

Dabei repräsentiert δ den Cosinus eines Winkels im ersten Quadranten und läuft somit von 1 bis 0. Wir erhalten damit folgende Vierergeschwindigkeit, bei der wir eine Frequenzverdopplung in der z-Komponente erkennen.

$$u = \begin{pmatrix} u_0^0 + \omega\alpha_0\frac{u_0^2\sqrt{1-\delta^2}(\cos(\Omega\tau)-1)-\delta u_0^1\sin(\Omega\tau)}{\Omega} + \omega{\alpha_0}^2\frac{(1-\delta^2)(1-2\cos(\Omega\tau))+\frac{1}{2}+(\frac{1}{2}-\delta^2)\cos(2\Omega\tau)}{2\Omega} \\ u_0^1 + \alpha_0\delta\sin(\Omega\tau) \\ u_0^2 + \alpha_0\sqrt{1-\delta^2}\,(1-\cos(\Omega\tau)) \\ u_0^3 + \omega\alpha_0\frac{u_0^2\sqrt{1-\delta^2}(\cos(\Omega\tau)-1)-\delta u_0^1\sin(\Omega\tau)}{\Omega} + \omega{\alpha_0}^2\frac{(1-\delta^2)(1-2\cos(\Omega\tau))+\frac{1}{2}+(\frac{1}{2}-\delta^2)\cos(2\Omega\tau)}{2\Omega} \end{pmatrix} \tag{3.16}$$

Durch elementare Integration nach τ und Multiplikation mit c erhalten wir die Ortskoordinaten in Abhängigkeit der Eigenzeit und somit durch Umstellen auch in Abhägigkeit der Koordinatenzeit t.

Mit der Anfangsbedingung, dass das Teilchen am Anfang ruht und sich im Koordinatenursprung befindet, ergibt sich für eine Polarization in die x-Richtung, d.h. $\delta = 1$ eine Trajektorie wie in Abb. 1. Betrachten wir ein Teilchen, welches sich im Mittel nicht bewegt, erhalten wir hingegen eine Trajektorie wie in Abb. 2. Der Fall einer gedämpften ebenen Welle kann mittels der Matlab gui Grafik.fig betrachtet werden. Diese Welle hat folgendes Vierer potenzial:

$$a = \alpha_0 e^{-\Omega\tau/\xi^2}(0, \delta\sin(\Omega\tau), -\sqrt{1-\delta^2}\cos(\Omega\tau), 0)^t \tag{3.17}$$

Dabei ist ξ ein Dämpfungsfaktor.

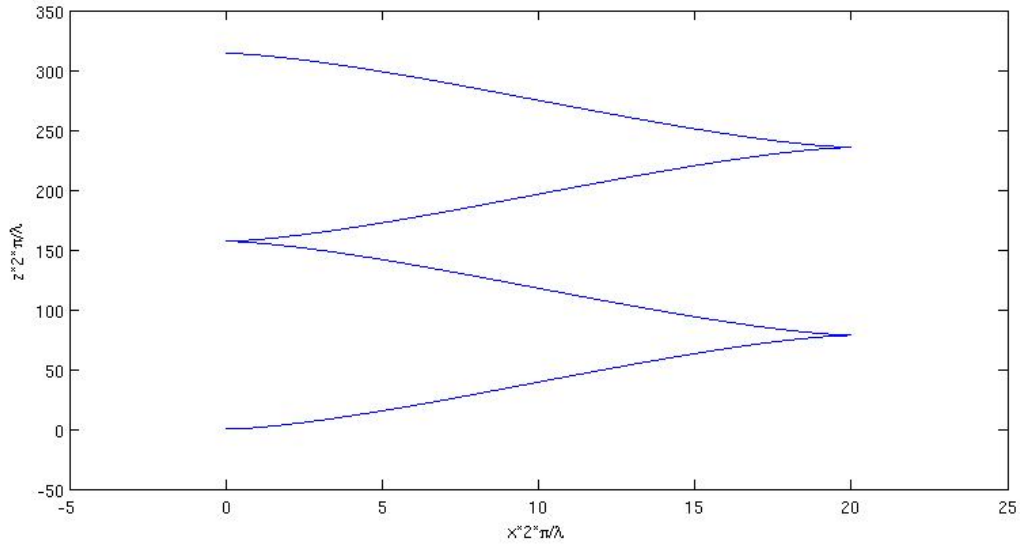

Abbildung 1: Dargestellt ist die Trajektorie in der x-z-Ebene des Laborsystems für zwei Zyklen bei einer linear in x-Richtung polarisierten Welle. Die Parametern sind $\alpha_0 = 10$ $\lambda = 10^{-4} cm$.

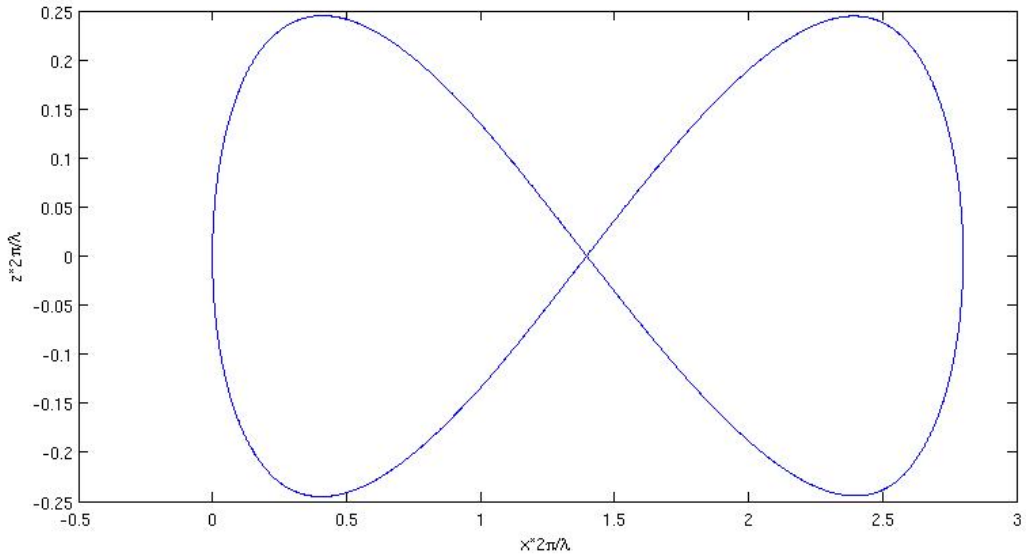

Abbildung 2: Dargestellt ist die Trajektorie in der x-z-Ebene des Laborsystems für zwei Zyklen bei einer linear in x-Richtung polarisierten Welle. Die Parametern sind $\alpha_0 = 10$ $\lambda = 10^{-4} cm$. Die Anfangsgeschwindigkeit ist so gewählt, dass das Teilchen im Mittel ruht.

4 Gaußstrahl als paraxiale Approximation des Laserstrahles

In diesem Abschnitt folgen wir den Ausführungen von Y. Salamin[3] und dem Vortrag von Prof. Wipf[1]. Charakteristische Größen für einen Gaußstrahl sind minimaler Strahlradius w_0 und die Rayleigh-Länge $z_0 = kw_0^2/2$. Damit definieren wir neue dimensionslose Koordinaten:

$$\xi = \frac{x}{w_0} \ , \ \eta = \frac{y}{w_0} \ , \ \zeta = \frac{z}{z_0} \tag{4.1}$$

Im folgenden soll mit einem Lorentz-geeichten Vektorpotenzial gearbeitet werden, so dass gilt:

$$\partial_\mu A^\mu = 0 \tag{4.2}$$

Wir wissen, dass diese Eichung äquivalent dazu ist, dass $\vec{A}$ im Vakuum die homogene Wellengleichung erfüllt. Deshalb verwenden wir den Ansatz der in früheren gaußschen Näherungen für die Lösung der $\mathrm{TEM}_{0,0}$ Mode von $\vec{E}$ gewählt wurde zur Lösung von A. Wir erhalten aus

$$\vec{E} = \frac{E_0}{\sqrt{1+\zeta^2}} \exp\left[(-\imath \arctan\zeta) * \left(\frac{\imath(\xi^2+\eta^2)\zeta}{1+\zeta^2}\right) * (\imath(\omega t - kz))\right]\vec{e}_x \tag{4.3}$$

einen Ansatz, der uns $\vec{A}$ in x Richtung und mit einer harmonischen Ausbreitung in z Richtung wählen lässt.

$$\vec{A} = \alpha_0 \Psi(x,y,z) e^{(\imath(\omega t - kz))}\vec{e}_x \tag{4.4}$$

Daraus folgt die Differentialgleichung für $\vec{A}$:

$$\begin{aligned} \Delta A_i - \frac{1}{c^2}\partial_t^2 A_i = \Delta A_1 + \frac{\omega^2}{c^2} A_1 = 0 \\ \Leftrightarrow \Delta\Psi + 2\imath k \partial_z \Psi = 0 \\ \Leftrightarrow \Delta_\perp \Psi + 4\imath \partial_\zeta \Psi + \epsilon^2 \partial_\zeta^2 \Psi = 0 \end{aligned} \tag{4.5}$$

Wobei wir in der letzten Gleichung schon unsere dimensionslose Grössen verwendeten und $\Delta_\perp = \partial_\xi^2 + \partial_\eta^2$ sowie $\epsilon^2 = \frac{\omega_0}{z_0}$ abgekürzten. Um eine Lösung für Ψ zu finden, wird ein störungstheoretischer Ansatz mit dem Störparameter ϵ^2 gewählt. Dazu wird Ψ in Form einer Reihe aufgeschrieben:

$$\Psi = \sum_0^\infty \epsilon^{2i} \Psi_i \tag{4.6}$$

Die Reihe wird in die Differentialgleichung 4.5 eingesetzt und anschliessend nach Potenzen in ϵ^2 sortiert. Man erhält damit folgendes System von gekoppelten Differentialgleichung:

$$\Delta_\perp \Psi_0 - 4\imath \partial_\zeta \Psi_0 = 0 \tag{4.7}$$

$$\Delta_\perp \Psi_i - 4\imath \partial_\zeta \Psi_i + \epsilon^2 \partial_\zeta^2 \Psi_{i-1} = 0 \ , \ i \in \mathbb{N} \setminus 0 \tag{4.8}$$

Auf der Suche nach einer Grundlösung, erkennen wir Gleichung 4.7 als paraxiale Wellengleichung deren Lösung für $\vec{E}$ 4.3 aus früheren Näherungen bekannt ist und nehmen als Lösung:

$$\Psi_0 = f e^{-f\rho^2} \ , \ f = \frac{\imath}{\zeta + \imath} = \frac{\imath\zeta - 1}{1+\zeta^2} = \frac{1}{\sqrt{1+\zeta^2}} e^{\imath \arctan\zeta} \ , \ \rho^2 = \xi^2 + \eta^2 \tag{4.9}$$

Es existiert analog zur elektrischen Feldstärke $\vec{E}$ auch weitere Lösungen, welche dort den anderen Moden entsprechen. Indem wir uns somit auf diese Lösung konzentrieren, beschränken wir somit unseren Laser auf eine Mode. Warum diese Wahl sinnvoll ist, wird im späteren Verlauf noch einmal aufgegriffen.

Um uns die Lösungen höherer Ordnung zu verschaffen, untersuchen wir die Wirkung der Differentialoperatoren auf ausgewählte Funktionen. Es gelten folgende Regeln:

$$\Delta_\perp \left(\rho^a e^{-f\rho^2}\right) = (a^2 - 4(a+1)\rho^2 f + 4\rho^2 f^2)\rho^{a-2} e^{-f\rho^2} \tag{4.10a}$$

$$f' = \imath f^2 \quad , \quad f'' = -2f^3 \tag{4.10b}$$

$$\partial_\zeta^2 (f^b e^{-f\rho^2}) = -(b(b+1) - 2(b+1)f\rho^2 + f^2\rho^4) f^{b+2} e^{-f\rho^2} \tag{4.10c}$$

Wie man leicht sieht gilt somit folgendes:

$$(\Delta_\perp - 4\imath\partial_\zeta)(\rho^a f^b e^{-f\rho^2}) = (a^2 + 4(b-a-1)\rho^2 f)\rho^{a-2} f^b e^{-f\rho^2} \tag{4.11a}$$

$$\partial_\zeta^2 (\rho^a f^b e^{-f\rho^2}) = -(b(b+1) - 2(b+1)f\rho^2 + f^2\rho^4)\rho^a f^{b+2} e^{-f\rho^2} \tag{4.11b}$$

Aufgrund der Struktur der Ausdrücke in den Klammern wählen wir folgenden Ansatz:

$$\Psi_p = f^{bp+1}\rho^{cp} \sum_{q=0} a_q^{(p)} (\rho^2 f)^q e^{-f\rho^2} \tag{4.12}$$

Dabei sind b und c zwei beliebige Konstanten. Die Struktur innerhalb der Summe entspricht der Struktur unserer obigen Ableitungen und die Faktoren außerhalb der Summe wurden so gewählt, dass für Ψ_0 die richtige Form gegeben ist.
Setzen wir nun diesen Ansatz in die Differentialgleichung ein erhalten wir folgende zwei Teilergebnisse die unsere Rekursion bestimmen:

$$\begin{aligned}(\Delta_\perp - 4\imath\partial_\zeta)\Psi_p &= \sum_{s=0} a_s^{(p)} \left[(2s+cp)^2 + 4(bp - s + cp)\rho^2 f\right] \rho^{2s+cp-2} f^{bp+s+1} e^{-f\rho^2} \\ &= \sum_{s=-1} \left[a_{s+1}^{(p)}(2s+2+cp)^2 + a_s^{(p)} 4(bp-s+cp)\rho^2 f\right] \rho^{2s+cp} f^{bp+s+2} e^{-f\rho^2}\end{aligned} \tag{4.13a}$$

$$\begin{aligned}&\partial_\zeta^2 \Psi_{p-1} = \\ &= \sum_{q=0} a_q^{(p-1)} \left[-((bp+q+1-b)(bp+q+2-b) - 2(bp+q+2-b)f\rho^2 + f^2\rho^4)\right] \rho^{2q+cp-c} f^{bp+q+3-b} e^{-f\rho^2} \\ &= \sum_{q=0} -a_q^{(p-1)}(bp+q+1-b)(bp+q+2-b) + a_{q-1}^{(p-1)} 2(bp+q+1-b) - a_{q-2}^{(p-1)} \rho^{2q+cp-c} f^{bp+q+3-b} e^{-f\rho^2}\end{aligned} \tag{4.13b}$$

Dabei sind alle Koeffizienten $a_q^{(p)}$=0 für q<0. Damit ein Koeffizientenvergleich zu einer nichttrivialen Lösung führt, muss aufgrund der Unabhängigkeit von ρ und f die Potenz von ρ und f für ein Paar (q,s) in den zwei Summen jeweils seperat übereinstimmen. Es ergibt sich somit folgendes einfaches Gleichungssystem, welches den Zusammenhang zwischen b und c beschreibt.

$$\begin{aligned} 2q + cp - c = 2s + cp \qquad & bp + q + 3 - b = bp + s + 2 \\ \Rightarrow \; c = 2(b-1) & \end{aligned} \tag{4.14}$$

Da aber q und s nur aus den natürlichen Zahlen gewählt werden dürfen ergibt sich auch, dass b und c ganz sind. Beachtet man dies, ergibt sich, dass eine Wahl von $b \neq 1$ und $c \neq 0$ nur eine Indexverschiebung zu b=1 und c=0 in der Summe bewirkt. Wir können somit ohne Beschränkung der Allgemeinheit b=1 und c=0 wählen. Aus den obigen Gleichungen ergibt sich somit folgender Koeffizientenvergleich:

$$a_q^{(p-1)}(p+q)(p+q+1) - 2a_{q-1}^{(p-1)}(p+q) + a_{q-2}^{(p-1)} = 4(a_{q+1}^{(p)}(q+1)^2 + a_q^{(p)}(p-q)) \tag{4.15}$$

Was uns zu folgendem Gleichungssystem führt:

$$M^{(p)}a^{(p)} = N^{(p)}a^{(p-1)} \tag{4.16}$$

Mit:

$$a^{(p)} = \begin{pmatrix} a_0^{(p)} \\ a_1^{(p)} \\ a_2^{(p)} \\ a_3^{(p)} \\ \vdots \end{pmatrix} \quad , \quad M^{(p)} = 4 \cdot \begin{pmatrix} p & 1 & 0 & 0 & \dots \\ 0 & p-1 & 2^2 & 0 & \dots \\ 0 & 0 & p-2 & 3^2 & \dots \\ 0 & 0 & 0 & p-3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$N^{(p)} = \begin{pmatrix} p(p+1) & 0 & 0 & 0 & \dots \\ -2(p+1) & (p+1)(p+2) & 0 & 0 & \dots \\ 1 & -2(p+2) & (p+2)(p+3) & 0 & \dots \\ 0 & 1 & -2(p+3) & (p+3)(p+4) & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Es ist aufgrund der Ober- bzw Unterdiagonalform ohne großen Aufwand möglich die Determinanten der Matrizen zu berechnen. Deshalb ist auch ersichtlich, dass das Kernel von $M^{(p)}$ eindimensional ist, da immer genau der (p+1). Diagonaleintrag Null ist. Daraus folgend ist die Matirx M für alle Störungsordnungen p nicht invertierbar. Die Matrix N hingegen ist für alle p>0, d.h. für alle relevanten p, invertierbar. Da $a^{(0)} = (1, 0, \dots)^t$ bekannt ist, wird bei der Auflösung des Gleichungssystems nach $a^{(p)}$ ein Parameter frei wählbar. Für eine Näherung interessieren uns die ersten beiden Korrekturen, d.h. wir suchen $a^{(1)}$ und $a^{(2)}$. Für diese Spezialfälle ist das Gleichungssystem leicht aufzulösen und wir erhalten folgende Ergebnisse:

$$a^{(0)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \quad , \quad a^{(1)} = \frac{1}{4}\begin{pmatrix} 4d_1 \\ 2-4d_1 \\ -1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \quad , \quad a^{(1)} = \frac{1}{32}\begin{pmatrix} 4d_2 \\ 48d_1 - 64d_2 \\ 2 - 28d_1 + 16d_2 \\ 8d_1 - 8 \\ 1 \\ 0 \\ \vdots \end{pmatrix} \tag{4.17}$$

Dabei ist der Vektor $d = (d_1, d_2, \dots)^t$ frei wählbar. Daraus gewinnen wir eine Darstellung für Ψ, wobei im folgenden alle Gleichungen nur bis zur Ordnung ϵ^5 notiert werden. Das heißt die vollständige Gleichung ist die angegebene plus $O(\epsilon^6)$.

$$\begin{aligned} \Psi =& f e^{-f\rho^2} \Bigg(1 + \epsilon^2 \left[4d_1 + (2-4d_1)f\rho^2 - f^2\rho^4\right]\frac{f}{4} + \\ &+ \epsilon^4 \left[32d_2 + (48d_1 - 64d_2)f\rho^2 + (12 - 48d_1 + 16d_2)f^2\rho^4 + (8d_1 - 8)f^3\rho^6 + f^4\rho^8\right]\frac{f^2}{32}\Bigg) \end{aligned} \tag{4.18}$$

Und somit ergibt sich $\vec{A}$ zu:

$$\begin{aligned} \vec{A} =& \vec{e}_1 \alpha_0 f e^{-f\rho^2 + \imath(\omega t - kz)} \Bigg(1 + \epsilon^2 \left[4d_1 + (2-4d_1)f\rho^2 - f^2\rho^4\right]\frac{f}{4} + \\ &+ \epsilon^4 \left[32d_2 + (48d_1 - 64d_2)f\rho^2 + (12 - 48d_1 + 16d_2)f^2\rho^4 + (8d_1 - 8)f^3\rho^6 + f^4\rho^8\right]\frac{f^2}{32}\Bigg) \end{aligned} \tag{4.19}$$

Daraus bestimmen wir die Divergenz unseres Vektorpotenzials sortiert nach Potenzen von ϵ und f:

$$\begin{aligned}\nabla\vec{A} = \alpha_0 f e^{-f\rho^2+\imath(\omega t-kz)}\frac{\xi}{\omega_0}\cdot \\ \left\{1+\epsilon^2\left[(1-4d_1)f^2+2(-1+d_1)\rho^2 f^3+\frac{1}{2}f^4\rho^4\right]\right. \\ \left.+\epsilon^4\left[(3d_1-6d_2)f^3+\left(\frac{3}{2}-9d_1+6d_2\right)f^4\rho^2+\left(-\frac{9}{4}+\frac{9}{2}d_1-d_2\right)f^5\rho^4+\left(\frac{3}{4}-\frac{1}{2}d_1\right)f^6\rho^6-\frac{1}{16}f^7\rho^8\right]\right\}\end{aligned} \tag{4.20}$$

