

Designstudien für das KATRIN-Experiment

Diplomarbeit
von
Björn Flatt

Mainz 2001

Inhaltsverzeichnis

1	Einführung	1
2	Das Mainzer Experiment und KATRIN	5
2.1	Das Tritium- β -Spektrum	5
2.2	Experimenteller Aufbau	8
2.2.1	Prinzip des MAC-E-Filter	8
2.3	KATRIN	12
2.3.1	Gründe für ein neues großes Experiment	12
2.3.2	Die Komponenten von KATRIN	17
3	Anforderungen und Werkzeuge	21
3.1	Designanforderungen	21
3.1.1	Das Spektrometerdesign	21
3.1.1.1	Magnetdesign	21
3.1.1.2	Elektrostatisches Design	22
3.1.2	Das Transportsystem	23
3.2	Werkzeuge	24
3.2.1	Bfield_3d	24
3.2.1.1	sngl_coil_bfield_2d	26
3.2.1.2	sngl_coil_pot_2d	26
3.2.1.3	Berechnung der Magnetfeldlinien	28
3.2.1.4	Tests von Bfield_3D	28
3.2.2	SIMION 7.0	31
3.3	Die untersuchten Größen	32

4	Das Transportsystem	35
4.1	Entwurf der Firma ACCEL	35
4.2	Entwurf der Troitsker Arbeitsgruppe	48
4.3	Zusammenfassung und Wertung	58
5	Das Hauptspektrometer	61
5.1	Das Einelektroden-Design	61
5.2	Vakuumtank auf Potential	78
5.3	Zusammenfassung und Wertung	93
6	Das Vorspektrometer	95
6.1	Time-of-Flight Modus und Pulsung	96
7	Zusammenfassung und Ausblick	99
A	Quelltexte	103
A.1	BField3D	103
A.2	mag_3d	104
A.3	b_util	108
A.4	sngl_coil_pot_2d	122
A.5	sngl_coil_bfield_2d	124
A.6	math_vector	128

Abbildungsverzeichnis

2.1	Signatur einer Neutrinomasse	7
2.2	MAC-E-Filter	9
2.3	Transmissionsfunktion des MAC-E-Filters	11
2.4	Monte Carlo Simulation des Time of Flight Modus	13
2.5	Layout von KATRIN	15
2.6	Unterschiede in der Massensensitivität	16
2.7	Layout von KATRIN	18
2.8	Der magnetische Flußschlauch	19
3.1	Struktur von Bfield3D	25
3.2	Konvergenzkriterien für eine dünne Spule	27
3.3	Einfache SpulenKonfiguration	28
3.4	Vergleich der Magnetfeldberechnung von bfield_11 und Bfield3D . .	29
3.5	Test der Berechnung des magnetischen Potentials	30
3.6	Schema der Relaxation in SIMION	31
4.1	Das ACCEL-System	36
4.2	Die numerische Genauigkeit	38
4.3	Das ACCEL-System	41
4.4	Das ACCEL-System	42
4.5	Das ACCEL-System	43
4.6	Das ACCEL-System	44
4.7	Das ACCEL-System	45
4.8	Das ACCEL-System	46
4.9	Das ACCEL-System	47