Mithilfe der Divergenz und den uns bekannten Formel zur Berechnung der komplexen elektrischen und magnetischen Feldstärke

$$\vec{E} = -\partial_t(A) - \frac{i}{k}\nabla(\nabla\cdot\vec{A}) \tag{4.21a}$$

$$\vec{B} = \nabla\times\vec{A} \tag{4.21b}$$

folgt für die reellen Felder folgender Ausdruck, sortiert in Potenzen von ϵ, wobei hier wie von Salamin et al. $d_1 = 1/2$ und d_2=3/8 gewählt wurde:

$$E_x = E\left[S_0+\epsilon^2\left[S_2\xi^2-\frac{S_3}{4}\rho^4\right]+\epsilon^4\left[\frac{S_2}{8}-\frac{S_3}{4}\rho^2-\frac{S_4}{16}(\rho^4-16\xi^2\rho^2)-\frac{S_5}{8}(\rho^6+2\xi^2\rho^4)+\frac{S_6}{32}\rho^8\right]\right] \tag{4.22a}$$

$$E_y = E\xi\eta\left[\epsilon^2 S_2+\epsilon^4\left[S_4\rho^2-\frac{S_5}{4}\rho^4\right]\right] \tag{4.22b}$$

$$E_z = E\xi\left[\epsilon C_1+\epsilon^3\left[-\frac{C_2}{2}+C_3\rho^2-\frac{C_4}{4}\rho^4\right]+\epsilon^5\left[-\frac{3C_3}{8}-\frac{3C_4}{8}\rho^2+\frac{17C_5}{16}\rho^4-\frac{3C_6}{8}\rho^6+\frac{C_7}{32}\rho^8\right]\right] \tag{4.22c}$$

$$B_x = 0 \tag{4.22d}$$

$$B_y = E\left[S_0+\epsilon^2\left[\frac{S_2}{2}\rho^2-\frac{S_3}{4}\rho^4\right]+\epsilon^4\left[-\frac{S_2}{8}+\frac{S_3}{4}\rho^2+\frac{5S4}{\rho^4}-\frac{S_5}{4}\rho^6+\frac{S_6}{32}\rho^8\right]\right] \tag{4.22e}$$

$$B_z = E\eta\left[\epsilon C_1+\epsilon^3\left[\frac{C_2}{2}+\frac{C_3}{2}\rho^2-\frac{C_4}{4}\rho^4\right]+\epsilon^5\left[\frac{3C_3}{8}+\frac{3C_4}{8}\rho^2+\frac{3C_5}{16}\rho^4-\frac{C_6}{4}\rho^6+\frac{C_7}{32}\rho^8\right]\right] \tag{4.22f}$$

Dabei wurde schon genutzt, dass $\epsilon = 2/(kw_0)$ gilt, wodurch die Ableitung nach x bzw. y zu einer Vergrösserung der Potenz von ϵ um 1 führt, sowie die Dartstellung von $f = \sqrt{1+\zeta^2}^{-1}e^{\imath\arctan(\zeta)}$ und abkürzend

$$E = k|\alpha_0|\frac{1}{\sqrt{1+\zeta^2}}e^{\frac{-\rho^2}{1+\zeta^2}} \tag{4.23}$$

$$S_n = \left(\frac{1}{\sqrt{1+\zeta^2}}\right)^n \sin\left(n\arctan(\zeta)+\omega t-kz-\frac{\zeta}{1+\zeta^2}\rho+\phi\right) \tag{4.24}$$

$$C_n = \left(\frac{1}{\sqrt{1+\zeta^2}}\right)^n \cos\left(n\arctan(\zeta)+\omega t-kz-\frac{\zeta}{1+\zeta^2}\rho+\phi\right) \tag{4.25}$$

eingeführt. Dabei sei an die Definition von $\zeta = \frac{z}{z_0}$ erinnert. Der konstante Phasenfaktor ϕ ist die Phase der komplex wählbaren Amplitude α_0. Hier erkennt man warum für Ψ_0 die Lösung 4.9 gewahlt wurde. Durch diese Wahl haben wir erreicht, dass die Lösung in nullter Ordnung von ϵ mit der früheren Lösung für eine Gaußstrahlapproximation 4.3 übereinstimmt. Wir haben bei

unserem Laserstrahl von einer Strahltaille w_0 gesprochen. Am schnellsten fällt die Amplitude mit der Entfernung von der z-Achse in der Ebene $z = 0$ und wir wollen somit davon sprechen, dass dort die Strahltaille liegt. Dies nehmen wir zum Anlass eine Funktion w als Strahlweite zu definieren. Der Term der einen Einfluss auf die Amplitude der Felder hat und nur vom Ort in z-Richtung abhängt, ist der Wurzelterm in E. Wir definieren uns deshalb w zu:

$$w(\zeta) = \frac{w_0}{\sqrt{1+\zeta^2}} \tag{4.26}$$

Ist bei fester z-Koordinate $\rho = w$, so ist die Amplitude im Vergleich zur Stelle $\rho = 0$ um den Faktor 1/e abgefallen. Wir wollen im folgenden die Feldstärke genau wie bei ebenen Wellen in Abhängigkeit der dimensionslosen Größe α betrachten. Zuerst betrachten wir den Fall, dass wir aus großer Entfernung auf den Strahlfokus zielen und beobachten den Energiegewinn beziehungsweise Verlust in Abhängigkeit von der Energie, mit der das Teilchen startet. Aus Gründen der Einfachheit wollen wir nur Bewegungen in der x-z-Ebene betrachten, in der das Teilchen auch verbleibt, da sowohl die y-Komponente des elektrischen Feldes wie auch die z-Komponente des magnetischen Feldes in diesem Fall Null wird. Wir erkennen in Abbildung 3, dass kein wesentlicher Energiegewinn auf diese Art möglich ist. Dies ist darin begründet, dass das Teilchen sehr schnell den Strahl wieder verlässt und somit der Effekt der Energiegewinnung aus der ponderomotorischen Kraft nur sehr gering ist.
Wir folgen somit der Idee von Salamin et al.[3] und richten den Geschwindigkeitsvektor das Teilchens nicht auf den Strahlfokus, bei uns der Koordinatenursprung, sondern um eine gewisse Länge s, gemessen in Einheiten von z_0, vom Fokus entfernt auf die z-Achse. Dadurch können wir den Fall realisieren, dass das Teilchen eingefangen wird. In Abbildung 4 sehen wir prinzipiell möglichen Fälle. Man beachte, dass ein beträchtlicher Energiegewinn sowohl in Reflexion, Transmission als auch Einfang des Elektrons beobachtet werden kann. Man beachte dabei, dass die Teilchen die nur knapp reflektiert/transmmittiert werden, sehr schnell einen hohen Energiegewinn erfahren und dann den Strahl verlassen. Es ist somit am günstigsten, eine Trajektorie für die Beschleunigung zu wählen wie in Abb. 5. Dagegen ist der Energiegewinn von Teilchen, die im Strahl verbleiben anfangs langsamer. Für genauere Analysen sei auf die Arbeit von Salamin et al. verwiesen [3]. Man beachte, dass die Einstellung von s sehr fein erfolgen muss, sofern man Teilchen mit fest definierter Anfangsenergie hat. Ansonsten kann man durch Variation beider Parameter eine Beschleunigung erreichen.
Auch die gewählte Phase hat einen wesentlichen Einfluss auf die erreichbare Energie. Die Phase kann durch eine entsprechende Wahl des Abstandes zum Auftreffpunkt reguliert werden. Betrachten wir für den Fall aus Abb. 5 zu 4 verschiedene Phasenwerte die gewonnen Energien erhalten wir Abb. 6. Wir erhalten durch die Variation der Phase einen Unterschied in der Energie, von rund $15,5\% = (E_{end}(\psi_0 = 0.3) - E_{end}(\psi_0 = 0))/E_{end}(\psi_0 = 0.3)$. Man beachte auch, dass der Energiezuwachs von dem Fall mit $\psi_0 = 0.3$ in den drei letzten berechneten Zyklen rund $186mc^2$ beträgt und somit bedeutet weniger, als der Energieunterschied zum Fall mit $\psi_0 = 0$ von $1,8 \cdot 10^5 mc^2$.

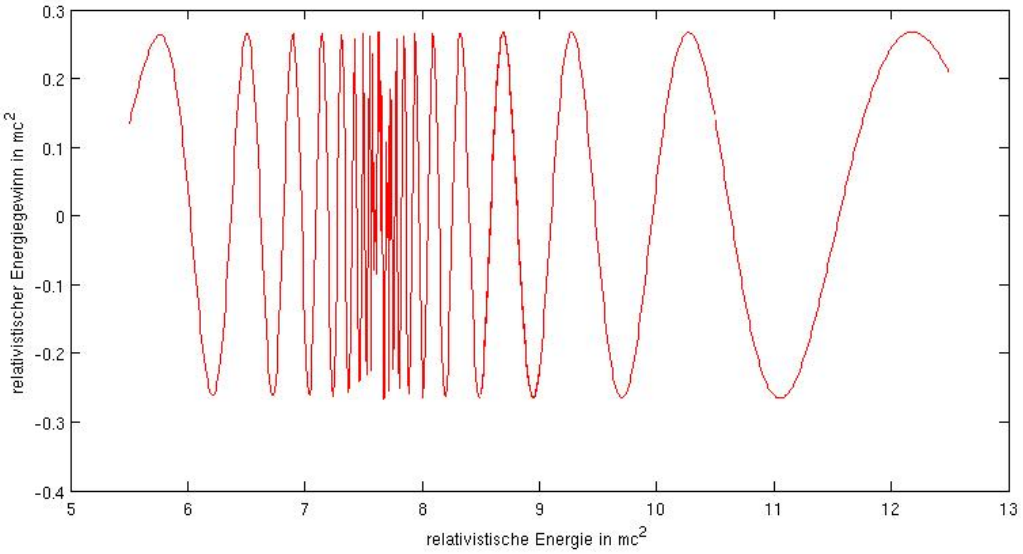

Abbildung 3: Wir sehen die Energieänderung aufgetragen über ein Startenergieintervall berechnet für ein Elektron als Testteilchen. Die gewählten Parameter sind: $\xi = 40$, $\zeta = -1$, $\lambda = 10^{-4}$, $w_0 = 5 \cdot 10^{-4}$, $\alpha = 10$, $\psi_0 = 0$ $s = 0$. Der Bereich in der Energiegewinn osziliert, ist der Energiebereich, in dem die Trajektorie des Teilchens von reflektions- zu einer Transmissionsbahn übergeht. Dies geschieht um einen Energiewert von $7,65mc^2$, wobei die relativistische Energie sich zu $E = \gamma mc^2$ berechnet.

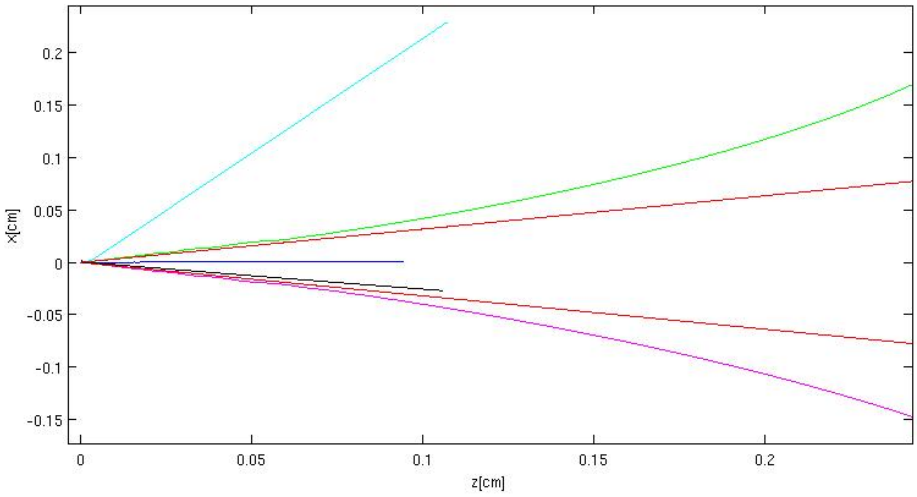

Abbildung 4: Aufgetragen sind 5 mögliche Fälle. Die Parameter wurden wie folgt gewählt: $\xi = 30$ $\zeta = -0,5$ $s = 2$ $\lambda = 10^{-4}cm$ $w_0 = 10^{-4}cm$ $\alpha = 10$ $\psi_0 = 0$. Die rote Linie stellt die Strahlweite dar. Im folgenden sind noch die Startenergie und die Endenergie in mc^2 nach einem Durchlauf von 300 Zyklen angegeben (gemessen in der Eigenzeit des Teilchens bezogen auf ω): Cyan:$E_0 = 3,3$ $E_{end} = 9,0388$, Grün:$E_0 = 9,0388$ $E_{end} = 5397,8769$, Blau:$E_0 = 3,40$ $E_{end} = 3,3989$, Schwarz:$E_0 = 3,41$ $E_{end} = 5,8585$ Magenta:$E_0 = 3,42$ $E_{end} = 6824.896$. Dabei tritt bei allen Trajektorien das Teilchen in den Strahl ein. Die dargestellten Kanten sind auf das verwendete Bildformat zurückzuführen und haben keine physikalische Bewandtnis.

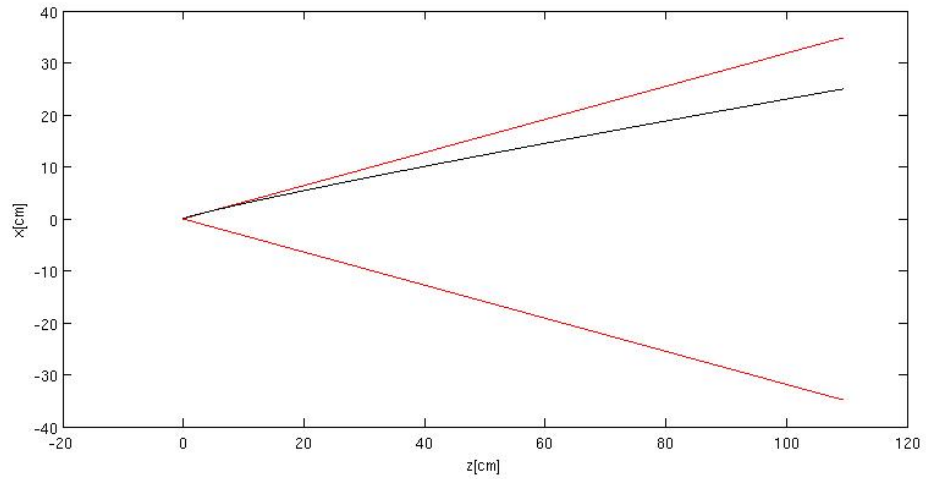

Abbildung 5: Aufgetragen ist die Trajektorie eines beschleunigten Teilchens. Die Parameter wurden wie folgt gewählt: $\xi = 30$ $\zeta = -0,5$ $s = 1,8$ $\lambda = 10^{-4}cm$ $w_0 = 10^{-4}cm$ $\alpha = 10$ $\psi_0 = 0$. Die rote Linie stellt die Strahlweite dar. Die Startenergie des Teilchens betrug 3,41mc^2 während schon nach 300 Zyklen eine Energie von 11749,4mc^2 erreicht wurde. Hier sieht man auch, dass ein Teilchen, welches anfänglich den Strahl verlässt auch wieder eingefangen werden kann, sofern man das Verlassen der Strahlweite als Verlassen des Strahls, wie hier von mir getan, definiert. Die dargestellten Kanten sind auf das verwendete Bildformat zurückzuführen und haben keine physikalische Bewandtnis.

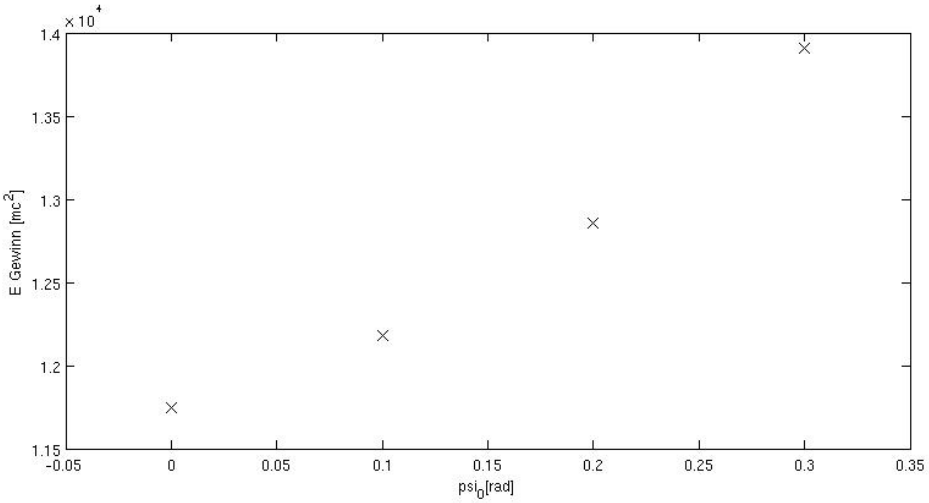

Abbildung 6: Gleiche Parameter wie in Abb. 5 bis auf eine Variation der Phase. Es ist gut zu erkennen, wie die Phasen einen nicht vernachlässigbaren Einfluß auf die Energie hat die das Teilchen nach 300 Zyklen erreicht hat.

5 Bessel Wellen

Bei Bessel Wellen handelt es sich um spezielle Lösungen der Maxwellgleichungen. Um die Lösung zu erhalten wie sie von I. Bialynicki-Birula [4] vorgeschlagen wurde, werden wir noch ein weiteres Konzept der Darstellung der Feldstärkevektoren behandeln. Anstatt auf ein Vektorpotenzial zurückzugreifen, arbeiten wir nur mit den realen physikalischen Grössen der elektrischen und magnetischen Feldstärke. Bisher hatten wir die Vektoren $\vec{E}$ und $\vec{B}$ als Realteil von komplexen Funktionen aufgefasst. Hier aber führen wir den sogenannten Riemann-Silberstein Vektor $\vec{R}$ ein, so dass sich $\vec{E}$ als Realteil und $\vec{B}$ als Imaginärteil des Vektors ergeben:

$$\vec{R} = \vec{E} + \imath\vec{B} \tag{5.1}$$

Man beachte, dass in unseren Einheitensystem E und B die gleiche Einheit haben und somit der Faktor c bei I. Bialynicki-Birula entfällt. Wir schreiben unsere Vakkuummaxwellgleichungen in solche für $\vec{R}$ um, wobei durch Seperation nach Imaginär- und Realteil die ursprünglichen Gleichungen zurückgewonnen werden könnnen. Es ergibt sich somit:

$$c\nabla \times \vec{R} = \imath\partial_t\vec{R} \tag{5.2}$$

$$\nabla \cdot \vec{R} = 0 \tag{5.3}$$

Da wir wieder eine Lösung für einen Laserstrahl suchen, machen wir einen ähnlichen Ansatz wie für das Vektorpotenzial des Gaußstrahls, sind aber noch restriktiver und verlangen, dass die Koordinate z der Ausbreitungsrichtung nur einen Einfluss auf die Phase aber nicht auf die Amplitude von des Riemann-Silberstein Vektors haben soll. Das heißt, dass die Vektoren $\vec{R}(x, y, z_1)$ und $\vec{R}(x, y, z_2)$ für beliebige feste x, y nur um einen Phasenfaktor gegeneinander variieren. Wir machen somit folgenden Ansatz:

$$\vec{R}(x, y, z) = \vec{S}(x, y)e^{\epsilon\imath(\omega t - k_z z)} \tag{5.4}$$

Dabei gibt $\epsilon \in \{-1, 1\}$ den Umlaufsinn der Welle vor $\vec{k} = (k_x + k_y + k_z)^t$ ist der Wellenzahlvektor und $\omega = c|\vec{k}|$ die Kreisfrequenz der Welle. Setzt man diesen Ansatz in die beiden Maxwellgleichungen ein ergibt sich folgendes Gleichungssystem für die Komponenten von $\vec{S}$

$$\omega\epsilon S_x = -c(\partial_y S_z + \epsilon\imath k_z S_y) \tag{5.5a}$$

$$\omega\epsilon S_y = c(\partial_x S_z + \epsilon\imath k_z S_x) \tag{5.5b}$$

$$\omega\epsilon S_z = -c(\partial_x S_y - \partial_y S_x) \tag{5.5c}$$

Wie man leicht mittels der Vertauschbarkeiten der partiellen Ableitungen zeigen kann, erfüllt $\vec{R}$ die Forderung nach Quellfreiheit: $\nabla \cdot \vec{F} = 0$. Wir stellen die obigen drei Gleichungen um, so dass wir $\vec{S}$ in Abhängigkeit von $S_z = \Psi$ darstellen können. Es ergibt sich:

$$S_x = -\frac{c}{\epsilon(w^2 - c^2k_z^2)}(\omega\partial_y\Psi + \imath k_z c\partial_x\Psi) \tag{5.6a}$$

$$S_y = \frac{c}{\epsilon(w^2 - c^2k_z^2)}(\omega\partial_x\Psi - \imath k_z c\partial_y\Psi) \tag{5.6b}$$

$$\Rightarrow \quad 0 = (\frac{\omega^2}{c^2} - k_z^2 + \partial_x^2 + \partial_y^2)\Psi = (k_\perp^2 + \Delta_\perp)\Psi \tag{5.6c}$$

$$\tag{5.6d}$$

Wir haben das Problem auf eine zweidimensionale Helmholtzgleichung reduziert, und können uns eine Lösung in geeigneten Koordinaten konstruieren. Um Besselwellen zu erhalten gehen wir zu Zylinderkoordinaten über und verwenden in der Differentialgleichung folgenden Lösungsansatz:

$$\Psi(\rho, \phi) = \Psi_\rho(\rho) * \Psi_\phi(\phi) \tag{5.7}$$

Dadurch entkoppelt die Differentailgleichung in folgende Gestalt mit der frei wählbaren Konstanten M^2.

$$\Psi_\rho'' + \frac{1}{\rho}\Psi_\rho' \left(k_\perp^2 - \frac{M^2}{\rho^2}\right) \Psi_\rho = 0 \tag{5.8a}$$

$$\Psi_\phi'' + M^2 \Psi_\phi = 0 \tag{5.8b}$$

Bedenkt man die 2π Periodizität von ϕ so muss $M^2 \geq 0$ gelten und unsere beliebige Konstante lässt sich so schreiben. Wir sehen die Lösung des Winkelanteils zu:

$$\Psi_\phi = A\sin(M\phi) + B\cos(M\phi) \tag{5.9}$$

Die Lösung des Radialanteils gewinnen wir aus einschlägiger Literatur, z.B. dem Handbuch der Feld Theorie [6] zu:

$$\Psi_\rho = CJ_M(k_\perp\rho) + DJ_{-M}(k_\perp\rho) \tag{5.10}$$

Dabei ist J_M eine Besselfunktion erster Art, und wir betrachten auch M im folgenden als nicht negativ. Im Falle $M \in \mathbb{N}$ wird nicht die volle Allgemeinheit der Lösung wiedergegeben, aber wir wollen fordern, dass die Funktion an der Stelle $\rho = 0$ regulär ist. Dadurch entfällt die Möglichkeit die Lösung in der Form

$$\Psi_\rho = \tilde{C}J_M(k_\perp\rho) + \tilde{D}Y_{-M}(k_\perp\rho) \tag{5.11}$$

aufzuschreiben, wobei Y_M eine Besselfunktion zweiter Art ist, ohne die Allgemeinheit zu beschränken, da sofort $\tilde{D} = 0$ folgen würde. Damit haben wir folgende allgemeine Lösung gefunden:

$$\Psi = \int (A_M\sin(M\phi) + B_M\cos(M\phi))(C_MJ_M(k_\perp\rho) + D_MJ_{-M}(k_\perp\rho))dM \tag{5.12}$$

Dabei ergeben sich für jede Nichttriviale Lösung zu einem gegebenem M nur drei unabhängige Variablen. Dies ist am einfachsten zu sehen, indem jeweils ein nichtverschwindender Koeffizient in den Summen ausgeklammert wird und die ausgeklammerten Koeffizienten zu einem neuen zusammengefasst werden. Wir haben somit für jede Teillösung vier unabhänige Variablen wie es uns die Lösungstheorie sagt. Diese sind M und die Koeffizienten A, B, C und D.