4.10	Der Troitsk-Entwurf	49
4.11	Der Troitsk-Entwurf	51
4.12	Der Troitsk-Entwurf	52
4.13	Der Troitsk-Entwurf	53
4.14	Der Troitsk-Entwurf	54
4.15	Der Troitsk-Entwurf	55
4.16	Der Troitsk-Entwurf	56
4.17	Der Troitsk-Entwurf	57
5.1	Einelektroden-Design	63
5.2	Vakuumtank auf HV	64
5.3	Das Einelektroden-Design	67
5.4	Das Einelektroden-Design	68
5.5	Das Einelektroden-Design	69
5.6	Das Einelektroden-Design	70
5.7	Das Einelektroden-Design	71
5.8	Das Einelektroden-Design	72
5.9	Das Einelektroden-Design	73
5.10	Das Einelektroden-Design	74
5.11	Das Einelektroden-Design	75
5.12	Der Vakuumtank auf HV	76
5.13	Das Einelektroden-Design	77
5.14	Vakuumtank auf HV	79
5.15	Potentialdurchgriff des Tanks auf HV	79
5.16	Übergang zwischen Vor- und Hauptspektrometer	80
5.17	Vakuumtank auf HV	81
5.18	Das Einelektroden-Design	82
5.19	Der Vakuumtank auf HV	83
5.20	Der Vakuumtank auf HV	84
5.21	Der Vakuumtank auf HV	85
5.22	Der Vakuumtank auf HV	86
5.23	Der Vakuumtank auf HV	87

5.24	Der Vakuumtank auf HV	88
5.25	Der Vakuumtank auf HV	89
5.26	Der Vakuumtank auf HV	90
5.27	Der Vakuumtank auf HV	91
5.28	Der Vakuumtank auf HV	92
6.1	Vor- und Hauptspektrometer	96
6.2	Pulsung der Festkörperquelle	96
6.3	Simulation der Flugzeit in Vor- und Hauptspektrometer	98

Tabellenverzeichnis

4.1	ACCEL-Design eines Transportelements	36
4.2	Troitsk-Design eines Transportelements	48

Kapitel 1

Einführung

1930 standen die Physiker vor einem Problem, das so gravierend war, daß sie in Erwägung zogen, den Energie- und den Drehimpulserhaltungssatz in Frage zu stellen. Die Ursache für die Ratlosigkeit war das Energiespektrum der Elektronen des Betazerfalls. Man hatte einen Zweikörperzerfall erwartet, was zu einer monoenergetischen Linie geführt hätte. Stattdessen fand man ein kontinuierliches Spektrum.

Zur Lösung des Problems und zur Rettung der Energie- und Drehimpulserhaltung postulierte Wolfgang Pauli [Pau30] ein neutrales Teilchen mit Spin $\frac{1}{2}$, das er zunächst Neutron nannte. Vier Jahre später gelang Enrico Fermi [Fer34] mit Hilfe dieses Teilchens, das er als masselos ansetzte, eine theoretische Beschreibung des Betazerfalls. Er benannte das Teilchen in Neutrino um, da mittlerweile das neutrale Nukleon gefunden war.

Der experimentelle Nachweis des Neutrinos gelang Frederick Reines und Clyde Cowan 1956. Sie beobachteten die Wechselwirkung eines Elektronantineutrinos aus einem Reaktor mit einem Proton einer wässrigen Kadmiumchlorid-Lösung:

$$\bar{\nu}_e + p \longrightarrow n + e^+ \quad (1.1)$$

Für diese Reaktion ergab sich ein mittlerer Wirkungsquerschnitt von $\sigma = (1.1 \pm 0.3) \cdot 10^{-43} \text{cm}^2$. Dieser Wert steht im Einklang mit der um V-A erweiterten Fermitheorie für die Neutrinoenergien von $E_\nu \leq 8 \text{ MeV}$. Die Entdeckung des Elektronneutrinos wurde 1995 mit dem Nobelpreis ausgezeichnet. Ledermann, Schwartz und Steinberger wiesen 1962 das Myonneutrino nach, wofür sie 1988 den Nobelpreis erhielten. Das Tauneutrino wurde als drittes Neutrino erst im Jahr 2000 vom DONUT-Experiment am Fermilab nachgewiesen [Don00].

Die drei Neutrinos (ν_e, ν_μ, ν_τ) bilden zusammen mit Elektron, Myon und Tau jeweils ein schwaches Isospindublett. Diese sechs Teilchen sind die Leptonen, zusammen mit den Quarks, die ebenfalls in drei Dubletts auftreten ((u,d) , (c,s) , (b,t)), sind sie nach dem Standardmodell der Teilchenphysik die fundamentalen Fermionen aus denen die gesamte Materie aufgebaut ist. Ordnet man die Quark- und Leptondubletts jeweils nach steigender Masse, so erhält man drei Familien.

Über die Massen der Fermionen macht das Standardmodell keine Aussagen, so daß diese experimentell bestimmt und von außen in die Theorie eingefügt werden müssen. Bis vor kurzem gelang noch kein experimenteller Nachweis von Neutrinomassen, daher wurden die Massen der drei Neutrinoarten auf Null gesetzt. In Theorien jenseits des Standardmodells (sog. neue Physik, z.B. Supersymmetrie, Grand Unified Theories (GUT)) werden den Neutrinos im allgemeinen endliche Ruhemassen zugewiesen.

Neutrinos spielen auch in der Kosmologie und der Astrophysik eine tragende Rolle. Das Interesse reicht von der Nukleosynthese leichter Kerne während des Urknalls über die Entwicklung und Strukturgebung des Universums bis zu Supernovae. Besonders hervorzuheben sind die Hintergrundneutrinos, die nach der Urknalltheorie analog zu den Photonen der kosmischen Mikrowellenhintergrundstrahlung auftreten müssen. Das Verhältnis dieser sog. Relic Neutrinos zu Baryonen (Protonen, Neutronen,...) ist $10^9 : 1$. Daher sind schon kleine Neutrinomassen von Interesse, da sie einen wichtigen Beitrag zur Dunklen Materie leisten. Es existieren verschiedene Modelle zur Beschreibung der Entwicklung des Universums, die von der Art der Dunklen Materie abhängen. Man unterscheidet Kalte Dunkle Materie und Heiße Dunkle Materie. Neutrinos sind die einzigen bekannten Kandidaten für Heiße Dunkle Materie.

Auf der internationalen Konferenz 'Neutrino 1998' sorgte das SuperKamiokande-Experiment für großes Aufsehen, als Evidenzen für von Null verschiedene Neutrinomassen bekannt gegeben wurden [Fuk98]. Bei diesem Experiment handelt es sich um ein Neutrinooszillationsexperiment, das Neutrinos, die in der Erdatmosphäre durch den Beschuß mit kosmischer Strahlung entstehen, untersucht. Die zugrunde liegende Theorie geht davon aus, daß sich Neutrinos einer bestimmten Familie auf dem Weg von der Quelle zum Detektor in Neutrinos einer anderen Familie umwandeln. Diese Umwandlung funktioniert nur, wenn man zwischen den schwachen Eigenzuständen (ν_e, ν_μ, ν_τ) und den Masseneigenzuständen (ν_1, ν_2, ν_3) unterscheidet. Desweiteren müssen die Massen ($m_{\nu_1}, m_{\nu_2}, m_{\nu_3}$) verschieden sein. Oszillationsexperimente sind nur auf die Unterschiede der Massenquadrate sensitiv und daher nicht in der Lage die absolute Massenskala für Neutrinos festzulegen. Die von SuperKamiokande veröffentlichten Daten bevorzugen kleine Massenquadratdifferenzen im Bereich von $10^{-2} \text{eV}^2/c^4$ für Oszillationen von $\nu_\mu \rightarrow \nu_\tau$. Neutrinooszillationen sind sehr wahrscheinlich auch die Erklärung für den zu niedrigen Neutrinofluß, den die sog. Sonnenneutrinoexperimente (Gallex, Sage,...) messen. Diese Experimente registrieren die von der Sonne kommenden Neutrinos und vergleichen ihre Daten mit Modellen der Sonnenprozesse.

Weitere Hinweise auf Neutrinooszillationen lieferte LSND, das Beschleunigerexperiment fand Massenquadratdifferenzen von $1 \text{eV}^2/c^4$ für Oszillationen von $\bar{\nu}_\mu \rightarrow \bar{\nu}_e$. [Ath95]. Diese Evidenz wurde jedoch nicht von dem KARMEN-Experiment bestätigt [Eit00], und steht teilweise sogar in Widerspruch zu KARMEN.