Wir wollen obigen Ausdruck für Ψ jeweils nur für ausgewählte M untersuchen und dabei auch für numerische Berechnungen die Koeffizienten A, B, C und D auf eine spezielle Art wählen. In der Auswahl der Koeffizienten folgen wir I. Bialynicki-Birula [4] und verwenden folgenden Ausdruck für Ψ:

$$\Psi = A(\cos(\phi) + \imath\sin(\phi))^M J_M(k_\perp\rho) \tag{5.13}$$

Dabei wurde $A = A$, $B = \imath A$, $C = 1$ und $D = 0$ gewählt. Wir erhalten für die Komponenten des Riemann-Silberstein Vektors $\vec{R}$ folgende Ausdrücke:

$$R_x = E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(\left(\frac{\omega}{c} + k_z\right) e^{\imath(M-1)\phi} J_{M-1}(k_\perp\rho) + \left(\frac{\omega}{c} - k_z\right) e^{\imath(M+1)\phi} J_{M+1}(k_\perp\rho)\right)/2k_\perp \tag{5.14a}$$

$$R_y = \imath E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(\left(\frac{\omega}{c} + k_z\right) e^{\imath(M-1)\phi} J_{M-1}(k_\perp\rho) - \left(\frac{\omega}{c} - k_z\right) e^{\imath(M+1)\phi} J_{M+1}(k_\perp\rho)\right)/2k_\perp \tag{5.14b}$$

$$R_z = E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(\imath\epsilon e^{iM\phi} J_M(k_\perp\rho)\right) \tag{5.14c}$$

Dabei gilt $E_0 = -\imath A\epsilon$ eine Konstante die nicht von den vier Parametern $\epsilon,\ M,\ , k_\perp, k_z$ unseres Vektors abhänigig sein soll. Wir erkennen, dass für nichtdivergente Lösungen von $\vec{R}$

$M \in \{0\} \cup [1, \infty)$ gelten muss. Interessant ist es den Spezialfall $k_z \to |\vec{k}| = k$ zu betrachten. Damit folgt $k_\perp \to 0$ und $ck_z \to \omega$. Dadurch entfällt in den Klammern jeweils der zweite Summand, wie uns folgende Umformung zeigt:

$$\frac{\omega/c - k_z}{k_\perp} = \frac{k - k_z}{k_\perp} = \frac{k - k_z}{\sqrt{k^2 - k_z^2}} = \sqrt{\frac{k - k_z}{k + k_z}} \tag{5.15}$$

Aus dem asymptotischen Verlauf für kleine Argumente und semipositivem Index sowie dem Zusammenhang zwischen Besselfunktionen mit positvem und negativem ganzen Index

$$J_\alpha(x) \to \frac{1}{\Gamma(\alpha+1)} \left(\frac{x}{2}\right)^\alpha \ , \ \alpha \geq 0 \ , \ J_{-n}(x) = (-1)^n J_n(x) \ , \ n \in \mathbb{N} \tag{5.16}$$

folgt, dass nur die Werte M=0, 2 existieren, so dass der Ausdruck für $\vec{R}$ weder trivial wird noch im Ursprung bzw. auf der ganzen z Achse divergiert. Im Falle $k_z \to -|\vec{k}|$ erhalten wir unter der gleichen Forderung nur die Lösung M=0. Diese beiden Fälle wollen wir im folgenden weiter untersuchen.

5.1 Der Fall M=2

Wir betrachten zunächst den Fall M=2, da dieser in dem von uns betrachteten Grenzwert auf ein einfacheres Problem als M=0 führt. Der Riemann-Silberstein Vektor nimmt in diesem Fall folgende Form an:

$$R_x = E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(\left(\frac{\omega}{c} + k_z\right) e^{\imath\phi} J_1(k_\perp \rho) + \left(\frac{\omega}{c} - k_z\right) e^{3\imath\phi} J_3(k_\perp \rho) \right) / 2k_\perp \tag{5.17a}$$

$$R_y = \imath E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(\left(\frac{\omega}{c} + k_z\right) e^{\imath\phi} J_1(k_\perp \rho) - \left(\frac{\omega}{c} - k_z\right) e^{3\imath\phi} J_3(k_\perp \rho) \right) / 2k_\perp \tag{5.17b}$$

$$R_z = E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(\imath\epsilon e^{2i\phi} J_2(k_\perp \rho) \right) \tag{5.17c}$$

Damit gilt im Grenzfall $k_z \to |\vec{k}| = k$ und somit auch in guter Näherung, das heißt bis zur linearen Ordnung in ρ, Allgemein nahe der z-Achse, für eine sich in positive z-Richtung bewegende Welle, folgende Beziehung:

$$R_x \to \frac{E_0 k}{2} e^{\imath\epsilon(\omega t - kz)} e^{\imath\phi} \rho = S_x \tag{5.18a}$$

$$R_y \to \frac{E_0 k}{2} e^{\imath\epsilon(\omega t - kz)} \imath e^{\imath\phi} \rho = S_y \tag{5.18b}$$

$$R_z \to 0 = S_z \tag{5.18c}$$

Wir haben auch hier weiter $k_z = k$ gewählt und wollen die Leistung betrachten, die ein Laser haben muss, der ein solches Feld bis zu einem Abstand ρ_0 von der z-Achse erzeugt.

$$P = \int\limits_F \vec{S} d\vec{o} = \int_0^{R_0} \int\limits_0^{2\pi} \left(E_x^2 + E_y^2\right) d\phi \rho d\rho \propto r h o_0^4 \tag{5.19}$$

Wir sehen, dass eine solche Welle nicht im ganzen Raum realisiert werden kann und aufgrund des schnellen Wachstums mit dem Abstand zur z-Achse auch nur paraxial eine gute Näherung sein kann. Deshal beschränken wir unsere Ausführungen auf paraxiale Betrachtungen und wählen unsere spezielle intuitive Form des Wellenzahlvektors $\underline{k} = \frac{\omega}{c}(1, 0, 0, 1)$. Wir können folgendes Vierer-Potenzial A wählen:

$$A = \frac{E_0}{2} \rho \begin{pmatrix} 0 \\ \sin(\omega t - kz)\cos(\phi) + \epsilon \cos(\omega t - kz)\sin(\phi) \\ -\sin(\omega t - kz)\sin(\phi) + \epsilon \cos(\omega t - kz)\cos(\phi) \\ 0 \end{pmatrix} \tag{5.20}$$

Dieses A erfüllt die Eichbedingung $\underline{k} \cdot A = 0$ aufgrund der Diagonalgestalt der Metrik, wobei $\underline{k}$ der Vierer-Wellenzahlvektor ist. Man kann sich auch leicht überzeugen, dass $(k^\mu \partial_\mu)A = 0$ erfüllt ist. Somit sind alle Vorraussetzungen der Herleitung für Gleichung 3.4 erfüllt und wir schliessen erneut $\underline{k} \cdot \underline{u} = \frac{\Omega}{c} = const.$ Damit hat der Vierer-Vektor u nur noch die zwei unabhängigen Einträge u^1 und u^2, wobei sich die anderen beiden Einträge wie folgt ergeben:

$$u^0 = \frac{\Omega}{\omega} + u^3 = \left(1 + \left(\frac{\Omega}{\omega}\right)^2 + (u^1)^2 + (u^2)^2\right)\frac{\omega}{2\Omega} \tag{5.21a}$$

$$u^3 = \left(1 - \left(\frac{\Omega}{\omega}\right)^2 + (u^1)^2 + (u^2)^2\right)\frac{\omega}{2\Omega} \tag{5.21b}$$

Wir stellen unsere Bewegungsgleichungen auf und nutzen die Tatsache, dass $\imath H_x = H_y$ und somit $E_x = B_y$ und $E_y = -B_x$ ist, sowie dass die z-Komponente beider Felder Null ist.

$$\frac{d^2(x^0)}{d\tau^2} = \ddot{x^0} = \frac{e}{m}(u^1 E_x + u^2 E_y) \tag{5.22a}$$

$$\frac{d^2(x^1)}{d\tau^2} = \ddot{x^1} = \frac{e}{m}(u^0 - u^3) E_x \tag{5.22b}$$

$$\frac{d^2(x^2)}{d\tau^2} = \ddot{x^2} = \frac{e}{m}(u^0 - u^3) E_y \tag{5.22c}$$

$$\frac{d^2(x^3)}{d\tau^2} = \ddot{x^3} = \frac{e}{m}(u^1 E_x + u^2 E_y) \tag{5.22d}$$

Auch hier sehen wir die Erhaltungsgrösse Ω und somit vereinfacht sich unsere Bewegungsgleichung zu folgender Form, wobei wir $ct - z|_{\tau=0} = const = 0$ setzen, da diese Konstante nur eine unwichtige Phase in den Gleichungen erzeugt. Es ergibt sich somit für die x^1 und x^2 folgendes Differentialgleichungssystem:

$$\ddot{x^1} = C\left(\cos(\Omega\tau)x^1 - \sin(\Omega\tau)x^2\right) \tag{5.23a}$$

$$\ddot{x^2} = -C\left(\sin(\Omega\tau)x^1 + \cos(\Omega\tau)x^2\right) \tag{5.23b}$$

Dabei haben wir $C = \frac{E_0 e\Omega}{2mc}$ eingeführt. Im nächsten Schritt übernehmen wir die Idee von I. Bialyncki-Birula [5] für die Lösung des Systems. Wir definieren neue Koordinaten, die mit der Hälfte der Frequenz Ω mitrotieren:

$$r = x^1 \cos(\Omega\tau/2) - x^2 \sin(\Omega\tau/2) \tag{5.24a}$$

$$s = x^1 \sin(\Omega\tau/2) + x^2 \cos(\Omega\tau/2) \tag{5.24b}$$

Wir stellen obige Gleichungen nach x^1 und x^2 um, leiten anschließend ab und vergleichen mit unserer Bewegungsgleichung, die wir mit Hilfe der Additionstheoreme auf eine einfache Form in r und s schreiben:

$$C(r\cos(\Omega\tau/2) - s(\sin\Omega\tau/2)) = \ddot{x^1} = \cos(\Omega\tau/2)\left(\ddot{r} + \dot{s}\Omega - r\frac{\Omega^2}{4}\right) - \sin(\Omega\tau/2)\left(-\ddot{s} + \dot{r}\Omega + s\frac{\Omega^2}{4}\right) \tag{5.25a}$$

$$C(r\sin(\Omega\tau/2) + s\cos(\Omega\tau/2)) = -\ddot{x^2} = \sin(\Omega\tau/2)\left(\ddot{r} + \dot{s}\Omega - r\frac{\Omega^2}{4}\right) + \cos(\Omega\tau/2)\left(-\ddot{s} + \dot{r}\Omega + s\frac{\Omega^2}{4}\right) \tag{5.25b}$$

Kobinieren wir beide Gleichungen erhalten wir folgende einfache Ausdrücke für die Bewegungsgleichung in r und s:

$$\ddot{r} = -\dot{s}\Omega + r\left(\frac{\Omega^2}{4} + C\right) \tag{5.26a}$$

$$\ddot{s} = \dot{r}\Omega + s\left(\frac{\Omega^2}{4} - C\right) \tag{5.26b}$$

Wir haben ein Differentialgleichungssystem zweiter Ordnung mit konstanten Koeffizienten erhalten, das wir durch eine einfach Substitution als vier gekoppelte homogene lineare Differentialgleichungen erster Ordnung aufschreiben können, und verlassen hier den Weg von I. Bialyncki-Birula[5], da wir auch später keine quantisierten Felder betrachten wollen. Für einen zu dieser Bewegungsgleichung gehörenden Hamiltonien sei auf die Literatur verwiesen. Die Lösung des Systems ist einfach z.B. über die Eigenwerte und -vektoren der Koeffizientenmatrix zu bestimmen und lautet:

$$\begin{aligned} r =& \Omega C_1 \cos\left(\sqrt{\frac{\Omega^2}{4}+C}\tau\right) + \Omega C_2 \sin\left(\sqrt{\frac{\Omega^2}{4}+C}\tau\right) - \\ & - 2\sqrt{\frac{\Omega^2}{4}-C} C_3 \cos\left(\sqrt{\frac{\Omega^2}{4}-C}\tau\right) - 2\sqrt{\frac{\Omega^2}{4}-C} C_4 \sin\left(\sqrt{\frac{\Omega^2}{4}-C}\tau\right) \qquad (5.27a) \\ s =& 2\sqrt{\frac{\Omega^2}{4}+C} C_1 \sin\left(\sqrt{\frac{\Omega^2}{4}+C}\tau\right) - 2\sqrt{\frac{\Omega^2}{4}+C} C_2 \cos\left(\sqrt{\frac{\Omega^2}{4}+C}\tau\right) - \\ & - \Omega C_3 \sin\left(\sqrt{\frac{\Omega^2}{4}-C}\tau\right) + \Omega C_4 \cos\left(\sqrt{\frac{\Omega^2}{4}-C}\tau\right) \qquad (5.27b) \end{aligned}$$

Dabei wurde die obige Lösung schon in eine reelle Form überführt, sofern $4|C| < \Omega^2$. Der Fall einer sehr guten Feinabstimmung mit $4|C| = \Omega^2$ ist in obiger Lösung nicht enthalten. Der fehlende Lösungsanteil kann aber durch einen linearen Ansatz leicht konstruiert werden. Wir erkennen, dass wir unsere Bewegung in zwei periodische Bewegungen zerlegen können, sofern obige Bedingung für das Verhältnis von Ω^2 und $|C|$ erfüllt ist. Wir schreiben unser Ergebnis in komplexer Form in den Laborkoordinaten auf und erhalten:

$$\begin{aligned} x^1 + \imath x^2 =& e^{-\imath\Omega\tau/2}\left[(\Omega C_1 - 2\Omega_+\imath C_2)\cos(\Omega_+\tau) + (\Omega C_2 + 2\imath\Omega_+ C_1)\sin(\Omega_+\tau) + \right. \\ & \left. + (-2\Omega_- C_3 + \imath\Omega C_4)\cos(\Omega_-\tau) + (-2\Omega_- C_4 - \imath\Omega C_3)\sin(\Omega_-\tau)\right] \qquad (5.28) \end{aligned}$$

Dabei haben wir $\Omega_\pm = \sqrt{\frac{\Omega^2}{4} \pm C}$ abgekürzt. Wir erkennen, dass wir somit vier Frequenzen in der paraxialen Näherung haben, mit der ein Teilchen eine rekurrente Bewegung, projeziert auf die x-y Achse, um die z-Achse ausführt. Dabei wird auch die Vierergeschwindigkeit in 1 und 2 Richtung ein oszillierendes Verhalten zeigen. Da wir an einer beschleunigten Bewegung interessiert waren, wollen wir uns noch über den zeitlichen Verlauf der Komponente u^3 der Vierergeschwindigkeit Klarheit verschaffen und nutzen dafür Gleichung 5.21b:

$$u^3 = \left(c^2 - \left(\frac{\Omega c}{\omega}\right)^2 + (\dot{x}^1)^2 + (\dot{x}^2)^2\right)\frac{\omega}{2\Omega c^2} \qquad (5.29a)$$

$$(5.29b)$$

Man erkennt aufgrund der möglichen Zerlegung von x^1 und x^2 in vier periodische Funktionen, dass u^3 keine effektive Beschleunigung erfährt, sondern in einen konstanten Anteil und und wieder vier periodische Funktionen zerlegbar ist. Damit sehen wir, dass eine effektive Beschleunigung in diesem Grenzfall nicht möglich ist. Die Bahn des Elektrons ist aber hervorzuheben, da diese offensichtlich ein gebundenes Teilchen beschreibt. Wie schon I. Bialyncki-Birula erwähnt[4], ist die Form der Bewegung in z-Richtung hauptsächlich relativistischen Korrekturen geschuldet. Vollführen wir in Gleichung 5.29a den nichtrelativistischen Übergang, indem wir den Grenzfall

$c \to \infty$ betrachten, wobei zu beachten ist, dass dann auch $\Omega \to \omega$ gilt, so ergibt sich:

$$\begin{aligned} cu^3 =& c\left(1-\left(\frac{\Omega}{\omega}\right)^2\right)\frac{\omega}{2\Omega}+O\left(\frac{1}{c}\right) = c\left(\frac{\omega^2(1-(u_0^0)^2-2u_0^0u_0^3-(u_0^3)^2}{\omega^2}\right)/2+O\left(\frac{1}{c}\right) \\ =& u_0^3 c + O\left(\frac{1}{c}\right) = (v_z)_0 + O\left(\frac{1}{c}\right) \end{aligned} \tag{5.30}$$

Somit ist die Geschwindigkeit in z-Richtung bis auf Korrekturen linear und höherer Ordnungen in $\frac{1}{c}$ konstant.

Wir wollen uns noch typische Arten von Projektorien der Trajektorien auf die x-y-Ebene vor Augen führen. Im ersten Fall sei die Bedingung $|C| \gg \Omega^2$ realisiert. Dadurch entsteht im wesentlichen eine Ellipsenbahn(s. Abb. 7), der eine Schwinung aufmoduliert ist. Diese Schwingung äußert sich in einer Bahn ähnlich der einer Zykloide bei einer ausreichend hohen Bahngeschwindigkeit.