Ein möglicher Zugang zur Neutrinomasse ist der neutrinolose doppelte β -Zerfall. Bei einem doppelten β -Zerfall laufen in einem Kern zwei β -Zerfälle gleichzeitig ab,

d.h. bei einem doppelten β^- -Zerfall werden zwei $\bar{\nu}_e$ emittiert. Dieser Prozeß wurde 1987 zum ersten Mal nachgewiesen [Ell87]. Im Gegensatz zum normalen doppelten β -Zerfall werden beim neutrinolosen doppelten β -Zerfall die Elektronantineutrinos nicht emittiert, sondern als virtuelle Teilchen zwischen den beteiligten Kernen ausgetauscht. Dieser Vorgang setzt voraus, daß Neutrinos Majorana-Teilchen sind, d.h. daß Teilchen und Antiteilchen identisch sind. Desweiteren muß das Neutrino eine Masse haben, da nur durch eine von Null verschiedene Masse ein Zustand gemischter Helizität möglich wird, die vorliegt wenn das Neutrino am ersten Zerfallsvertex als linkshändiges Teilchen emittiert und am zweiten Vertex als rechtshändiges Teilchen absorbiert wird. Ein weiterer limitierender Faktor ist, daß diese Experimente nur auf eine effektive Neutrinomasse $m_{ee}(\nu)$ sensitiv sind. Bei dieser effektiven Neutrinomasse handelt es sich um die kohärente Summe aller Neutrinomasseneigenzustände ν_i , die zum Elektronneutrino ν_e auf Grund der Mischungsmatrix U_{ei} beitragen.

$$m_{ee} = \left| \sum_{i=1}^3 U_{ei}^2 \cdot m(\nu_i) \right| \quad (1.2)$$

Da die Matrixelemente im allgemeinen komplex sind, kann es zu Auslöschungen kommen und $m_{ee}(\nu)$ Null werden, obwohl die einzelnen Massenwerte $m(\nu_i)$ von Null verschieden sind. Das zur Zeit sensitivste Experiment ist das Heidelberg-Moskau-Experiment [Die00] an ^{76}Ge . Das Experiment hat bisher keinen Zerfall beobachten können, woraus eine Massengrenze für den Fall eines Majoranateilchens von $m_{ee}c^2 \leq 0.34\text{eV}$ (90% CL) folgt.

Einen direkten Zugang zu Neutrinomassen findet man über die Untersuchung der Kinematik schwacher Zerfälle.

Die Untersuchung des Tritium β -Spektrums im Endpunktsbereich liefert einen direkten Zugang zur Neutrinomasse, der frei von theoretischen Annahmen außer der Energie- und Impulserhaltung ist. Zur Zeit gibt es zwei laufende Experimente in Mainz und Troitsk, die diese Untersuchung vornehmen. Die Obergrenzen der Masse des Elektronantineutrinos liegt nach diesen Gruppen bei $2.5\text{ eV}/c^2$ (Troitsk) [Lob99] bzw. $2.2\text{ eV}/c^2$ [Bon01]. Diese Experimente sind auf

$$m^2(\nu_e) = \sum_{i=1}^3 |U_{ei}|^2 \cdot m^2(\nu_i) \quad (1.3)$$

sensitiv. Hier sind die Quadrate der Matrixelemente involviert, so daß keine Auslöschungen auftreten können und die Neutrinomasse bestimmt werden kann. Mit diesen Ergebnissen haben beide Experimente annähernd ihre Sensitivitätsgrenzen erreicht. Zur Erhöhung der Sensitivität, die zur Festlegung der absoluten Neutrinomassenskala und zur Verifikation kosmologischer Modelle nötig ist, wird derzeit ein um ein Vielfaches größeres Experiment geplant, das am Forschungszentrum Karlsruhe durchgeführt werden soll. Das Ziel von KATRIN, dem Karlsruhe TRitium Neutrinoexperiment, ist es eine Sensitivität auf die Elektronantineutrinomasse im sub-eV-Bereich zu erreichen.