Die Lage der Ellipse ändert sich im Laufe der Zeit ähnlich der Periheldrehung eines Planeten

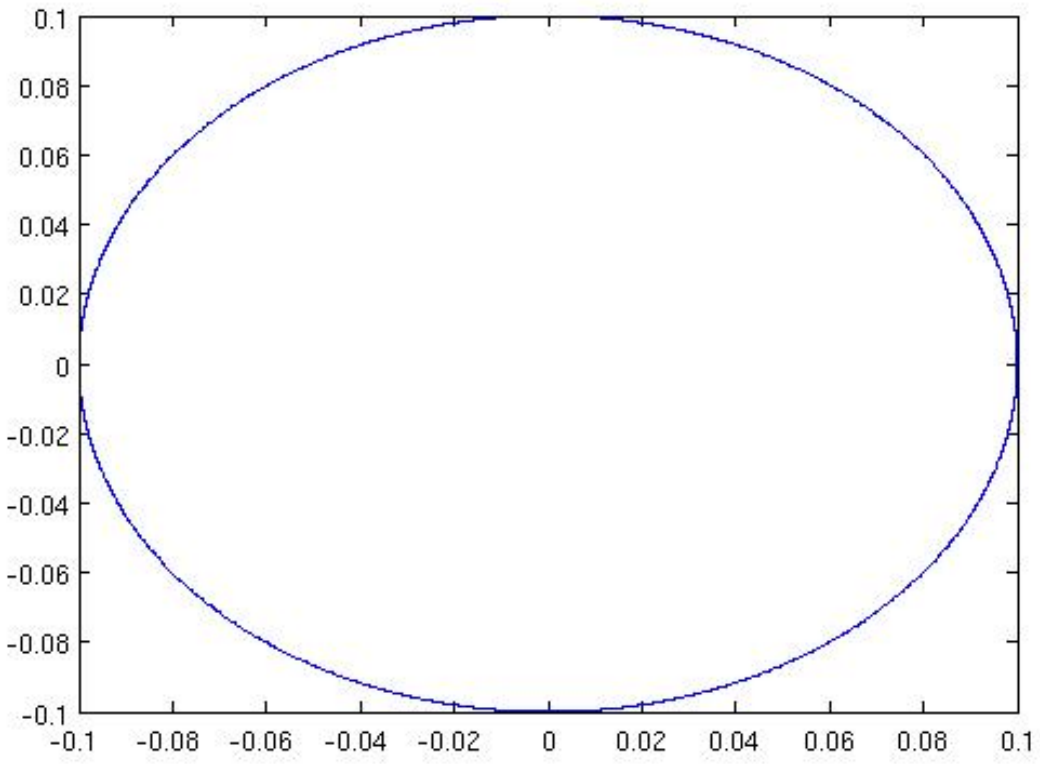

Abbildung 7: Trajektorie eines Testelektrons in der x-y-Ebene gefangen im paraxialen Grenzfall mit folgenden Parametern: $x_0 = 0,1$, $y_0 = 0$, $u_x = 0$, $u_y = 0$, $\epsilon = 1$, $\lambda = 10^{-4}$, $C = -5.8721e24$, $\tau_{end} = 1e7 \cdot \Omega$ mit Längen in cm

um sein Zentralgestirn, hervorgerufen unter anderem durch allgemein relativistische Korrekturen. Die Frequenzen $\Omega - \Omega_\pm$ als auch $\Omega + \Omega_\pm$ sind fast gleich, wodurch es zu Effekten wie bei einer Schwebung kommt. verlgeicht man die Frequenzen $\omega_{schweb} = ||\Omega - \Omega_+| - |\Omega - \Omega_-||/2$ mit der zu beobachtenden Frequenz ω_{beob} ergibt sich folgendes Verhältnis:

$$\omega_{schweb} \approx \omega_{beob} \tag{5.31}$$

Das Verhältnis konnte für verschiedene Anfangsbedingungen in Startorten als auch Startenergien und Feldamplituden mit einer Fehlertoleranz von maximal 5% bestätigt werden. Dies bestätigt die Einschätzung, dass es zu Effekten wie bei einer Schwebung kommt.

Des weiteren kann mit sehr starken Feldern der Fall realisiert werden, dass sich $\Omega^2 \lesssim 4|C|$ ergibt. Wir erhalten dann, dass sich eine scheinbar komplizierte Bewegung ergibt, die aber wieder aus den gleichen Elementen zusammengesetzt ist, wobei aber die Drehung der Ellipse in der Ebene schneller erfolgt und sich so die Trajektorie ergibt. Dies resultiert aus der Tatsache, dass ω_{schweb} zunehmend größer wird (s. Abb. 8). Die Rotationsfrequenz der Ellipse bleibt dabei dem obigem Fall gleich. Wählt man Anfangsbedingungen so, dass zwei Koeffizienten Null sind ergibt sich eine Schwebung zweier Frequenzen.

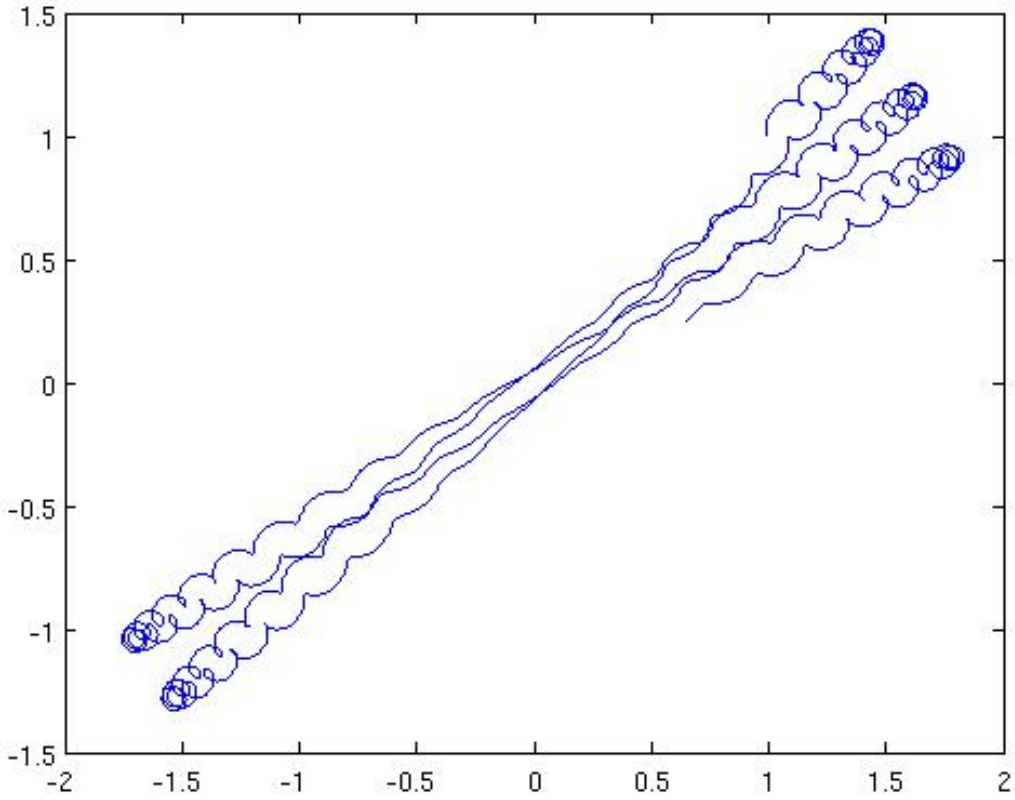

Abbildung 8: Trajektorie eines Testelektrons in der x-y-Ebene gefangen im paraxialen Grenzfall mit folgenden Parametern: $x_0 = 1$, $y_0 = 1$, $u_x = 5$, $u_y = 5$, $\epsilon = 1$, $\lambda = 10^{-4}$, $C = -4.1936e30$, $\tau_{end} = 100 \cdot 2\pi\Omega$ mit Längen in cm. Man erkennt die schnelle Ellipsendrehung in Form der Keulen die sich ausbilden.

Besitzt die Ellipse, der das Teilchen grob folgt, eine große Elliptizität geht die Bahn an den Schnittpunkten mit der großen Halbachse in eine Spiralbahn über.

Haben wir den Fall, dass $4|C| > \Omega^2$ und setzen die entsprechenden Koeffizienten nicht Null erhalten wir einen exponetielles Wachstum in ρ der Komponenten in x^1 und x^2. Dadurch würde die Bedingung der paraxialen Näherung verletzt werden.

5.2 Der Fall M=0

Betrachten wir die Menge, aus der M gewählt werden darf, so nimmt dieser Fall offenbar eine Sonderrolle ein. Während alle anderen M Werte innere oder Randpunkte der Menge der möglichen

M Werte sind, ist 0 der einzige isolierte Punkt. Diese Sonderrole wollen wir im folgenden weiter untersuchen. Im diesem Fall hat unsere Lösung folgende Gestalt:

$$R_x = E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(-k_z \cos(\phi) + \imath \frac{\omega}{c} \sin(\phi)\right) \frac{J_1(k_\perp \rho)}{k_\perp} \tag{5.32}$$

$$R_y = \imath E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(-\frac{\omega}{c} \cos(\phi) + \imath k_z \sin(\phi)\right) \frac{J_1(k_\perp \rho)}{k_\perp} \tag{5.33}$$

$$R_z = E_0 e^{\imath\epsilon(\omega t - k_z z)} \left(\imath\epsilon J_0(k_\perp \rho)\right) \tag{5.34}$$

Wir vergleichen unsere Lösung mit der, die im Skript zur Vorlesung Elektrodynamik von Prof.Wipf [2] für einen einfacheren Ansatz mit Besselwellen angegeben ist:

$$\Phi = \Phi_0 J_0(\alpha\rho) e^{\imath(\omega t - kz)} \ , \ \vec{A} = \frac{\omega}{kc} \Phi \vec{e}_z \tag{5.35}$$

Daraus gewinnen wir folgende Ausdrücke für die Komponenten der Feldstärke Vektoren:

$$\vec{E_V} = \Phi_0 e^{\imath(\omega t - kz)} \begin{pmatrix} \alpha J_1(\alpha\rho)\cos(\phi) \\ \alpha J_1(\alpha\rho)\sin(\phi) \\ J_0(\alpha\rho)\imath\left(\frac{k^2c^2-\omega^2}{kc^2}\right) \end{pmatrix} \tag{5.36a}$$

$$\vec{B_V} = \Phi_0 e^{\imath(\omega t - kz)} \frac{\omega}{kc} \begin{pmatrix} -\alpha J_1(\alpha\rho)\sin(\phi) \\ \alpha J_1(\alpha\rho)\cos(\phi) \\ 0 \end{pmatrix} \tag{5.36b}$$

$$\tag{5.36c}$$

Würden wir formal aus diesen Komponenten erneut einen Riemann-Silberstein Vektor bilden, so ergäbe sich, dass beide Lösungen übereinstimmen, wobei $\alpha = k_\perp$, $k = k_z$, $1 = \epsilon$ und $\Phi_0 = -\frac{k_z}{k_\perp^2} E_0$ in unserer Notation. Man muss allerdings bedenken, dass die Vektoren E_V und E_B keine reellen sondern komplexe Vektoren sind, so dass nicht sie selbst, sondern ihre Realteile zum Riemann-Silberstein Vektor zusammengefasst werden müssen. Dadurch wird aber beim Zusammenfassen nur ein Teil unserer Lösung reproduziert. Dies ist auch vollkommen schlüssig denn aufgrund der Kopplung von Real- und Imaginärteil durch den oszillierenden Ebene-Welle Anteil in $\vec{R}$ kann bei uns die z-Komponente von $\vec{B}$ nicht verschwinden, ohne dass auch diese Komponente von $\vec{E}$ verschwindet, da ein Übergang $t \to t + \frac{\pi}{2\omega}$ die Komponente E_z in $\pm B_z$ überführt. Damit ist es bei beliebiger Wahl der Koeffizienten A,B,C und D ausgeschlossen, dass wir unseren Fall in die andere Lösung überführen.
Man sieht hier aber eine interessante Möglichkeit, wie man in unseren Einheiten und im Vakuum sehr einfach eine zweite Lösung gewinnen kann, wenn man mit komplexen Feldern $\vec{E}$ und $\vec{B}$ arbeitet. Man addiere den Imaginärteil von $\vec{E}$ zum Realteil von $\vec{B}$ und den negativen Imaginärteil von $\vec{B}$ zum Realteil von $\vec{E}$. Da Real und Imaginärteil des elektrischen und magnetischen Feldes seperat die Maxwellgleichungen erfüllen und diese im Vakuum bis auf ein Vorzeichen symmetrisch in beiden Feldern sind, erfüllt die neue Lösung weiterhin die Maxwellgleichungen und wir haben eine neue Lösung konstruiert.
Wir betracheten den oben angesprochenen Fall $k_z \to |\vec{k}| = k$. Es ergibt sich somit folgende Asymptotik:

$$R_x \to -\frac{E_0 k}{2} e^{\imath\epsilon(\omega t - kz)} e^{-\imath\phi} \rho = S_x \tag{5.37a}$$

$$R_y \to -\imath \frac{E_0 k}{2} e^{\imath\epsilon(\omega t - kz)} e^{-\imath\phi} \rho = S_y \tag{5.37b}$$

$$R_z \to E_0 e^{\imath\epsilon(\omega t - kz)} \imath\epsilon = S_z \tag{5.37c}$$

Diese erfüllt weiterhin die Maxwellgleichungen und kann nahe der z-Achse als gute Beschreibung des Feldes genommen werden. Damit haben wir einen Fall realisiert, bei dem wir eine Ausbreitung

der Welle in z Richtung sowie eine exakte Lösung der Maxwellgleichung haben und zugleich die z-Komponente des Elektrischen Feldes nicht verschwindet. Damit ist dieser Fall von besonderem Interesse für die Untersuchung einer beschleunigten Bewegung. Wir wollen untersuchen ob es möglich ist, dass ein Teilchen nahe der z-Achse verbleibt, während es eine Beschleunigung in diese Richtung erfährt.
Dazu untersuchen wir die Bewegungsgleichungen:

$$\dot{u^0} = \frac{e}{mc}(E_x u^1 + E_y u^2 + E_z u^3) \tag{5.38a}$$

$$\dot{u^1} = \frac{e}{mc}(E_x(u^0 - u^3) + B_z u^2) \tag{5.38b}$$

$$\dot{u^2} = \frac{e}{mc}(E_y(u^0 - u^3) - B_z u^1) \tag{5.38c}$$

$$\dot{u^3} = \frac{e}{mc}(E_x u^1 + E_y u^2 + E_z u^0) \tag{5.38d}$$

Wir kombinieren Gleichung 5.38a und 5.38d indem wir sie voneinander subtrahieren und erhalten, nachdem wir E_z eingesetzt haben, folgende Differentialgleichung:

$$\ddot{\eta} = D\dot{\eta}\sin(\eta) \tag{5.39}$$

Wir haben dabei schon $D = \frac{eE_0}{mc}$ und $k(x^0 - x^3) = \eta$ abgekürzt. Wir können einmal direkt integrieren und erhalten:

$$\frac{d\eta}{d\tau} = -D(\cos(\eta) + C) \tag{5.40}$$

Dabei ist $-CD$ unsere frei wählbare Integrationskonstante, welche die Geschwindigkeit von η zur Zeit τ=0 beschreibt. Diese kann so geschrieben werden, da $D = 0$ zu $E_0 = 0$ und somit zu einer trivialen nicht interessanten Lösung führt. Wir wollen uns, ohne die Allgemeinheit zu beschränken, in unserem Laborsystem vorgeben, dass zum Eigenzeitpunkt $\tau = 0$ $\eta = 0$ gilt. Ansonsten führen wir eine einfache Verschiebung des Koordinatensystems in z-Richtung durch und erhalten obigen Fall. Zunächst wird der Wertebereich von C genauer betrachtet. Wir schätzen die rechte Seite von 5.40 in zwei Richtungen ab und kommen zu folgendem Ergebnis:

$$\dot{\eta} = ck\dot{t} - k\dot{z} > 0 \quad \Leftrightarrow \quad \frac{dz}{dt} < c \tag{5.41}$$

$$\dot{\eta} = ck\dot{t} - k\dot{z} \leq 0 \quad \Leftrightarrow \quad \frac{dz}{dt} \geq c \tag{5.42}$$

Offensichtlich ist der Fall, dass die Geschwindigkeit von z bezogen auf die Koordinatenzeit größer oder gleich der Lichgeschwindigkeit c ist, für ein massebehaftetes Teilchen kein physikalisch sinnvoller. Setzen wir noch, dass D eine positive Konstante ist, folgt, dass C<-1 gelten muss, während eine negatives D ein C>-1 erfordert. Interessant ist, dass, somit der Fall bei dem C nur wenig größer als 1 ist, den Wertebereich von η deutlich einschränkt. Wir lösen im folgenden obige Differentialgleichung.

$$\tau = -\frac{1}{D}\int_0^{\eta} \frac{1}{\cos(\eta') + C} d\eta' \tag{5.43}$$

Die Lösung, umgeschrieben auf eine Funktion $\eta(\tau)$, lautet für $|C| > 1$:

$$\eta = -2\arctan\left(\sqrt{\frac{C+1}{C-1}}\tan\left(\frac{D}{2}\sqrt{C^2-1}\tau\right)\right) \tag{5.44}$$

Dabei ist jeweils der Zweig des arctan zu nehmen, so dass $\eta(\tau)$ eine stetige Funktion bleibt, und bei $\tau = 0$ ist der Hauptzweig zu wählen, so dass die Anfangsbedingung erfüllt ist. Im Falle $|C| <= 1$ erhalten wir folgenden Ausdruck:

$$\eta = -2 \arctan \sqrt{\frac{1+C}{1-C}} \tanh\left(\frac{D}{2}\sqrt{1-C^2}\tau\right) \tag{5.45}$$

Mittels dieser Formeln berechnen wir mehrere Trajektorien für verschiedene Parameter und betrachten, ob wir einen paraxialen Grenzfall realisieren können. Es stellt sich aber heraus, dass die Trajektorien in ihrer Radialkomponente ein exponentielles Wachstum aufweisen, während das Wachstum der Energie gering ist. In Abb. 9 erkennt man die beschleunigte Bewegung in der Radialkomponente. Betrachtet man Felder, bei denen C in der Nähe von eins liegt Abb. 10, ergibt sich ein noch schnelleres Wachstum. Die Frage, die hier nicht beantwortet wird, ist wie weit man die Besselwelle in guter Näherung durch obige Approximation beschreiben kann. Sollte dies über eine große Anzahl von Wellenlängen der Fall sein so ist eine effektive Beschleunigung gut möglich. Man beachte aber, dass hier eine sehr große Feldstärke angelegt wurde. Der Wert von $D = -1 \cdot 10^{15}$ korrespondiert mit einer Feldstärke von $E_0 = 1,6 \cdot 10^7 \frac{esu}{cm^2} = 1,7 \cdot 10^{12} \frac{V}{m}$. Dieser Wert ist derzeit wohl kaum zu realisieren. Man beachte trotzdem den exponentiell Energieanstieg wie man ihn in Abb. 11 sehen kann. Man bedenke, dass ρ in Einheiten der Wellenlänge aufgetragen ist.
Man muss somit sagen, dass eine Beschleunigung erzielbar ist, aber damit auch eine Beschleunigung aus dem paraxialen Grenzfall hinaus verbunden ist.

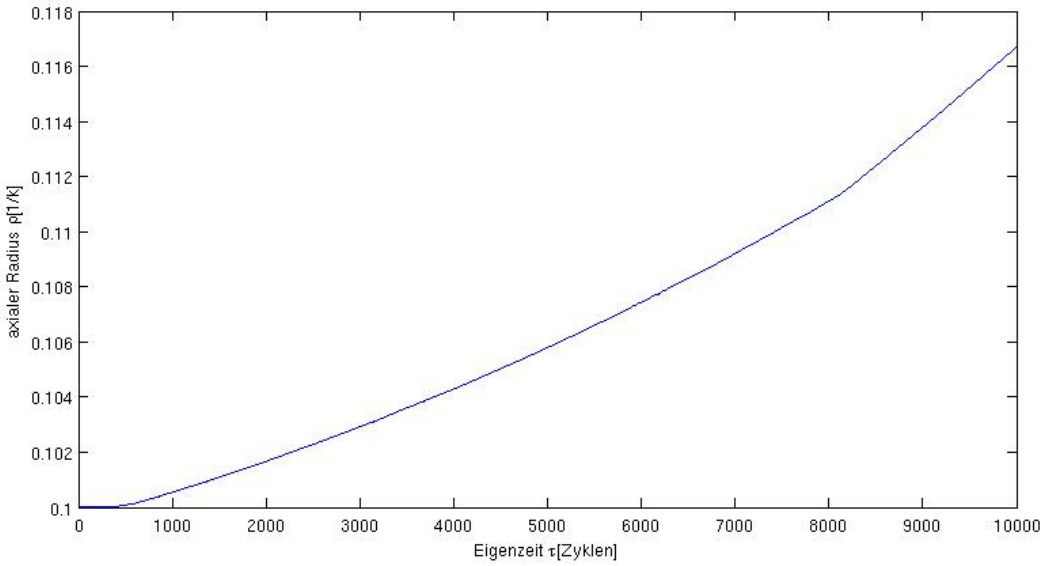

Abbildung 9: Radialkomponente der Trajektorie eines Testelektrons aufgetragen über der Eigenzeit. Folgende Startparameter wurden gewählt: $x_0 = 0.1 * k$, $y_0 = 0$ $z_0 = 0$, $u^0 = 1$, $\epsilon = 1$, $\lambda = 10^{-4} cm$, $D = -1 \cdot 10^{10} 1/s$, $C = 1 \cdot 10^5$, $\tau_{end} = 10000 \cdot 2\pi\Omega$.