Gegenstand dieser Arbeit ist eine Designstudie einiger Komponenten des geplanten Experimentes. Dazu wurden zunächst Programme erstellt, die es erlauben Magnetfelder in dreidimensionalen kartesischen Koordinaten zu berechnen. Mit Hilfe dieser Programme wurden anschließend Simulationen verschiedener Designs des Spektrometers und des Transportsystems durchgeführt. Die Arbeit ist wie folgt gegliedert:

- Im zweiten Kapitel wird das Tritium-Spektrum und der Einfluß der Neutrinomasse auf das Spektrum beschrieben. Weiterhin wird die prinzipielle Wirkungsweise des MAC-E-Filters erläutert. Der letzte Abschnitt des Kapitels motiviert den Bau eines neuen großen Experiments und erläutert die geplanten Komponenten eines solchen Experiments.
- Das dritte Kapitel beginnt mit einer Erläuterung der Designkriterien für die verschiedenen Experimentkomponenten. Im weiteren Verlauf werden die Programme, die im Laufe dieser Arbeit erstellt wurden beschrieben. Diese Programme bilden das Fundament für die in den folgenden Kapiteln erklärten Simulationen. Die durch diese Simulationen untersuchten Größen werden ebenfalls im dritten Kapitel beschrieben.
- Das vierte Kapitel befaßt sich mit den Simulationen zum Transportsystem. Es wurden zwei Designvorschläge untersucht und verglichen. Die Ergebnisse sind hier dargestellt.
- Kapitel fünf widmet sich den Simulationen des zukünftigen Spektrometers. Auch hier wurden zwei mögliche Designs miteinander verglichen.
- Im sechsten Kapitel werden die gewonnenen Ergebnisse zusammengefaßt und bewertet. Desweiteren wird ein Ausblick auf noch offene Fragen und Problemstellungen gegeben.

Kapitel 2

Das Mainzer Experiment und KATRIN

In diesem Kapitel wird das Prinzip des Mainzer Neutrinomassen Experiments vorgestellt, da es sich bei KATRIN, dem geplanten KARlsruhe TRItium Neutrino Experiment, um einen Aufbau gleicher Funktionsweise handelt. Zunächst wird das Tritium- β -Spektrum erläutert und gezeigt wie die Neutrinomasse das β -Spektrum im Endpunktsbereich beeinflusst. Weiterhin wird die Wirkungsweise des Mainzer Spektrometers (MAC-E-Filter) erläutert. Den Abschluß des Kapitels bilden die physikalische Motivation eines neuen Experiments, sowie die Beschreibung der Komponenten einer solchen Apparatur.

2.1 Das Tritium- β -Spektrum

Das neutronenreiche Wasserstoffisotop Tritium unterliegt dem β^- -Zerfall, bei dem sich ein Neutron im Atomkern unter Emission eines Elektrons und eines Elektronantineutrinos in ein Proton umwandelt. Die Kernladungszahl Z erhöht sich also um 1, während die Massenzahl A konstant bleibt:



An dieser Formel erkennt man, daß es sich bei einem β -Zerfall um einen Dreikörperzerfall handelt. Man erwartet daher ein kontinuierliches Elektronenspektrum. Die genaue Form des Spektrums beruht auf Fermis Goldener Regel und läßt sich theoretisch durch die Phasenraumdichte der möglichen Endzustände des Elektrons und des Neutrinos und das Quadrat des Kernmatrixelements $|M|^2$ beschreiben. Für das Spektrum der kinetischen Energie der Elektronen E gilt dann:

$$\frac{dN}{dE} = \frac{G_f^2 m_e^5 c^4}{2\pi^3 \hbar^7} \cos^2(\Theta_c) |M|^2 F(Z, E) p(E + m_e c^2) (E_0 - E) \sqrt{(E_0 - E)^2 - m_\nu^2 c^4} \quad (2.2)$$

mit : G_f : Fermi-Kopplungskonstante

Θ_c : Cabbibo-Winkel

M : Kernmatrixelement

F : Fermi-Funktion

E_0 : β -Endpunkt

m_e, m_ν : Ruhemasse des Elektrons und des Neutrinos.