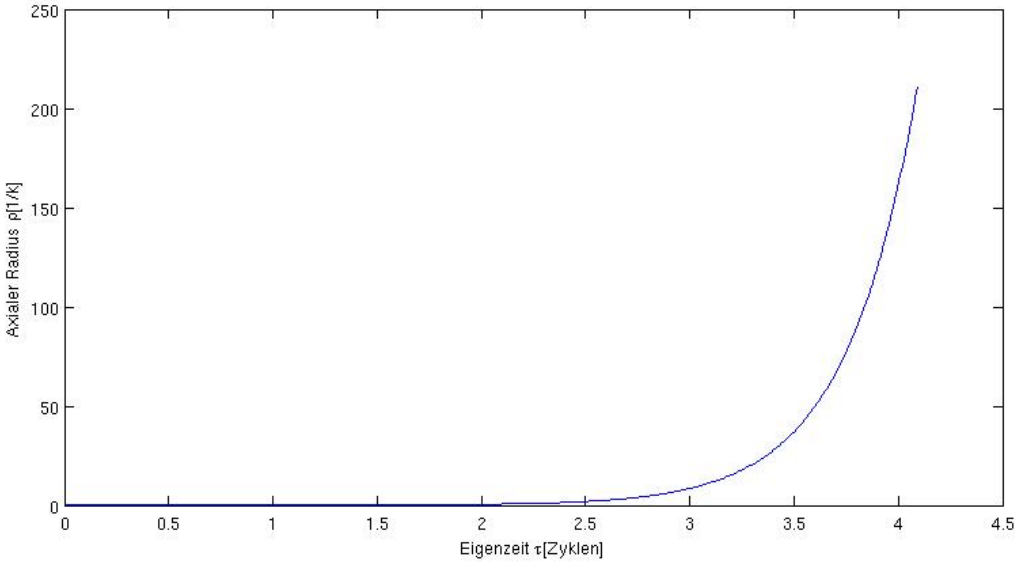

Abbildung 10: Radialkomponente der Trajektorie eines Testelektrons aufgetragen über der Eigenzeit. Folgende Startparameter wurden gewählt: $x_0 = 0.1 * k$, $y_0 = 0$ $z_0 = 0$, $u_0^0 = 1$, $\epsilon = 1$, $\lambda = 10^{-4} cm$, $D = -1 \cdot 10^{15} 1/s$, $C = 0,8837$, $\tau_{end} = 10000 \cdot 2\pi\Omega$.

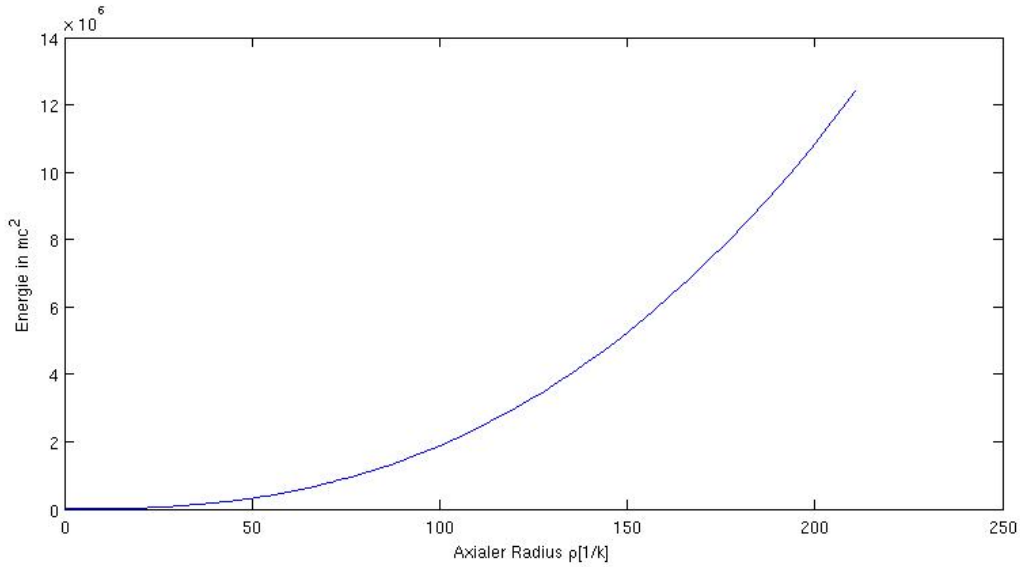

Abbildung 11: Energie des Testelektrons aufgetragen über der Radialkomponente. Folgende Startparameter wurden gewählt: $x_0 = 0.1 * k$, $y_0 = 0$ $z_0 = 0$, $u_0^0 = 1$, $\epsilon = 1$, $\lambda = 10^{-4} cm$, $D = -1 \cdot 10^{15} 1/s$, $C = 0,8837$, $\tau_{end} = 10000 \cdot 2\pi\Omega$.

Literatur

[1] Andreas Wipf; Skript zum Vortrag an der Chris Engelbrecht Sommer School in Theoretical Physics, January 16-28, 2010

[2] Andreas Wipf; Skript zur Vorlesung Elektrodynamik an der Friedrich Schiller Universität Jena; http://www.physik-jena.de/login/skripte/skripte/skript _theoretischeelektrodynamik_ws0607_profdrandreaswipf_.pdf 22.09.2010 18:00

[3] Yousef I. Slamin, Guido R. Mocken und Christoph H.Keitel; Electron scattering and acceleration by a tightly focused laser beam; Physical Review Special Topics - Accelrators and beams, Volume 5, 101301 (2002)

[4] Iwo Bialynicki-Birula; Trojan states of electrons guided by Bessel beams; Laser Physics 15, 1371-1380 (2005)

[5] Iwo Bialynicki-Birula; Particle Beams guided by electromagnetic vortices: New Solutions of the Lorentz, Schrödinger, Klein-Gordon and Dirac equations; Phys.Rev.Lett. 93 (2004) 020402

[6] Field Theory Handbook; Parry Moon, Domina Eberle Spencer; Springer Verlag Berlin Göttingen Heidelberg 1961

6 Anhang

Ein weiterer Teil der Arbeit ist die Grafische Benutzeroberfläche Grafik.fig, die es erlaubt für die oben diskutierten Felder die Trajektorien für verschiedene Parameter darzustellen. Zur Ausführung wird das Programm Matlab benötigt und Matlab benötigt Schreibrechte im Arbeitsverzeichnis, da die erzeugten aktuellen Datensätze, für jeden Fall einzeln, in einer jeweils entsprechend dem Fall benannten Datei gespeichert werden. Die Benutzeroberfläche ist in Abb. 12 zu sehen. Geöffnet wird das Programm über Matlab. Dazu wird Matlab gestartet und als Arbeitsverzeichnis wird der Unterordner gewählt, in dem sich Grafik.fig befindet. Diese Datei ist auf der DVD im Ordner Programme. Anschließend wird mittels des Befehls

```
run Grafik
```

die Benutzeroberfläche gestartet.

6.1 Eingabe

Zuerst muss der entsprechende Fall ausgewählt werden. Danach kann gegebenfalls noch ein Unterpunkt gewählt werden.

6.1.1 Ebene Wellen und Besselwellen

Die Eingabe der Längen erfolgt in Einheiten des inversen Betrages des Wellenzahlvektors k, d.h. einzugeben ist, statt der dimensionsbehafteten Größe x, die dimensionslos Größe $x \cdot k$. Die Energie des Teilchens zum Zeitpunkt $\tau = 0$ erfolgt nur durch die Wahl des Parameters Startenergie. Der räumliche Anteil des Vierergeschwindigkeitsvektors wird entsprechend der Energie-Impuls Beziehung normiert. Die Eingabe der Endzeit erfolgt in Einheiten von $2 * \pi/\Omega$ gemessen in der Eigenzeit. Dabei gilt $\Omega = \omega(u_0^0 - u_0^3)$. Die Eingabe der Ladung und der Masse erfolgt in Einheiten der Elementarladung respektive der Elektronenmasse. Somit hat ein ruhendes Elektron die Ladung -1, die Masse 1 und die Eigenenergie 1. Die Intensität E_0 für Besselwellen wird in Einheiten von $e/(mc)$ eingegeben, während die Eingabe von α_0 bei Ebenen Wellen in Einheiten von $e/(mc^2)$ erfolgt. Die Umrechnung zwischen beiden Grössen ist $E_0 = \omega\alpha_0$. Die Eingabe der Wellenlänge erfolgt aufgrund des cgs Einheitensystems in cm. Sofern Eingaben erfolgen, die nicht sinnvoll sind, wie eine Startenergie kleiner eins, wird der Wert auf einen vorher definierten Wert zurückgesetzt. Alle Werte sind am Anfang so gesetzt, dass ein sinnvoller Fall entsteht. Für ebene Wellen erfolgt noch eine Eingabe der Polaristation δ, einem Wert zwischen 0 und 1 entsprechend der Art der Polarisation, wobei 0 einer Polarisation in x-Richtung entspricht, 1 einer in y Richtung und $\sqrt{2}/2$ einer zirkularen Polarisation. Im Falle der ebenen gedämpften Welle ist noch der Dämpfungsfaktor in Einheiten von 1/k einzugeben. Für die Besselwellen ist noch der Drehsinn einzugeben, wobei ein Wert von ± 1 erwartet wird.

6.1.2 Gaußstrahl

Bei dieser Option werden die Koordinaten in x und y in Einheiten von w_0 eingegeben, während die Koordinaten in z und s in Einheiten von z_0 eizugeben sind, d.h. es werden effektiv $\xi\ \eta\ \zeta$ eingegeben. Zur Orientierung wird aus den in cm einzugebenden Grössen w_0 und λ noch z_0 berechnet. Aus der Energie, eingegeben in Einheiten von mc^2, Startkoordinaten und dem Zielpunkt auf der z-Achse wird der Geschwindigkeitsvektor berechnet. Die eingegebene Endzeit entspricht der Anzahl an Perioden gemessen in ω in der Eigenzeit. Die tatsächlich gerechnete Zeit kann kleiner als die angegebene sein, da die Rechnung beendet wird, sobald das Teilchen einen Abstand zur z-Achse vom zehnfachen des Abstandes Strahlweite z-Achse hat und sich vorher im Strahl befunden hatte. Ladung und Masse werden in Einheiten der Elementarladung bzw. Elektronenmasse angegeben. Die Eingabe der Intensität erfolgt in Einheiten von $e/(mc^2)$.

6.2 Ausgabe

Nachdem der Knopf Start gedrückt wurde, wird die Trajektorie berechnet und ausgegeben. Dabei erfolgt die Ausgabe in zwei Diagrammen. Die Einstellung für ebene Wellen, ist der Art, dass auf der linken Seite die Trajektorie eines Teilchens, dass Anfangs die gegebene Energie hatte, und rechts der Verlauf der Energie über der Eigenzeit des Teilchens. Bei Besselwellen, wird auf der linken Seite die komplette Trajektorie in der x-y-Ebene ausgegeben, während auf der rechten Seite auch die Energie über der Eigenzeit dargestellt wird. Bei Gauß-Wellen wird links die komplette Trajektorie dargestellt, während auf der rechten Seite ein Ausschnitt der Trajektorie dargestellt wird, bei dem das Teilchen sich in der Nähe der Strahlweite befunden hat. Je nach Verlauf der Trajektorie, kann aber auch die komplette Trajektorie dargestellt sein. Die Farbkodierung ist derart, dass die Strahlweite rot und die Trajektorie rot ist. Man erkennt dann, ob der Startort womöglich schon im Strahl lag oder die Wechselwirkung weit vom Strahl entfernt stattfand.
Alle Darstellungen können aber vom Benutzer gewechselt werden. Dabei bestehen die Varianten die Raumkoordinaten, die Eigenzeit oder die Energie des Teilchens auszuwählen. Auch besteht die Option in den Diagrammen den Achsenausschnitt auszuwählen. Es besteht des weiteren die Möglichkeit mit der Maus in die Grafik hineinzuzoomen. Sofern keine Änderungen an den Achseneinstellungen vorgenommen wurden, ist es möglich mit Plot wieder das Gesamtbild zu sehen. Wird bei der Applikate der Punkt 2-D Plot gewählt, erfolgt die Ausgabe eines zwei-dimensionalen Diagramms. Im Falle einer anderen Auswahl wird ein drei-dimensionales Diagramm ausgegeben. In der unteren linken Ecke wird die Energie des Teilchens angegeben, nachdem es die entsprechende Anzahl Zyklen durchlaufen hat. Bei einer ebenen Welle wird auf diese Anzeige verzichtet.

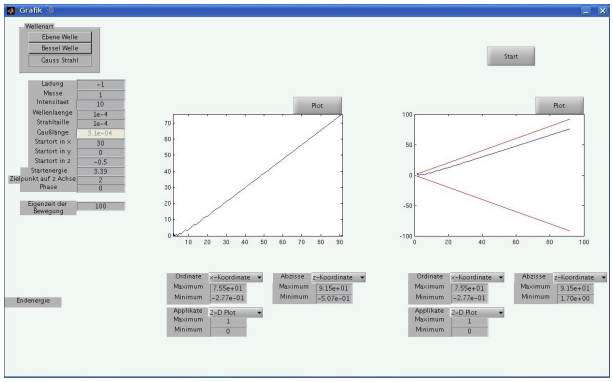

Abbildung 12: Dargestellt ist die Benutzeroberfläche nachdem für die Standardeinstellungen eines Gaußstrahls die Rechnung durchgeführt wurde.

6.3 Programm-Code

Listing 1: Hauptprogramm

```
1  function varargout = Grafik(varargin)
2
3  gui_Singleton = 1;
4  gui_State = struct('gui_Name',       mfilename, ...
5                     'gui_Singleton',  gui_Singleton, ...
6                     'gui_OpeningFcn', @Grafik_OpeningFcn, ...
7                     'gui_OutputFcn',  @Grafik_OutputFcn, ...
8                     'gui_LayoutFcn',  [] , ...
9                     'gui_Callback',   []);
10 if nargin && ischar(varargin{1})
```

```
11      gui_State.gui_Callback = str2func(varargin{1});
12  end
13
14  if nargout
15      [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
16  else
17      gui_mainfcn(gui_State, varargin{:});
18  end
19
20  function Grafik_OpeningFcn(hObject, eventdata, handles, varargin)
21  handles.output = hObject;
22  guidata(hObject, handles);
23
24  function varargout = Grafik_OutputFcn(hObject, eventdata, handles)
25  varargout{1} = handles.output;
26
27  function beginn_Callback(hObject, eventdata, handles)
28  zoom on
29  start=zeros(8,1);
30  set(handles.text20,'Visible','on');
31  flag=0;
32  pause(0.01)
33
34  if get(handles.Unter1,'Value')==1
35      flag=1;
36  elseif get(handles.Unter2,'Value')==1
37      flag=2;
38
39  end
40  if (get(handles.bessel,'Value'))==1
41      start(1)=0;
42      start(2)=str2double(get(handles.startx,'String'));
43      start(3)=str2double(get(handles.starty,'String'));
44      start(4)=str2double(get(handles.startx,'String'));
45      start(5)=str2double(get(handles.startE,'String'));
46      start(6)=str2double(get(handles.startvx,'String'));
47      start(7)=str2double(get(handles.startvy,'String'));
48      start(8)=str2double(get(handles.startvz,'String'));
49      nq=str2double(get(handles.ladung,'String'));
50      nm=str2double(get(handles.mass,'String'));
51      epsilon=str2double(get(handles.eps,'String'));
52      lambda=str2double(get(handles.lambda,'String'));
53      E_0=str2double(get(handles.intens,'String'))*1e7;
54      tend=str2double(get(handles.Endzeit,'String'));
55      if norm(start(6:8))~=0
56          start(6:8)=start(6:8)/norm(start(6:8))*sqrt(start(5)^2-1);
57      else
58          start(5)=1;
59          set(handles.startE,'String','1');
60      end
61      if flag==1
```

```
62          [tau rundu]=Besselrechnen2(nq,nm,epsilon,lambda,E_0,start,tend);
63          save Bessel2
64          set(handles.text22,'String',num2str(rundu(end,5)));
65      end
66      if flag==2
67          [tau rundu]=initbessel0(start,nq,lambda,nm,epsilon,E_0,tend);
68          save Bessel0
69          set(handles.text22,'String',num2str(rundu(end,5)));
70      end
71          plot(handles.dia1,rundu(:,2),rundu(:,3));
72          xlim(handles.dia1,[min(rundu(:,2)) max(rundu(:,2))]);
73          ylim(handles.dia1,[min(rundu(:,3)) max(rundu(:,3))]);
74          set(handles.ordmin1,'String',num2str(min(rundu(:,3)),'%4.2e'));
75          set(handles.ordmax1,'String',num2str(max(rundu(:,3)),'%4.2e'));
76          set(handles.abzmin1,'String',num2str(min(rundu(:,2)),'%4.2e'));
77          set(handles.abzmax1,'String',num2str(max(rundu(:,2)),'%4.2e'));
78          set(handles.appmin1,'String','0');
79          set(handles.appmax1,'String','1');
80          set(handles.ord1,'Value',3)
81          set(handles.abz1,'Value',2)
82          set(handles.app1,'Value',1)
83
84          plot(handles.dia2,tau(:),rundu(:,5));
85          xlim(handles.dia2,[0 tau(end)]);
86          Emin=min(rundu(:,5));
87          if(min(rundu(:,5))==max(rundu(:,5)))
88              Emax=Emin+1;
89          else
90              Emax=max(rundu(:,5));
91          end
92          ylim(handles.dia2,[Emin Emax]);
93          set(handles.ordmin2,'String',num2str(min(rundu(:,5)),'%4.2e'));
94          set(handles.ordmax2,'String',num2str(max(rundu(:,5)),'%4.2e'));
95          set(handles.abzmin2,'String',num2str(0,'%4.2e'));
96          set(handles.abzmax2,'String',num2str(tau(end),'%4.2e'));
97          set(handles.appmin2,'String','0');
98          set(handles.appmax2,'String','1');
99          set(handles.ord2,'Value',5)
100         set(handles.abz2,'Value',1)
101         set(handles.app2,'Value',1)
102         set(handles.text22,'String',num2str(rundu(end,5)));
103
104 elseif (get(handles.eben,'Value'))==1
105     start(1)=0;
106     start(2)=str2double(get(handles.startx,'String'));
107     start(3)=str2double(get(handles.starty,'String'));
108     start(4)=str2double(get(handles.startx,'String'));
109     start(5)=str2double(get(handles.startE,'String'));
110     start(6)=str2double(get(handles.startvx,'String'));
111     start(7)=str2double(get(handles.startvy,'String'));
112     start(8)=str2double(get(handles.startvz,'String'));
```

```
113     if norm(start(6:8))~=0
114         start(6:8)=start(6:8)/norm(start(6:8))*sqrt(start(5)^2-1);
115     else
116         start(5)=1;
117         set(handles.startE,'String','1');
118     end
119     lambda=str2double(get(handles.lambda,'String'));
120     alpha0=str2double(get(handles.intens,'String'));
121     tend=str2double(get(handles.Endzeit,'String'));
122     xi=str2double(get(handles.startct,'String'));
123     delta=str2double(get(handles.eps,'String'));
124     nq=str2double(get(handles.ladung,'String'));
125     nm=str2double(get(handles.mass,'String'));
126     if flag ~=0
127         [tauruh rruh]=initeben(start,tend,nq,nm,alpha0...
128             ,lambda,flag,xi,delta);
129 
130         plot3(handles.dia1,rruh(:,2),rruh(:,3),rruh(:,4));
131         xmin=min(rruh(:,2));
132         xmax=max(rruh(:,2));
133         ymin=min(rruh(:,3));
134         ymax=max(rruh(:,3));
135         zmin=min(rruh(:,4));
136         zmax=max(rruh(:,4));
137         if xmin==xmax
138             xmin=xmin-0.5;
139             xmax=xmax+0.5;
140         end
141         if ymin==ymax
142             ymin=ymin-0.5;
143             ymax=ymax+0.5;
144         end
145         if zmin==zmax
146             zmin=zmin-0.5;
147             zmax=zmax+0.5;
148         end
149         xlim(handles.dia1,[xmin xmax]);
150         ylim(handles.dia1,[ymin ymax]);
151         zlim(handles.dia1,[zmin zmax]);
152         set(handles.ordmin1,'String',num2str(ymin,'%4.2e'));
153         set(handles.ordmax1,'String',num2str(ymax,'%4.2e'));
154         set(handles.abzmin1,'String',num2str(xmin,'%4.2e'));
155         set(handles.abzmax1,'String',num2str(xmax,'%4.2e'));
156         set(handles.appmin1,'String',num2str(zmin,'%4.2e'));
157         set(handles.appmax1,'String',num2str(zmax,'%4.2e'));
158 
159         set(handles.ord1,'Value',3)
160         set(handles.abz1,'Value',2)
161         set(handles.app1,'Value',5)
162         plot(handles.dia2,tauruh(:),rruh(:,5));
163         xmin=min(tauruh);
```

```
164         xmax=max(tauruh);
165         ymin=min(rruh(:,5));
166         ymax=max(rruh(:,5));
167 
168         if xmin==xmax
169             xmin=xmin-0.5;
170             xmax=xmax+0.5;
171         end
172         if ymin==ymax
173             ymin=ymin-0.5;
174             ymax=ymax+0.5;
175         end
176 
177         xlim(handles.dia2,[xmin xmax]);
178         ylim(handles.dia2,[ymin ymax]);
179 
180         set(handles.ordmin2,'String',num2str(ymin,'%4.2e'));
181         set(handles.ordmax2,'String',num2str(ymax,'%4.2e'));
182         set(handles.abzmin2,'String',num2str(xmin,'%4.2e'));
183         set(handles.abzmax2,'String',num2str(xmax,'%4.2e'));
184 
185         set(handles.ord2,'Value',5)
186         set(handles.abz2,'Value',1)
187         set(handles.app2,'Value',1)
188         save eben
189     end
190     set(handles.text22,'Visible','off');
191 
192 elseif (get(handles.gauss,'Value'))==1
193     nq=str2double(get(handles.ladung,'String'));
194     nm=str2double(get(handles.mass,'String'));
195     nx=str2double(get(handles.startx,'String'));
196     nz=str2double(get(handles.startz,'String'));
197     ny=str2double(get(handles.starty,'String'));
198     gammaa=str2double(get(handles.startE,'String'));
199     lambda=str2double(get(handles.lambda,'String'));
200     LasE=str2double(get(handles.intens,'String'));
201     LasE=LasE*nq/nm;
202     tend=str2double(get(handles.Endzeit,'String'));
203     s=str2double(get(handles.startvx,'String'));
204     w0=str2double(get(handles.eps,'String'));
205     z0=str2double(get(handles.startct,'String'));
206     psi_0=str2double(get(handles.startvy,'String'));
207     [tau rundu flag ye1 ye2 ye3 Laengeneu]=initgauss...
208         (nx,nz,ny,tend,gammaa,lambda,w0,LasE,psi_0,s);
209     taille=sqrt(1+rundu(:,4).^2/z0^2);
210     t_eins=find(tau==ye1(1));
211     if isempty(t_eins)
212         t_eins=1;
213     end
214     t_zwei=find(tau==ye2(1));
```

```
215     if isempty(t_zwei)
216         t_zwei=1;
217     end
218     save Gauss
219     axes(handles.dia1);
220     rundu(:,2:3)=rundu(:,2:3)/w0;
221     rundu(:,4)=rundu(:,4)/z0;
222     plot(rundu(:,4),rundu(:,2));
223     xmin1=min(rundu(:,2));
224     xmax1=max(rundu(:,2));
225     zmin1=min(rundu(:,4));
226     zmax1=max(rundu(:,4));
227     if xmin1==xmax1
228             xmin1=xmin1-0.5;
229             xmax1=xmax1+0.5;
230     end
231
232     if zmin1==zmax1
233             zmin1=zmin1-0.5;
234             zmax1=zmax1+0.5;
235     end
236
237     xlim(handles.dia1,[zmin1 zmax1]);
238     ylim(handles.dia1,[xmin1 xmax1]);
239     set(handles.ordmin1,'String',num2str(xmin1,'%4.2e'));
240     set(handles.ordmax1,'String',num2str(xmax1,'%4.2e'));
241     set(handles.abzmin1,'String',num2str(zmin1,'%4.2e'));
242     set(handles.abzmax1,'String',num2str(zmax1,'%4.2e'));
243     set(handles.appmin1,'String','0');
244     set(handles.appmax1,'String','1');
245     set(handles.ord1,'Value',2)
246     set(handles.abz1,'Value',4)
247     set(handles.app1,'Value',1)
248     axes(handles.dia2);
249     if  (t_zwei(1)==1)
250             t_zwei=size(tau,2);
251     end
252
253     plot(handles.dia2,rundu(t_eins(1):t_zwei(1),4)...
254         ,rundu(t_eins(1):t_zwei(1),2));
255     hold on
256     plot(handles.dia2,rundu(t_eins(1):t_zwei(1),4)...
257         ,taille(t_eins(1):t_zwei(1)),'r');
258     plot(handles.dia2,rundu(t_eins(1):t_zwei(1),4)...
259         ,-taille(t_eins(1):t_zwei(1)),'r');
260     xmin2=min(rundu(t_eins:t_zwei,2));
261     xmax2=max(rundu(t_eins:t_zwei,2));
262     zmin2=min(rundu(t_eins:t_zwei,4));
263     zmax2=max(rundu(t_eins:t_zwei,4));
264     if xmin2==xmax2
265             xmin2=xmin2-0.5;
```

```
266                 xmax2=xmax2+0.5;
267     end
268
269     if zmin2==zmax2
270                 zmin2=zmin2-0.5;
271                 zmax2=zmax2+0.5;
272     end
273     xlim(handles.dia2,[zmin2 zmax2]);
274     ylim(handles.dia2,[xmin2 xmax2]);
275     set(handles.ordmin2,'String',num2str(xmin2,'%4.2e'));
276     set(handles.ordmax2,'String',num2str(xmax2,'%4.2e'));
277     set(handles.abzmin2,'String',num2str(zmin2,'%4.2e'));
278     set(handles.abzmax2,'String',num2str(zmax2,'%4.2e'));
279     set(handles.appmin2,'String','0');
280     set(handles.appmax2,'String','1');
281     set(handles.ord2,'Value',2)
282     set(handles.abz2,'Value',4)
283     set(handles.app2,'Value',1)
284     hold off
285
286     set(handles.text22,'String',num2str(rundu(end,5)));
287 end
288
289 set(handles.text20,'Visible','off');
290
291
292
293 function pushbutton2_Callback(hObject, eventdata, handles)
294 if get(handles.Unter1,'Value')==1
295     flag=1;
296 elseif get(handles.Unter2,'Value')==1
297     flag=2;
298 end
299 if (get(handles.bessel,'Value'))==1
300     if flag==1
301         open Bessel2.mat;
302         zeichne(hObject,eventdata,handles,ans.rundu,ans.tau,1);
303     end
304     if flag==2
305         open Bessel0.mat;
306         zeichne(hObject,eventdata,handles,ans.rundu,ans.tau,1);
307     end
308
309
310 elseif (get(handles.eben,'Value'))==1
311     open eben.mat;
312     zeichne(hObject,eventdata,handles,ans.rruh,ans.tauruh,1);
313
314 elseif (get(handles.gauss,'Value'))==1
315     open Gauss.mat;
316     zeichne(hObject,eventdata,handles,ans.rundu,ans.tau,1);
```

```
317 end
318
319
320
321
322
323
324 function pushbutton3_Callback(hObject, eventdata, handles)
325 if get(handles.Unter1,'Value')==1
326     flag=1;
327 elseif get(handles.Unter2,'Value')==1
328     flag=2;
329 end
330 if (get(handles.bessel,'Value'))==1
331     if flag==1
332         open Bessel2.mat;
333         zeichne(hObject,eventdata,handles,ans.rundu,ans.tau,2);
334     end
335     if flag==2
336         open Bessel0.mat;
337         zeichne(hObject,eventdata,handles,ans.rundu,ans.tau,2);
338     end
339
340
341 elseif (get(handles.eben,'Value'))==1
342     open eben.mat;
343
344     zeichne(hObject,eventdata,handles,ans.rruh,ans.tauruh,2);
345
346 elseif (get(handles.gauss,'Value'))==1
347     open Gauss.mat;
348     zeichne(hObject,eventdata,handles,ans.rundu,ans.tau,2);
349 end
350
351 function ladung_Callback(hObject, eventdata, handles)
352 nq=str2num(get(hObject,'String'));
353 if (isempty(nq))
354      set(hObject,'String','-1')
355 elseif nq==0
356     set(hObject,'String','-1')
357 elseif nq~=str2double(get(hObject,'String'))
358     set(hObject,'String',num2str(nq))
359 end
360 guidata(hObject, handles);
361
362
363
364
365 function ladung_CreateFcn(hObject, eventdata, handles)
366 if ispc && isequal(get(hObject,'BackgroundColor'),...
367         get(0,'defaultUicontrolBackgroundColor'))
```

```
368     set(hObject,'BackgroundColor','white');
369 end
370
371
372
373 function mass_Callback(hObject, eventdata, handles)
374 nm=str2num(get(hObject,'String'));
375 if (isempty(nm))
376       set(hObject,'String','1')
377 elseif nm<=0
378     set(hObject,'String','1')
379 elseif nm~=str2double(get(hObject,'String'))
380     set(hObject,'String',num2str(nm))
381 end
382 guidata(hObject, handles);
383
384
385 function mass_CreateFcn(hObject, eventdata, handles)
386 if ispc && isequal(get(hObject,'BackgroundColor'),...
387         get(0,'defaultUicontrolBackgroundColor'))
388     set(hObject,'BackgroundColor','white');
389 end
390
391
392
393 function lambda_Callback(hObject, eventdata, handles)
394 number=str2num(get(hObject,'String'));
395 if (isempty(number))
396       set(hObject,'String','1e-4')
397 elseif number<=0
398     set(hObject,'String','1e-4')
399 elseif number~=str2double(get(hObject,'String'))
400     set(hObject,'String',num2str(number))
401 end
402 if strcmp(get(handles.text5,'String'),'Strahltaille')==1
403     rechnez0(hObject, eventdata, handles)
404 end
405 guidata(hObject, handles);
406
407
408 function lambda_CreateFcn(hObject, eventdata, handles)
409 if ispc && isequal(get(hObject,'BackgroundColor'),...
410         get(0,'defaultUicontrolBackgroundColor'))
411     set(hObject,'BackgroundColor','white');
412 end
413
414
415
416 function eps_Callback(hObject, eventdata, handles)
417 number=str2num(get(hObject,'String'));
418 if (isempty(number))
```

```
419     if strcmp(get(handles.text5,'String'),'Drehrichtung')==1
420         set(hObject,'String','1');
421         number=1;
422     elseif strcmp(get(handles.text5,'String'),'Polarisation')==1
423          set(hObject,'String','0')
424          number=0;
425     elseif strcmp(get(handles.text5,'String'),'Strahltaille')==1
426         set(hObject,'String','1e-4');
427         number=1e-4;
428     end
429 end
430 if strcmp(get(handles.text5,'String'),'Drehrichtung')==1
431     if number>=0
432        set(hObject,'String','1')
433        number=1;
434     else
435         set(hObject,'String','-1')
436         number=-1;
437     end
438
439 elseif strcmp(get(handles.text5,'String'),'Polarisation')==1
440     if number<=0
441          set(hObject,'String','0')
442          number=0;
443     elseif number>=1
444         set(hObject,'String','1')
445         number=1;
446     end
447 elseif strcmp(get(handles.text5,'String'),'Strahltaille')==1
448     if number<=0
449         set(hObject,'String','1e-4')
450         number=1e-4;
451     end
452     rechnez0(hObject, eventdata, handles)
453 end
454 if number~=str2double(get(hObject,'String'))
455     set(hObject,'String',num2str(number))
456 end
457
458 guidata(hObject, handles);
459
460
461
462 function eps_CreateFcn(hObject, eventdata, handles)
463 if ispc && isequal(get(hObject,'BackgroundColor'), ...
464         get(0,'defaultUicontrolBackgroundColor'))
465     set(hObject,'BackgroundColor','white');
466 end
467
468
469
```

```
function intens_Callback(hObject, eventdata, handles)
number=str2num(get(hObject,'String'));
if (isempty(number))
     set(hObject,'String','5')
elseif number==0
    set(hObject,'String','5')
elseif number~=str2double(get(hObject,'String'))
    set(hObject,'String',num2str(number))
end
guidata(hObject, handles);


function intens_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
        get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end









function beginn_ButtonDownFcn(hObject, eventdata, handles)



function startct_Callback(hObject, eventdata, handles)
number=str2num(get(hObject,'String'));
if (isempty(number))
    set(hObject,'String','2')

elseif number<=0
        set(hObject,'String','2')

elseif number~=str2double(get(hObject,'String'))
    set(hObject,'String',num2str(number))
end

guidata(hObject, handles);


function startct_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),...
        get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function startx_Callback(hObject, eventdata, handles)
number=str2num(get(hObject,'String'));
```

```
521 if (isempty(number))
522       set(hObject,'String','0')
523 elseif number==0
524     set(hObject,'String','0')
525 elseif number~=str2double(get(hObject,'String'))
526     set(hObject,'String',num2str(number))
527 end
528 guidata(hObject, handles);
529
530
531 function startx_CreateFcn(hObject, eventdata, handles)
532 if ispc && isequal(get(hObject,'BackgroundColor'),...
533         get(0,'defaultUicontrolBackgroundColor'))
534     set(hObject,'BackgroundColor','white');
535 end
536
537
538
539 function starty_Callback(hObject, eventdata, handles)
540 number=str2num(get(hObject,'String'));
541 if (isempty(number))
542       set(hObject,'String','0')
543 elseif number==0
544     set(hObject,'String','0')
545 elseif number~=str2double(get(hObject,'String'))
546     set(hObject,'String',num2str(number))
547 end
548 guidata(hObject, handles);
549
550
551 function starty_CreateFcn(hObject, eventdata, handles)
552 if ispc && isequal(get(hObject,'BackgroundColor'), ...
553         get(0,'defaultUicontrolBackgroundColor'))
554     set(hObject,'BackgroundColor','white');
555 end
556
557
558
559 function startz_Callback(hObject, eventdata, handles)
560 number=str2num(get(hObject,'String'));
561 if (isempty(number))
562       set(hObject,'String','0')
563 elseif number==0
564     set(hObject,'String','0')
565 elseif number~=str2double(get(hObject,'String'))
566     set(hObject,'String',num2str(number))
567 end
568 guidata(hObject, handles);
569
570
571 function startz_CreateFcn(hObject, eventdata, handles)
```

```
572 if ispc && isequal(get(hObject,'BackgroundColor'), ...
573         get(0,'defaultUicontrolBackgroundColor'))
574     set(hObject,'BackgroundColor','white');
575 end
576
577
578
579 function startE_Callback(hObject, eventdata, handles)
580 number=str2num(get(hObject,'String'));
581 if (isempty(number))
582      set(hObject,'String','1')
583 elseif number<=1
584     set(hObject,'String','1')
585 elseif number~=str2double(get(hObject,'String'))
586     set(hObject,'String',num2str(number))
587 end
588 guidata(hObject, handles);
589
590
591 function startE_CreateFcn(hObject, eventdata, handles)
592 if ispc && isequal(get(hObject,'BackgroundColor'), ...
593         get(0,'defaultUicontrolBackgroundColor'))
594     set(hObject,'BackgroundColor','white');
595 end
596
597
598
599 function startvx_Callback(hObject, eventdata, handles)
600 number=str2num(get(hObject,'String'));
601 if (isempty(number))
602      set(hObject,'String','0')
603 elseif number==0
604     set(hObject,'String','0')
605 elseif number~=str2double(get(hObject,'String'))
606     set(hObject,'String',num2str(number))
607 end
608 guidata(hObject, handles);
609
610
611 function startvx_CreateFcn(hObject, eventdata, handles)
612 if ispc && isequal(get(hObject,'BackgroundColor'), ...
613         get(0,'defaultUicontrolBackgroundColor'))
614     set(hObject,'BackgroundColor','white');
615 end
616
617
618
619 function startvy_Callback(hObject, eventdata, handles)
620 number=str2num(get(hObject,'String'));
621 if (isempty(number))
622      set(hObject,'String','0')
```

```
623 elseif number==0
624     set(hObject,'String','0')
625 elseif number~=str2double(get(hObject,'String'))
626     set(hObject,'String',num2str(number))
627 end
628 guidata(hObject, handles);
629
630
631 function startvy_CreateFcn(hObject, eventdata, handles)
632 if ispc && isequal(get(hObject,'BackgroundColor'), ...
633         get(0,'defaultUicontrolBackgroundColor'))
634     set(hObject,'BackgroundColor','white');
635 end
636
637
638
639 function startvz_Callback(hObject, eventdata, handles)
640 number=str2num(get(hObject,'String'));
641 if (isempty(number))
642       set(hObject,'String','0')
643 elseif number==0
644     set(hObject,'String','0')
645 elseif number~=str2double(get(hObject,'String'))
646     set(hObject,'String',num2str(number))
647 end
648 guidata(hObject, handles);
649
650
651 function startvz_CreateFcn(hObject, eventdata, handles)
652 if ispc && isequal(get(hObject,'BackgroundColor'), ...
653         get(0,'defaultUicontrolBackgroundColor'))
654     set(hObject,'BackgroundColor','white');
655 end
656
657
658
659 function Endzeit_Callback(hObject, eventdata, handles)
660 number=str2num(get(hObject,'String'));
661 if (isempty(number))
662       set(hObject,'String','0')
663 elseif number<=0
664     set(hObject,'String','1')
665 elseif number~=str2double(get(hObject,'String'))
666     set(hObject,'String',num2str(number))
667 end
668 guidata(hObject, handles);
669
670
671 function Endzeit_CreateFcn(hObject, eventdata, handles)
672 if ispc && isequal(get(hObject,'BackgroundColor'), ...
673         get(0,'defaultUicontrolBackgroundColor'))
```

```
674     set(hObject,'BackgroundColor','white');
675 end
676
677
678
679
680
681
682
683
684 function gauss_Callback(hObject, eventdata, handles)
685 if get(hObject,'Value')==0
686     set(handles.eps,'Visible','off');
687     set(handles.text5,'Visible','off');
688     set(handles.unterart,'Visible','off');
689     set(handles.Unter2,'Visible','off');
690
691 else
692     set(handles.text58,'Visible','off')
693     set(handles.eps,'Visible','on')
694     set(handles.startE,'String','3.39');
695     set(handles.text5,'Visible','on','String','Strahltaille');
696     set(handles.eps,'Visible','on','String','1e-4');
697     set(handles.text16,'Visible','on','String','Zielpunkt auf z Achse',...
698         'Position',[1.333,28.5,23.5,1.214]);
699     set(handles.startvx,'Visible','on','String','2');
700     set(handles.text17,'Visible','on','String','Phase');
701     set(handles.startvy,'Visible','on');
702     set(handles.text18,'Visible','off');
703     set(handles.startvz,'Visible','off');
704     set(handles.text8,'Visible','on','String','Gaußlaenge');
705     set(handles.startct,'Visible','on','Enable','off');
706     rechnez0(hObject, eventdata, handles);
707     set(handles.unterart,'Visible','off');
708     set(handles.startx,'String','30');
709     set(handles.starty,'String','0');
710     set(handles.startz,'String','-0.5');
711     set(handles.startvy,'String','0');
712     set(handles.Endzeit,'String','100');
713     set(handles.intens,'String','10');
714     set(handles.text22,'Visible','on');
715     set(handles.text21,'Visible','on');
716     set(handles.Endzeit,'String','100');
717 end
718
719
720 function bessel_Callback(hObject, eventdata, handles)
721 set(handles.startct,'Visible','off');
722 set(handles.text8,'Visible','off');
723 if get(hObject,'Value')==0
724     set(handles.eps,'Visible','off');
```

```
725     set(handles.text5,'Visible','off');
726     set(handles.unterart,'Visible','off');
727     set(handles.text58,'Visible','off')
728 else
729     set(handles.unterart,'Visible','on');
730     set(handles.eps,'Visible','on')
731     set(handles.text5,'Visible','on','String','Drehrichtung');
732     set(handles.eps,'Visible','on','String','1');
733     set(handles.Unter1,'Visible','on','String','M=2 paraxial','Value',1);
734     set(handles.Unter2,'Visible','on','String','M=0 paraxial');
735     set(handles.startE,'String','1');
736     set(handles.intens,'String','1e7');
737     set(handles.startx,'String','0.1');
738     set(handles.starty,'String','0.1');
739     set(handles.startz,'String','0');
740     set(handles.text16,'Visible','on','String','Startrichtung in x',...
741         'Position',[5.333,28.5,19.5,1.214]);
742     set(handles.startvx,'Visible','on','String','0');
743     set(handles.text17,'Visible','on','String','Startrichtung in y');
744     set(handles.startvy,'Visible','on','String','0');
745     set(handles.text18,'Visible','on');
746     set(handles.startvz,'Visible','on','String','0');
747     set(handles.text58,'Visible','on');
748     set(handles.text22,'Visible','on');
749     set(handles.text21,'Visible','on');
750     set(handles.Endzeit,'String','100');
751 end
752
753 function eben_Callback(hObject, eventdata, handles)
754 set(handles.startct,'Visible','off','Enable','on');
755 set(handles.text8,'Visible','off');
756 if get(hObject,'Value')==0
757     set(handles.eps,'Visible','off');
758     set(handles.text5,'Visible','off');
759     set(handles.unterart,'Visible','off');
760
761 else
762     set(handles.text58,'Visible','off')
763     set(handles.unterart,'Visible','on');
764     set(handles.eps,'Visible','on','String','0')
765     set(handles.text5,'Visible','on','String','Polarisation');
766     set(handles.Unter1,'Visible','on','String',...
767         'Elliptische Polarisation','Value',1);
768     set(handles.Unter2,'Visible','on','String','Gedämpfte Welle');
769     set(handles.intens,'String','1');
770     set(handles.startx,'String','0');
771     set(handles.starty,'String','0');
772     set(handles.startz,'String','0');
773     set(handles.text16,'Visible','on','String','Startrichtung in x'...
774         ,'Position',[5.333,28.5,19.5,1.214]);
775     set(handles.startvx,'Visible','on','String','0');
```

```
776     set(handles.text17,'Visible','on','String','Startrichtung in y');
777     set(handles.startvy,'Visible','on','String','0');
778     set(handles.text18,'Visible','on');
779     set(handles.startvz,'Visible','on','String','0');
780     set(handles.startE,'String','1');
781     set(handles.text22,'Visible','off');
782     set(handles.text21,'Visible','off');
783     set(handles.Endzeit,'String','2');
784 end
785
786
787
788 function Unter2_Callback(hObject, eventdata, handles)
789 if strcmp(get(hObject,'String'),'Gedämpfte Welle')
790 if get(hObject,'Value')==0
791
792     set(handles.text8,'Visible','off');
793     set(handles.startct,'Visible','off');
794 else
795
796     set(handles.text8,'Visible','on','String','Dämpfung');
797     set(handles.startct,'Visible','on','String','2');
798
799 end
800 end
801
802
803
804 function Unter1_Callback(hObject, eventdata, handles)
805
806 set(handles.startct,'Visible','off');
807 set(handles.text8,'Visible','off');
808
809
810
811 function ord1_Callback(hObject, eventdata, handles)
812
813
814
815 function ord1_CreateFcn(hObject, eventdata, handles)
816 if ispc && isequal(get(hObject,'BackgroundColor'),...
817         get(0,'defaultUicontrolBackgroundColor'))
818     set(hObject,'BackgroundColor','white');
819 end
820
821
822
823 function ordmin1_Callback(hObject, eventdata, handles)
824 number=str2num(get(hObject,'String'));
825 if (isempty(number))
826      set(hObject,'String','0')
```

```
827 elseif number==0
828     set(hObject,'String','0')
829 elseif number~=str2double(get(hObject,'String'))
830     set(hObject,'String',num2str(number))
831 end
832 guidata(hObject, handles);
833
834
835 function ordmin1_CreateFcn(hObject, eventdata, handles)
836 if ispc && isequal(get(hObject,'BackgroundColor'),...
837         get(0,'defaultUicontrolBackgroundColor'))
838     set(hObject,'BackgroundColor','white');
839 end
840
841
842
843 function abzmin1_Callback(hObject, eventdata, handles)
844 number=str2num(get(hObject,'String'));
845 if (isempty(number))
846      set(hObject,'String','0')
847 elseif number==0
848     set(hObject,'String','0')
849 elseif number~=str2double(get(hObject,'String'))
850     set(hObject,'String',num2str(number))
851 end
852 guidata(hObject, handles);
853
854
855 function abzmin1_CreateFcn(hObject, eventdata, handles)
856 if ispc && isequal(get(hObject,'BackgroundColor'),...
857         get(0,'defaultUicontrolBackgroundColor'))
858     set(hObject,'BackgroundColor','white');
859 end
860
861
862
863 function ordmax1_Callback(hObject, eventdata, handles)
864 number=str2num(get(hObject,'String'));
865 if (isempty(number))
866      set(hObject,'String','0')
867 elseif number==0
868     set(hObject,'String','0')
869 elseif number~=str2double(get(hObject,'String'))
870     set(hObject,'String',num2str(number))
871 end
872 guidata(hObject, handles);
873
874
875 function ordmax1_CreateFcn(hObject, eventdata, handles)
876 if ispc && isequal(get(hObject,'BackgroundColor'),...
877         get(0,'defaultUicontrolBackgroundColor'))
```

```
878     set(hObject,'BackgroundColor','white');
879 end
880
881
882
883 function abzmax1_Callback(hObject, eventdata, handles)
884 number=str2num(get(hObject,'String'));
885 if (isempty(number))
886       set(hObject,'String','0')
887 elseif number==0
888     set(hObject,'String','0')
889 elseif number~=str2double(get(hObject,'String'))
890     set(hObject,'String',num2str(number))
891 end
892 guidata(hObject, handles);
893
894
895 function abzmax1_CreateFcn(hObject, eventdata, handles)
896 if ispc && isequal(get(hObject,'BackgroundColor'),...
897         get(0,'defaultUicontrolBackgroundColor'))
898     set(hObject,'BackgroundColor','white');
899 end
900
901
902
903
904
905
906 function ord2_Callback(hObject, eventdata, handles)
907
908
909
910 function ord2_CreateFcn(hObject, eventdata, handles)
911 if ispc && isequal(get(hObject,'BackgroundColor'),...
912         get(0,'defaultUicontrolBackgroundColor'))
913     set(hObject,'BackgroundColor','white');
914 end
915
916
917
918 function abz1_Callback(hObject, eventdata, handles)
919
920
921
922 function abz1_CreateFcn(hObject, eventdata, handles)
923 if ispc && isequal(get(hObject,'BackgroundColor'),...
924         get(0,'defaultUicontrolBackgroundColor'))
925     set(hObject,'BackgroundColor','white');
926 end
927
928
```

```
929
930 function abz2_Callback(hObject, eventdata, handles)
931
932
933
934
935 function abz2_CreateFcn(hObject, eventdata, handles)
936 if ispc && isequal(get(hObject,'BackgroundColor'),...
937         get(0,'defaultUicontrolBackgroundColor'))
938     set(hObject,'BackgroundColor','white');
939 end
940
941
942
943 function app1_Callback(hObject, eventdata, handles)
944
945
946
947 function app1_CreateFcn(hObject, eventdata, handles)
948 if ispc && isequal(get(hObject,'BackgroundColor'),...
949         get(0,'defaultUicontrolBackgroundColor'))
950     set(hObject,'BackgroundColor','white');
951 end
952
953
954 function app2_Callback(hObject, eventdata, handles)
955
956
957
958 function app2_CreateFcn(hObject, eventdata, handles)
959 if ispc && isequal(get(hObject,'BackgroundColor'), ...
960         get(0,'defaultUicontrolBackgroundColor'))
961     set(hObject,'BackgroundColor','white');
962 end
963
964
965
966 function appmin1_Callback(hObject, eventdata, handles)
967 number=str2num(get(hObject,'String'));
968 if (isempty(number))
969       set(hObject,'String','0')
970 elseif number==0
971     set(hObject,'String','0')
972 elseif number~=str2double(get(hObject,'String'))
973     set(hObject,'String',num2str(number))
974 end
975 guidata(hObject, handles);
976
977
978 function appmin1_CreateFcn(hObject, eventdata, handles)
979 if ispc && isequal(get(hObject,'BackgroundColor'), ...
```

```
980           get(0,'defaultUicontrolBackgroundColor'))
981       set(hObject,'BackgroundColor','white');
982   end
983
984
985
986   function appmax1_Callback(hObject, eventdata, handles)
987   number=str2num(get(hObject,'String'));
988   if (isempty(number))
989         set(hObject,'String','0')
990   elseif number==0
991       set(hObject,'String','0')
992   elseif number~=str2double(get(hObject,'String'))
993       set(hObject,'String',num2str(number))
994   end
995   guidata(hObject, handles);
996
997
998   function appmax1_CreateFcn(hObject, eventdata, handles)
999   if ispc && isequal(get(hObject,'BackgroundColor'), ...
1000          get(0,'defaultUicontrolBackgroundColor'))
1001      set(hObject,'BackgroundColor','white');
1002  end
1003
1004
1005
1006  function ordmin2_Callback(hObject, eventdata, handles)
1007  number=str2num(get(hObject,'String'));
1008  if (isempty(number))
1009        set(hObject,'String','0')
1010  elseif number==0
1011      set(hObject,'String','0')
1012  elseif number~=str2double(get(hObject,'String'))
1013      set(hObject,'String',num2str(number))
1014  end
1015  guidata(hObject, handles);
1016
1017
1018  function ordmin2_CreateFcn(hObject, eventdata, handles)
1019  if ispc && isequal(get(hObject,'BackgroundColor'), ...
1020         get(0,'defaultUicontrolBackgroundColor'))
1021     set(hObject,'BackgroundColor','white');
1022  end
1023
1024
1025
1026  function abzmin2_Callback(hObject, eventdata, handles)
1027  number=str2num(get(hObject,'String'));
1028  if (isempty(number))
1029        set(hObject,'String','0')
1030  elseif number==0
```

```
1031      set(hObject,'String','0')
1032  elseif number~=str2double(get(hObject,'String'))
1033      set(hObject,'String',num2str(number))
1034  end
1035  guidata(hObject, handles);
1036
1037  function abzmin2_CreateFcn(hObject, eventdata, handles)
1038  if ispc && isequal(get(hObject,'BackgroundColor'), ...
1039          get(0,'defaultUicontrolBackgroundColor'))
1040      set(hObject,'BackgroundColor','white');
1041  end
1042
1043
1044
1045  function ordmax2_Callback(hObject, eventdata, handles)
1046  number=str2num(get(hObject,'String'));
1047  if (isempty(number))
1048       set(hObject,'String','0')
1049  elseif number==0
1050      set(hObject,'String','0')
1051  elseif number~=str2double(get(hObject,'String'))
1052      set(hObject,'String',num2str(number))
1053  end
1054  guidata(hObject, handles);
1055
1056
1057  function ordmax2_CreateFcn(hObject, eventdata, handles)
1058  if ispc && isequal(get(hObject,'BackgroundColor'), ...
1059          get(0,'defaultUicontrolBackgroundColor'))
1060      set(hObject,'BackgroundColor','white');
1061  end
1062
1063
1064
1065  function abzmax2_Callback(hObject, eventdata, handles)
1066  number=str2num(get(hObject,'String'));
1067  if (isempty(number))
1068       set(hObject,'String','0')
1069  elseif number==0
1070      set(hObject,'String','0')
1071  elseif number~=str2double(get(hObject,'String'))
1072      set(hObject,'String',num2str(number))
1073  end
1074  guidata(hObject, handles);
1075
1076
1077  function abzmax2_CreateFcn(hObject, eventdata, handles)
1078  if ispc && isequal(get(hObject,'BackgroundColor'), ...
1079          get(0,'defaultUicontrolBackgroundColor'))
1080      set(hObject,'BackgroundColor','white');
1081  end
```

```
1082
1083
1084
1085
1086 function appmin2_Callback(hObject, eventdata, handles)
1087 number=str2num(get(hObject,'String'));
1088 if (isempty(number))
1089       set(hObject,'String','0')
1090 elseif number==0
1091     set(hObject,'String','0')
1092 elseif number~=str2double(get(hObject,'String'))
1093     set(hObject,'String',num2str(number))
1094 end
1095 guidata(hObject, handles);
1096
1097
1098 function appmin2_CreateFcn(hObject, eventdata, handles)
1099 if ispc && isequal(get(hObject,'BackgroundColor'), ...
1100         get(0,'defaultUicontrolBackgroundColor'))
1101     set(hObject,'BackgroundColor','white');
1102 end
1103
1104
1105
1106 function appmax2_Callback(hObject, eventdata, handles)
1107 number=str2num(get(hObject,'String'));
1108 if (isempty(number))
1109       set(hObject,'String','0')
1110 elseif number==0
1111     set(hObject,'String','0')
1112 elseif number~=str2double(get(hObject,'String'))
1113     set(hObject,'String',num2str(number))
1114 end
1115 guidata(hObject, handles);
1116
1117
1118 function appmax2_CreateFcn(hObject, eventdata, handles)
1119 if ispc && isequal(get(hObject,'BackgroundColor'),...
1120         get(0,'defaultUicontrolBackgroundColor'))
1121     set(hObject,'BackgroundColor','white');
1122 end
1123
1124
1125
1126 function dia1_CreateFcn(hObject, eventdata, handles)
1127
1128
1129 function dia1_ButtonDownFcn(hObject, eventdata, handles)
1130
1131
1132 function dia1_DeleteFcn(hObject, eventdata, handles)
```

```
1133
1134
1135 function dia2_CreateFcn(hObject, eventdata, handles)
1136
1137
1138 % --- Executes on button press in auto1.
1139 function auto1_Callback(hObject, eventdata, handles)
1140 % hObject    handle to auto1 (see GCBO)
1141 % eventdata  reserved - to be defined in a future version of MATLAB
1142 % handles    structure with handles and user data (see GUIDATA)
1143
1144 % Hint: get(hObject,'Value') returns toggle state of auto1
1145
1146
1147 % --- Executes on button press in auto2.
1148 function auto2_Callback(hObject, eventdata, handles)
1149 % hObject    handle to auto2 (see GCBO)
1150 % eventdata  reserved - to be defined in a future version of MATLAB
1151 % handles    structure with handles and user data (see GUIDATA)
1152
1153 % Hint: get(hObject,'Value') returns toggle state of auto2
```