Die gesuchte Neutrinomasse geht hier quadratisch ein, die Observable der Untersuchung des Tritium- β -Spektrums ist also $m_\nu^2 c^4$.

In Gleichung (2.2) wird der Zerfall eines 'nackten Atomkerns' beschrieben, als Quelle dient in Mainz und Troitsk jedoch molekulares Tritium. Da hier Hüllenelektronen auftreten, muß das Spektrum modifiziert werden. Die Elektronen können neben dem Grundzustand auch angeregte Zustände des Tochterkerns besetzen. Die Wahrscheinlichkeit dafür kann mit Hilfe der 'sudden approximation'-Methode berechnet werden. Bei dieser Näherung geht man davon aus, daß die Umwandlung des Kerns schnell gegenüber den in der Elektronenhülle vorgehenden Prozessen ist, d.h. für die Hüllenelektronen ändert sich die Kernladungszahl plötzlich. Die Wahrscheinlichkeit für einen Übergang $i \rightarrow n$ läßt sich mit Hilfe des Überlappintegrals

$$W_{i \rightarrow f_n} = | \langle \varphi_{f_n} | \varphi_i \rangle |^2 \quad (2.3)$$

durch Einsetzen der Wellenfunktion φ_i vor den Zerfall und den möglichen Endzuständen φ_{f_n} berechnen.

Damit ergibt sich das Spektrum eines Tritiumatoms oder -moleküls aus der Summe der mit $W_{i \rightarrow f_n}$ gewichteten Einzelspektren der Zerfälle in die möglichen Endzustände, wobei auch Kontinuumszustände berücksichtigt werden:

$$\frac{dN}{dE} = \text{const} \cdot |M|^2 F(Z, E) p(E + m_e c^2) \sum_n W_{i \rightarrow f_n} \varepsilon_n \sqrt{\varepsilon_n^2 - m_\nu^2 c^4} \quad (2.4)$$

mit $\varepsilon_n = E_0 - V_n - E$.

In Abb. (2.1) ist deutlich der Einfluß der Neutrinomasse auf den Endpunktsbereich des β -Spektrums zu erkennen. Die Mindestenergie des emittierten Neutrinos beträgt $m_\nu c^2$, dieser Anteil der Zerfallsenergie steht dem freiwerdenden Elektron nicht mehr zur Verfügung, so daß das Spektrum vor dem Endpunkt E_0 abbricht.