Listing 2: berechnet F für Bessel M0 Fall

```
1 function F=berechneFBesselM0(ort,epsilon,k,E_0)
2 E=E_0*exp(1i*epsilon*(ort(1)-ort(4))*k);
3 S_x=-E*k/2*(ort(2)-1i*ort(3));
4 S_y=1i*S_x;
5 S_z=E*1i*epsilon;
6 F=[ 0 real(S_x) real(S_y) real(S_z);
7      real(S_x) 0 imag(S_z) -imag(S_y);
8      real(S_y) -imag(S_z) 0 Imag(S_x);
9      real(S_z) imag(S_y) -imag(S_x) 0];
```

Listing 3: berechnet Trajektorie im Bessel M0 Fall

```
1 function [tau weg]=initbessel0(start,nq,lambda,nm,epsilon,E_0,tend)
2 global c
3
4 k=2*pi()/lambda;
5 omega=k*c;
6 beginn=[start(2)*c/omega start(3)*c/omega start(4)*c/omega start(6)*c ...
7      start(7)*c];
8 D=nq*E_0/nm;
9 C=-(start(5)-start(8))*omega/D-1;
10 Omega=omega*(start(5)-start(8));
11 tend=tend*2*pi()/Omega;
12 options=odeset('InitialStep',1/10/omega,'MaxStep',1/4/omega);
13 [tau,rundu]=ode45(@(tau,rundu) bewegungbessel0(tau,rundu,C,D,k,...
14      omega,c,epsilon),[0 tend],beginn,options);
15 tau=tau/2/pi()*Omega;
16 eta=etaber(tau,C,D);
17 etapunkt=-D*(cos(eta)+C);
```

```
weg(:,6:7)=rundu(:,4:5)/c;
weg(:,5)=omega/2./etapunkt.*(weg(:,6).^2+weg(:,7).^2+etapunkt.^2/omega^2+1);
weg(:,8)=etapunkt/omega-weg(:,5);
weg(:,2:4)=rundu(:,1:3)*omega/c;
end

function erg=etaber(tau,C,D)
if abs(C)>1
    erg=-2*atan(sqrt((C+1)/(C-1))*tan(D/2*sqrt(C^2-1)*tau));
    erg=erg+floor(((D/2*sqrt(C^2-1)*tau)+pi()/2)/pi())*pi();
else
    erg=-2*atan(sqrt((C+1)/(1-C))*tanh(D/2*sqrt(-C^2+1)*tau));
end
end

function drundu=bewegungbessel0(tau,rundu,C,D,k,omega,c,epsilon)
eta=etaber(tau,C,D);
etapunkt=-D*(cos(eta)+C);
drundu=zeros(5,1);
drundu(1:2)=rundu(4:5);
drundu(3)=(rundu(4)^2+rundu(5)^2)/c*omega/2/etapunkt-...
    (etapunkt/omega-omega/etapunkt)/2;
drundu(4)=c*D*(-etapunkt/omega*k/2*(cos(eta)*rundu(1)+sin(eta)*rundu(2)))...
    +rundu(4)/c*cos(eta)*epsilon);
drundu(5)=c*D*(etapunkt/omega*k/2*(sin(eta)*rundu(1)-cos(eta)*rundu(2)))...
    -rundu(3)/c*cos(eta)*epsilon);
end
```

Listing 4: berechnet Trajektorie im Bessel M2 Fall

```
function [tau weg]=Besselrechnen2(nq,nm,epsilon,lambda,E_0,start,tend)

global c
c=2.99792458e10;
q0=1.702691e-9;
m=9.1093826e-28;
omega=2*pi()*c/lambda;
Omega=omega*(start(5)-start(8))*epsilon;
C=E_0*nq*Omega*epsilon/(2*nm);
omegam=sqrt(Omega^2/4-C);
omegap=sqrt(Omega^2/4+C);
A=[Omega 0 -2*omegam 0;0 -2*omegap 0 Omega; 0 0 0 2*C; 2*C 0 0 0]/Omega;
B=inv(A)/Omega;
Koeff=zeros(4,1);
Koeff(1:4)=B*[start(2)/Omega*c ; start(3)/Omega*c; start(6)*c; start(7)*c];
tend=2*pi*tend/abs(Omega);
schritt=min(tend/1000,1/(Omega/2+omegap)/30);
if schritt<tend/1e7
    schritt=tend/1e7;
end
tau=(0:schritt:tend);
r=exp(-1i*Omega*tau/2).*((Omega*Koeff(1)-2*omegap*1i*Koeff(2))*...
```

```
      cos(omegap*tau)+(Omega*Koeff(2)+2*1i*omegap*Koeff(1))*...
      sin(omegap*tau)+(1i*Omega*Koeff(4)-2*omegam*Koeff(3))*...
      cos(omegam*tau)+(-1i*Omega*Koeff(3)-2*omegam*Koeff(4))*sin(omegam*tau));
rpunkt=(omegap*cos(omegap*tau)*(Koeff(2)*Omega + 2*Koeff(1)*1i*omegap)...
      - omegam*cos(omegam*tau)*(2*Koeff(4)*omegam + Koeff(3)*Omega*1i)...
      - omegap*sin(omegap*tau)*(Koeff(1)*Omega - 2*Koeff(2)*1i*omegap)...
      + omegam*sin(omegam*tau)*(2*Koeff(3)*omegam - Koeff(4)*Omega*1i))...
      ./exp((Omega*1i*tau)/2) - (Omega*1i*(cos(omegap*tau)*(Koeff(1)*Omega...
      - 2*Koeff(2)*1i*omegap) - cos(omegam*tau)*(2*Koeff(3)*omegam...
      - Koeff(4)*Omega*1i) + sin(omegap*tau)*(Koeff(2)*Omega +...
      2*Koeff(1)*1i*omegap) - sin(omegam*tau)*(2*Koeff(4)*omegam ...
      + Koeff(3)*Omega*1i)))./(2*exp((Omega*1i*tau)/2));
rpunkt=rpunkt/c;
hold off

weg(:,2)=real(r);
weg(:,3)=imag(r);
weg(:,6)=real(rpunkt);
weg(:,7)=imag(rpunkt);
weg(:,8)=(1-(Omega/omega)^2+weg(:,6).^2+weg(:,7).^2)*omega/(2*Omega);
weg(:,5)=Omega/omega+weg(:,8);
k=4*c/Omega^2;
kp=sqrt(1+k);
km=sqrt(1-k);
weg(:,4)=k^2*omega*(Koeff(4)^2-Koeff(3)^2*sin(2*omegam*tau))/(16*c*km)+...
      k^2*omega*(Koeff(4)^2-Koeff(3)^2*sin(2*omegap*tau))/(16*c*kp)-...
      k^2*omega*(Koeff(3)*Koeff(4)*(1-cos(2*omegam*tau)))/(8*c*km)+...
      k^2*omega*(Koeff(1)*Koeff(2)*(1-cos(2*omegap*tau)))/(8*c*kp)+...
      c*tau/2*(-omega/Omega+Omega/omega-Omega/omega*k^2*omega^2/(8*c^2)*...
      (norm(Koeff(1:4)))^2);
weg(:,1)=weg(:,4)+Omega/omega*tau(:);
tau=tau/2/pi*abs(Omega);
weg(:,1:4)=weg(:,1:4)*Omega/c;
```