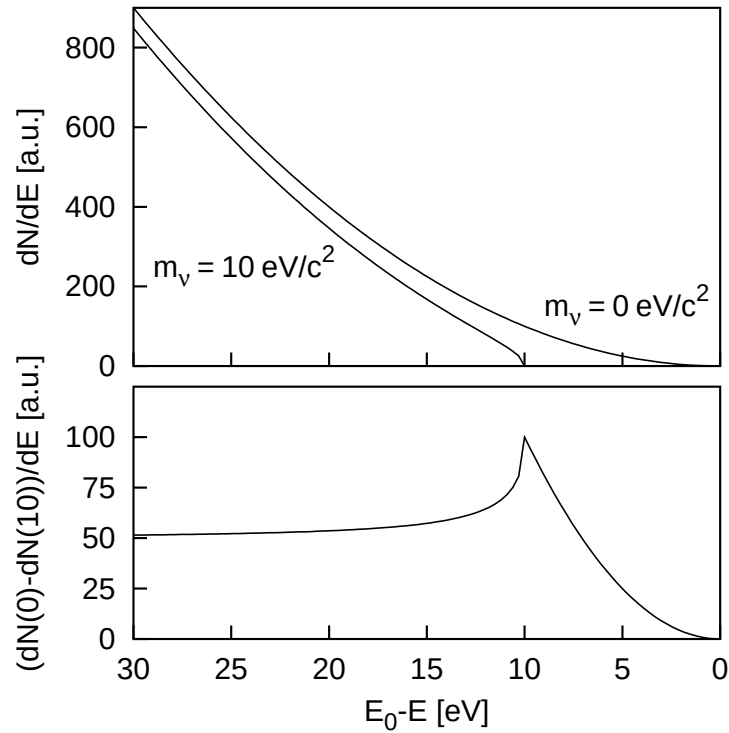


Abbildung 2.1: Signatur einer von Null verschiedenen Neutrinomasse. Oben dargestellt ist der Endpunktsbereich des Tritium- β -Spektrums für zwei angenommene Neutrinomassen von 0 und 10 eV. Im unteren Bild ist die Differenz der beiden Spektren, die Signatur einer endlichen ν -Ruhemasse aufgetragen. Sie ist in der Nähe des β -Endpunktes am stärksten, verschwindet jedoch nirgendwo ganz.

Eine Voraussetzung für eine Bestimmung der Neutrinomasse ist die genaue Vermessung des β -Spektrums im Endpunktsbereich, also dort, wo die Zählrate $\frac{dN}{dE}$ klein ist. Hieraus ergeben sich die Vorteile von Tritium als Elektronenquelle:

- Die Endpunktsenergie ist mit $E_0 = 18.6$ keV niedrig, so daß der auf die Neutrinomasse sensitive Teil des Spektrums relativ groß ist.
- Tritium hat eine kurze Halbwertszeit von 12.3 Jahren, wodurch hohe Zählraten bei einer geringen Quelldicke erreicht werden können. Dies ist wichtig, da in hohen Quelldicken verstärkt Energieverluste auftreten, die das Spektrum verformen.
- Die Endzustände des Tritiumzerfalls lassen sich berechnen, was für die theoretische Beschreibung (Gl. (2.4)) des Spektrums wichtig ist.

Aus diesen Bedingungen ergeben sich Anforderungen an den Aufbau eines Experiments:

- Akzeptierter Raumwinkel:
Auf Grund der niedrigen Zählrate im Endpunktsbereich des β -Spektrums muß das Spektrometer einen möglichst großen akzeptierten Raumwinkel aufweisen.

- Energieauflösung:
Die Energieauflösung sollte im Bereich der statistischen Empfindlichkeitsgrenze für die Neutrinomasse liegen, auf die man sensitiv sein will. Dies entspricht bei einer Empfindlichkeit auf $m_{\bar{\nu}_e} = 2 \text{ eV}/c^2$ einer relativen Auflösung $\frac{\Delta E}{E}$ von 10^{-4} .
- Untergrund: Das Spektrometer sollte im Betrieb eine möglichst geringe Untergrundzählrate aufweisen.

Das in Mainz entwickelte Spektrometer, das im nächsten Abschnitt beschrieben wird, entspricht diesen Anforderungen.

2.2 Experimenteller Aufbau

Das Mainzer Experiment besteht im wesentlichen aus drei Teilen:

- der Tritiumquelle
- dem Spektrometer
- dem Detektor

Im weiteren Verlauf wird das Prinzip des Spektrometers, das auf der magnetischen, adiabatischen Kollimation mit einem nachfolgenden elektrostatischen Filter beruht, erläutert. Bei der Tritiumquelle handelt es sich um einen schockkondensierten Tritiumfilm, der auf 1.9 K gekühlt wird. Dieser Film hat einen Durchmesser von 17 mm und eine durchschnittliche Dicke von 480 Å, dies entspricht einer Aktivität von etwa 20 mCi. Der Detektor ist eine fünffach segmentierte Si-Diode, jedes dieser Segmente hat eine Fläche von 1 cm². Die Energieauflösung des Detektors beträgt etwa 1.4 keV (FWHM). Der Detektor wird mit einer Standardelektronik ausgelesen. Auf eine detaillierte Beschreibung der Quelle und des Detektors wird an dieser Stelle verzichtet und für eine genaue Beschreibung auf [Bor00] verwiesen.