Listing 5: berechnet Trajektorie im Gauß Fall

```
function [zeit weg flag ye1 ye2 ye3 Laengeneu]=initgauss...
      (nx,nz,ny,tend,gammaa,lambda,w0,LasE,psi_0,s)
% berechnet die Trajektorie eines Teilchens welches mit obigen Parametern
% startet
%nx ny Startort in x und y angegeben in w0
%nz Startort in z angegeben in z0
%s Zielort auf der z-Achse angegeben in z0
%gammaa Anfangsenergie
%lambda Wellenlaenge
%w0 Strahltaille
%LasE Laserintensitaet A0 angegeben in e/(mc^2)
%psi_o Startphase


global z0 w0intern c
w0intern=w0;
```

```
c=2.99792458e10;
omega=c*2*pi()/lambda;
k=omega/c;
z0=k*w0^2/2;

ye1=0;
ye2=0;
ye3=0;
ct_0=0;
x_0=w0*nx;
y_0=w0*ny;
z_0=z0*nz;
s=s*z0;
u00=gammaa*c;
betrag3u=(gammaa^2-1)^(1/2)*c;
u01=-betrag3u*x_0/(x_0^2+(z_0-s)^2+y_0^2)^(1/2);
u02=-betrag3u*y_0/(x_0^2+(z_0-s)^2+y_0^2)^(1/2);
u03=-betrag3u*(z_0-s)/(x_0^2+(z_0-s)^2+y_0^2)^(1/2);

tend2=tend*2*pi()/omega;


ye=[ct_0 x_0 y_0 z_0 u00 u01 u02 u03];
te=0;
Laengealt=0;
zeit=te;
flag=0;
max=w0/u00/2;
while flag==0
optionsstart=odeset('InitialStep',1/omega/1000,'MaxStep',max,...
    'Events',@Eintritt,'RelTol',1e-7);
[tau,rundu,te,ye,ie]=ode45(@(tau,rundu) bewegunggauss(tau,rundu,...
    lambda,w0,LasE,psi_0),[te(end) tend2],(ye(end,:)),optionsstart);

Laengeneu=size(tau,1)+Laengealt;
zeit(Laengealt+1:Laengeneu)=tau/2/pi()*omega;
weg(Laengealt+1:Laengeneu,1:4)=rundu(:,1:4);
weg(Laengealt+1:Laengeneu,5:8)=rundu(:,5:8)/c;
Laengealt=Laengeneu;

if tau(end)==tend2

    return
end

clear tau rundu

if ie==2
    Laengealt=Laengealt-3;
    ye(end,1:4)=weg(Laengealt,1:4);
    ye(end,5:8)=weg(Laengealt,5:8)*c;
```

```
68        ye(end,6:8)=ye(end,6:8)/norm(ye(end,6:8))*sqrt(ye(end,5)^2-c^2);
69        weg(Laengealt,1:4)=ye(end,1:4);
70        weg(Laengealt,5:8)=ye(end,5:8)/c;
71        te(end)=zeit(Laengealt)*2*pi()/omega;
72
73    else
74        flag=1;
75
76    end
77    if flag==1
78
79        break
80    end
81
82    end
83    ye1=te/2/pi()*omega;
84    flag=0;
85
86    while flag==0
87    clear tau rundu
88    optionsmitte=odeset('InitialStep',1/omega,'MaxStep',max...
89        ,'Events',@Eintritt,'RelTol',1e-8);
90    [tau,rundu,te,ye,ie]=ode45(@(tau,rundu) bewegunggauss(tau,rundu...
91        ,lambda,w0,LasE,psi_0),[te(end) tend2],ye(end,:),optionsmitte);
92
93    Laengeneu=size(tau,1)+Laengealt;
94    zeit(Laengealt+1:Laengeneu)=tau/2/pi()*omega;
95    weg(Laengealt+1:Laengeneu,1:4)=rundu(:,1:4);
96    weg(Laengealt+1:Laengeneu,5:8)=rundu(:,5:8)/c;
97    Laengealt=Laengeneu;
98    if tau(end)==tend2
99
100       return
101   end
102
103
104
105
106   if ie==2
107         Laengealt=Laengealt-3;
108       ye(end,1:4)=weg(Laengealt,1:4);
109       ye(end,5:8)=weg(Laengealt,5:8)*c;
110       ye(end,6:8)=ye(end,6:8)/norm(ye(end,6:8))*sqrt(ye(end,5)^2-c^2);
111       weg(Laengealt,1:4)=ye(end,1:4);
112       weg(Laengealt,5:8)=ye(end,5:8)/c;
113       te(end)=zeit(Laengealt)*2*pi()/omega;
114   else
115       flag=1;
116
117
118   end
```

```
119
120 if flag==1
121
122     break
123
124 end
125
126 end
127 ye2=te/2/pi()*omega;
128 flag=0;
129 while flag==0
130 clear tau rundu
131 optionsende=odeset('InitialStep',1/omega','Events',@Ende...
132     ,'MaxStep',max,'RelTol',1e-7);
133 [tau,rundu,te,ye,ie]=ode45(@(tau,rundu) bewegunggauss(tau,rundu...
134     ,lambda,w0,LasE,psi_0),[te(end) tend2],ye(end,:),optionsende);
135
136 Laengeneu=size(tau,1)+Laengealt;
137 zeit(Laengealt+1:Laengeneu)=tau/2/pi()*omega;
138 weg(Laengealt+1:Laengeneu,1:4)=rundu(:,1:4);
139 weg(Laengealt+1:Laengeneu,5:8)=rundu(:,5:8)/c;
140 Laengealt=Laengeneu;
141 if tau(end)==tend2
142
143     return
144 end
145
146 if ie==2
147     Laengealt=Laengealt-3;
148     ye(end,1:4)=weg(Laengealt,1:4);
149     ye(end,5:8)=weg(Laengealt,5:8)*c;
150     ye(end,6:8)=ye(end,6:8)/norm(ye(end,6:8))*sqrt(ye(end,5)^2-c^2);
151     weg(Laengealt,1:4)=ye(end,1:4);
152     weg(Laengealt,5:8)=ye(end,5:8)/c;
153     te(end)=zeit(Laengealt)*2*pi()/omega;
154 else
155     flag=1;
156
157
158 end
159
160 if flag==1
161     break
162 end
163 end
164 ye3=te/2/pi()*omega;
165 end
166
167  function [value,isterminal,direction] = Eintritt(tau,rundu)
168  global w0intern z0 c
169  value=[rundu(2)^2+rundu(3)^2-4*w0intern^2*(1+rundu(4)^2/z0^2)...
```

```
170         ;abs(rundu(5)^2-(norm(rundu(6:8)))^2-c^2)/c^2-0.0001];
171   isterminal=[1;1];
172   direction=[0;0];
173   end
174
175   function [value,isterminal,direction] = Ende(tau,rundu)
176   global w0intern z0 c
177   value=[rundu(2)^2+rundu(3)^2-100*w0intern^2*(1+rundu(4)^2/z0^2)...
178         ;abs(rundu(5)^2-(norm(rundu(6:8)))^2-c^2)/c^2-0.0001];
179   isterminal=[1;1];
180   direction=[0;0];
181   end
```

Listing 6: berechnet Integrationsschritt für Gauß Fall

```
1  function drundu=bewegunggauss(tau,rundu,lambda,w0,LasE,psi_0)
2  %u=rundu(5:8) entspricht u*c
3  %F entspricht F*q/(m*c)
4  A=berechneFgauss(lambda,w0,LasE,psi_0,rundu(1:4));
5  drundu=zeros(8,1);
6  drundu(1:4)=rundu(5:8);
7  drundu(5:8)=A*rundu(5:8);
```

Listing 7: berechnet F für Gauß Fall

```
1  function F=berechneFgauss(lambda,w0,LasE,psi_0,ort)
2  c=2.99792458e10;
3  k=2*pi()/lambda;
4  z0=k*w0^2/2;
5
6  xi=ort(2)/w0;
7  eta=ort(3)/w0;
8  zetaa=ort(4)/z0;
9  rho=sqrt(xi^2+eta^2);
10 r=rho*w0;
11 w=w0*sqrt(1+zetaa^2);
12 w0dw=w0/w;
13 R=ort(4)+z0^2/ort(4);
14 psiR=k*r^2/(2*R);
15 psiG=atan(zetaa);
16 psiP=k*(ort(1)-ort(4));
17 epsilon=w0/z0;
18 psi=psi_0+psiP-psiR+psiG;
19 S0=sin(psi);
20 %S1=(w0dw)^1*sin(psi+1*psiG);
21 S2=(w0dw)^2*sin(psi+2*psiG);
22 S3=(w0dw)^3*sin(psi+3*psiG);
23 S4=(w0dw)^4*sin(psi+4*psiG);
24 S5=(w0dw)^5*sin(psi+5*psiG);
25 S6=(w0dw)^6*sin(psi+6*psiG);
26 %C0=(w0dw)^0*cos(psi+0*psiG);
27 C1=(w0dw)*cos(psi+psiG);
28 C2=(w0dw)^2*cos(psi+2*psiG);
```

```
C3=(w0dw)^3*cos(psi+3*psiG);
C4=(w0dw)^4*cos(psi+4*psiG);
C5=(w0dw)^5*cos(psi+5*psiG);
C6=(w0dw)^6*cos(psi+6*psiG);
C7=(w0dw)^7*cos(psi+7*psiG);
%C8=(w0/w)^8*cos(psi+8*psiG);
E0=k*LasE*c;
E=E0*w0dw*exp(-r^2/w^2);
% (-r^2/w^2)
%Unsere Felder kompakt aufgeschrieben:
Ex=E*(S0+epsilon^2*(xi^2*S2-rho^4*S3/4)+epsilon^4*(S2/8-rho^2*S3/4-...
    rho^2*(rho^2-16*xi^2)*S4/16-rho^4*(rho^2+2*xi^2)*S5/8+rho^8*S6/32));
Ey=E*xi*eta*(epsilon^2*S2+epsilon^4*(rho^2*S4-rho^4*S5/4));
Ez=E*xi*(epsilon*C1+epsilon^3*(-C2/2+rho^2*C3-rho^4*C4/4)+epsilon^5*...
    (-3*C3/8-3*rho^2*C4/8+17*rho^4*C5/16-3*rho^6*C6/8+rho^8*C7/32));
Bx=0;
By=E*(S0+epsilon^2*(rho^2*S2/2-rho^4*S3/4)+epsilon^4*(-S2/8+rho^2*S3/4+...
    5*rho^4*S4/16-rho^6*S5/4+rho^8*S6/32));
Bz=E*eta*(epsilon*C1+epsilon^3*(C2/2+rho^2*C3/2-rho^4*C4/4)+epsilon^5*...
    (3*C3/8+3*rho^2*C4/8+3*rho^4*C5/16-rho^6*C6/4+rho^8*C7/32));
% Wir erzeugen uns den Feldtensor F schon modifiziert
%mit dem Ladungs-Masse Verhaeltnis
F=[ 0 Ex Ey Ez;
    Ex 0 Bz -By;
    Ey -Bz 0 Bx;
    Ez By -Bx 0];
```

Listing 8: berechnet Trajektorie im ebenen Wellen Fall

```
function [tauruh rruh]=initeben(start,tend,nq,nm,alpha0,lambda,flag,xi,delta)

c=2.99792458e10;
omega=2*pi*c/lambda;
k=omega/c*[1;0;0;1];
unull=zeros(4,1);
unull(1:4)=start(5:8)
Omega=c*(k(1)*unull(1)-k(4)*unull(4));
alpha0=alpha0*nq/nm;

x0=start(1)/omega*c;
x1=start(2)/omega*c;
x2=start(3)/omega*c;
x3=start(4)/omega*c;
t=0;
etanull=x0*k(1)-x3*k(4);
if flag==1
    anull=alpha0*[0;-sin(Omega*t+etanull)*delta;...
        cos(Omega*t+etanull)*sqrt(1-delta^2);0];
elseif flag==2
    anull=alpha0*exp(-(Omega*t)/xi^2)*[0;-sin(Omega*t+etanull)...
        *delta;cos(Omega*t+etanull)*sqrt(1-delta^2);0];
end
```

```
24
25 tend=2*pi()/Omega*tend;
26 options=odeset('Refine',1000);
27 [tauruh rruh]=ode45(@(tau,r) bewegeben(tau,r,unull,anull,k,Omega...
28       ,c,flag,alpha0,delta,xi),[0 tend],([x0; x1; x2; x3]),options)
29 rruh(1:100,:)
30 rruh(:,5:8)=urechne2(unull,anull,k,Omega,c,tauruh,flag,alpha0,delta,xi);
31 tauruh=tauruh/2/pi()*Omega;
32 rruh(:,1:4)=rruh(:,1:4)*omega/c;
```

Listing 9: berechnet Integrationsschritt für ebene Welle Fall

```
1 function ds=bewegeben(tau,s,unull,anull,k,Omega,c,flag,alpha0,delta,xi)
2
3 ds=urechne(unull,anull,k,Omega,c,tau,flag,alpha0,delta,xi)*c;
```

Listing 10: berechnet Matrix u für ebene Welle

```
1 function u=urechne2(unull,anull,k,Omega,c,t,flag,alpha0,delta,xi)
2
3
4       atau(:,3)=alpha0*cos(Omega*t)*sqrt(1-delta^2);
5       atau(:,2)=alpha0*(-sin(Omega*t)*delta);
6       atau(:,4)=0;
7
8 if flag==2
9       atau(:,2)=atau(:,2).*exp(-(Omega*t)/xi^2);
10      atau(:,3)=atau(:,3).*exp(-(Omega*t)/xi^2);
11 end
12 au=0;
13 aa=0;
14 for i=2:4
15      au=(atau(:,i)-anull(i))*unull(i)+au;
16 end
17 au=(atau(:,1)-anull(1))*unull(1)-au;
18 for i=2:4
19      aa=(atau(:,i)-anull(i)).^2+aa;
20 end
21 aa=(atau(:,1)-anull(1)).^2-aa;
22 ku=Omega/c;
23 u(:,1)=unull(1)+k(1)*au(:)/ku-k(1)*(aa/(2*ku));
24 u(:,4)=unull(4)+k(4)*au(:)/ku-k(4)*(aa/(2*ku));
25 u(:,2)=unull(2)-(atau(:,2)-anull(2));
26 u(:,3)=unull(3)-(atau(:,3)-anull(3));
```

Listing 11: berechnet Vektor u für ebene Welle

```
1 function u=urechne(unull,anull,k,Omega,c,t,flag,alpha0,delta,xi)
2
3 if flag==1
4       atau=alpha0*[0;-sin(Omega*t)*delta;cos(Omega*t)*sqrt(1-delta^2);0];
5 elseif flag==2
6       atau=alpha0*exp(-(Omega*t)/xi^2)*[0;sin(Omega*t)*delta...
```

```
          ;cos(Omega*t)*sqrt(1-delta^2);0];
end
au=0;
aa=0;
for i=2:4
    au=(atau(i)-anull(i))*unull(i)+au;
end
au=(atau(1)-anull(1))*unull(1)-au;
for i=2:4
    aa=(atau(i)-anull(i)).^2+aa;
end
aa=(atau(1)-anull(1))^2-aa;
ku=Omega/c;
u=unull-(atau-anull-k*au/ku)-k*(aa/(2*ku));
```

Listing 12: berechnet Gaußlänge

```
function rechnez0(hObject, eventdata, handles)
lambda=str2double(get(handles.lambda,'String'));
w0=str2double(get(handles.eps,'String'));
z0=pi()/lambda*w0^2;
set(handles.startct,'String',num2str(z0, '%10.1e\n'));
```

Listing 13: Programm für Plot

```
function zeichne(hObject,eventdata,handles,rundu,tau,flag)
if flag==1
        rundu(:,1)=tau;
        rundu(:,6)=(rundu(:,2).^2+rundu(:,3).^2).^(1/2);

        abz=(get(handles.abz1,'Value'));
        ord=(get(handles.ord1,'Value'));
        app=(get(handles.app1,'Value'));
        if get(handles.auto1,'Value')==0
        abzmax=str2double(get(handles.abzmax1,'String'));
        abzmin=str2double(get(handles.abzmin1,'String'));
        ordmax=str2double(get(handles.ordmax1,'String'));
        ordmin=str2double(get(handles.ordmin1,'String'));
        appmax=str2double(get(handles.appmax1,'String'));
        appmin=str2double(get(handles.appmin1,'String'));
        if abzmax<=abzmin
            if abzmin~=0
                abzmax=abzmin*1.01;
            else
                abzmax=1;
            end
            set(handles.abzmax1,'String',num2str(abzmax));
         end
        if ordmax<=ordmin
            if ordmin~=0
                ordmax=ordmin*1.01;
            else
                ordmax=1;
```

```
29              end
30
31              set(handles.ordmax1,'String',num2str(ordmax));
32          end
33          if appmax<=appmin
34              if appmin~=0
35                  appmax=appmin*1.01;
36              else
37                  appmax=1;
38              end
39              set(handles.appmax1,'String',num2str(appmax));
40          end
41
42
43          if app==1
44              plot(handles.dia1,rundu(:,abz),rundu(:,ord));
45
46              xlim(handles.dia1,[abzmin abzmax]);
47              ylim(handles.dia1,[ordmin ordmax]);
48          else
49
50              plot3(handles.dia1,rundu(:,abz),rundu(:,ord),rundu(:,app-1));
51              xlim(handles.dia1,[abzmin abzmax]);
52              ylim(handles.dia1,[ordmin ordmax]);
53              zlim(handles.dia1,[appmin appmax]);
54          end
55          else
56              xmin=min(rundu(:,abz));
57              xmax=max(rundu(:,abz));
58              ymin=min(rundu(:,ord));
59              ymax=max(rundu(:,ord));
60
61              if xmin==xmax
62                  xmin=xmin-0.5;
63                  xmax=xmax+0.5;
64              end
65              if ymin==ymax
66                  ymin=ymin-0.5;
67                  ymax=ymax+0.5;
68              end
69
70             if app==1
71              plot(handles.dia1,rundu(:,abz),rundu(:,ord));
72              set(handles.ordmin1,'String',num2str(ymin,'%4.2e'));
73              set(handles.ordmax1,'String',num2str(ymax,'%4.2e'));
74              set(handles.abzmin1,'String',num2str(xmin,'%4.2e'));
75              set(handles.abzmax1,'String',num2str(xmax,'%4.2e'));
76              set(handles.appmin1,'String',num2str(0,'%4.2e'));
77              set(handles.appmax1,'String',num2str(0,'%4.2e'));
78             else
79              zmin=min(rundu(:,app-1));
```

```
80              zmax=max(rundu(:,app-1));
81              if zmin==zmax
82                  zmin=zmin-0.5;
83                  zmax=zmax+0.5;
84              end
85              plot3(handles.dia1,rundu(:,abz),rundu(:,ord),rundu(:,app-1));
86              set(handles.ordmin1,'String',num2str(ymin,'%4.2e'));
87              set(handles.ordmax1,'String',num2str(ymax,'%4.2e'));
88              set(handles.abzmin1,'String',num2str(xmin,'%4.2e'));
89              set(handles.abzmax1,'String',num2str(xmax,'%4.2e'));
90              set(handles.appmin1,'String',num2str(zmin,'%4.2e'));
91              set(handles.appmax1,'String',num2str(zmax,'%4.2e'));
92            end
93          end
94  elseif flag==2
95          rundu(:,1)=tau;
96          rundu(:,6)=(rundu(:,2).^2+rundu(:,3).^2).^(1/2);
97
98          abz=(get(handles.abz2,'Value'));
99          ord=(get(handles.ord2,'Value'));
100         app=(get(handles.app2,'Value'));
101         if get(handles.auto2,'Value')==0
102         abzmax=str2double(get(handles.abzmax2,'String'));
103         abzmin=str2double(get(handles.abzmin2,'String'));
104         ordmax=str2double(get(handles.ordmax2,'String'));
105         ordmin=str2double(get(handles.ordmin2,'String'));
106         appmax=str2double(get(handles.appmax2,'String'));
107         appmin=str2double(get(handles.appmin2,'String'));
108         if abzmax<=abzmin
109             if abzmin~=0
110                 abzmax=abzmin*1.01;
111             else
112                 abzmax=1;
113             end
114             set(handles.abzmax2,'String',num2str(abzmax));
115         end
116         if ordmax<=ordmin
117             if ordmin~=0
118                 ordmax=ordmin*1.01;
119             else
120                 ordmax=1;
121             end
122             set(handles.ordmax2,'String',num2str(ordmax));
123         end
124         if appmax<=appmin
125             if appmin~=0
126                 appmax=appmin*1.01;
127             else
128                 appmax=1;
129             end
130             set(handles.appmax2,'String',num2str(appmax));
```

```
131         end
132
133         if app==1
134             plot(handles.dia2,rundu(:,abz),rundu(:,ord));
135             xlim(handles.dia2,[abzmin abzmax]);
136             ylim(handles.dia2,[ordmin ordmax]);
137         else
138             plot3(handles.dia2,rundu(:,abz),rundu(:,ord),rundu(:,app-1));
139             xlim(handles.dia2,[abzmin abzmax]);
140             ylim(handles.dia2,[ordmin ordmax]);
141             zlim(handles.dia2,[appmin appmax]);
142         end
143         else
144             xmin=min(rundu(:,abz));
145             xmax=max(rundu(:,abz));
146             ymin=min(rundu(:,ord));
147             ymax=max(rundu(:,ord));
148             if xmin==xmax
149                 xmin=xmin-0.5;
150                 xmax=xmax+0.5;
151             end
152             if ymin==ymax
153                 ymin=ymin-0.5;
154                 ymax=ymax+0.5;
155             end
156             if app==1
157               plot(handles.dia2,rundu(:,abz),rundu(:,ord));
158               set(handles.ordmin2,'String',num2str(ymin,'%4.2e'));
159               set(handles.ordmax2,'String',num2str(ymax,'%4.2e'));
160               set(handles.abzmin2,'String',num2str(xmin,'%4.2e'));
161               set(handles.abzmax2,'String',num2str(xmax,'%4.2e'));
162               set(handles.appmin2,'String',num2str(0,'%4.2e'));
163               set(handles.appmax2,'String',num2str(0,'%4.2e'));
164             else
165               zmin=min(rundu(:,app-1));
166               zmax=max(rundu(:,app-1));
167               if zmin==zmax
168                 zmin=zmin-0.5;
169                 zmax=zmax+0.5;
170               end
171               plot3(handles.dia2,rundu(:,abz),rundu(:,ord),rundu(:,app-1));
172               set(handles.ordmin2,'String',num2str(ymin,'%4.2e'));
173               set(handles.ordmax2,'String',num2str(ymax,'%4.2e'));
174               set(handles.abzmin2,'String',num2str(xmin,'%4.2e'));
175               set(handles.abzmax2,'String',num2str(xmax,'%4.2e'));
176               set(handles.appmin2,'String',num2str(zmin,'%4.2e'));
177               set(handles.appmax2,'String',num2str(zmax,'%4.2e'));
178             end
179         end
180 end
```

Abbildungsverzeichnis