2.2.1 Prinzip des MAC-E-Filter

Die Funktionsweise des MAC-E-Filters basiert auf der Überlagerung magnetischer und elektrostatischer Felder (siehe Abb. 2.2 a). Das zur Mittelebene des Spektrometers symmetrische Magnetfeld wird von zwei supraleitenden Solenoiden erzeugt. Das elektrische Feld, das radialsymmetrisch ist, wird durch Zylinderelektroden erzeugt, die den magnetischen Flußschlauch umgeben.

Im linken Magneten befindet sich im Magnetfeld B_S die isotrop abstrahlende β -Quelle. Elektronen, die in den vorderen Halbraum austreten, werden so auf Zyklotronbahnen um die Magnetfeldlinien gezwungen. Da das Magnetfeld zur Spektrometernmitte hin abnimmt, wirkt eine Gradientenkraft

$$\vec{F}_\nabla = \vec{\nabla} (\vec{\mu} \cdot \vec{B}), \quad (2.5)$$

1) gilt daher für die Beziehung zwischen Energie und Magnetfeldstärke:

$$E_{\perp f} = E_{\perp i} \cdot \frac{B_f}{B_i} \quad (2.7)$$

Die Energie der Elektronen liegt also in der Spektrometermitte (bei B_{min}), bei einem Verhältnis $\frac{B_{min}}{B_{max}} \approx \frac{1}{4000}$, fast ausschließlich als Parallelenergie vor. Für die Auflösung des Spektrometers folgt:

$$\Delta E = E_{\perp f, max} = E_0 \cdot \frac{B_{min}}{B_{max}} \approx 4.8 \text{ eV} \quad (2.8)$$

wobei E_0 die Startenergie und damit die maximale Energie in der Zyklotronbewegung ist.

Das bereits erwähnte Gegenpotential sorgt nun dafür, daß nur Elektronen deren Parallelenergie größer als das elektrische Gegenpotential $-eU_0$

$$E_{\parallel f} = E_0 - E_{\perp f} > -e \cdot U_0 \quad (2.9)$$

ist, die elektrische Analysierebene passieren und zum Detektor im rechten Spektrometer-Solenoiden gelangen können.

Das Verhältnis der von der Quelle mit einer bestimmten Startenergie E_S emittierten Elektronen, die das Analysierpotential überwinden, zu allen von der Quelle in den vorderen Halbraum emittierten Elektronen, wird durch die sog. Transmissionsfunktion T_H beschrieben. Die die Transmission limitierenden Faktoren sind wie bereits erwähnt das Analysierpotential, sowie die Elektronenquelle:

- Der Tritiumfilm sitzt nicht direkt im Feldmaximum des Solenoiden, sondern kurz davor im Feld B_S (s. Abb. (2.2)). Das führt dazu, daß der akzeptierte Raumwinkel nicht 90° beträgt, sondern durch das Verhältnis des Magnetfeldes am Ort der Quelle B_S zum maximalen Magnetfeldes B_{max} bestimmt wird. Elektronen deren Startwinkel größer sind als

$$\theta_{max} = \arcsin \left(\sqrt{\frac{B_S}{B_{max}}} \right) \quad (2.10)$$

werden also vor dem Spektrometereingang magnetisch reflektiert. Auf diese Weise werden Elektronen ausgefiltert, die die Quelle unter großen Winkeln und damit langen Wegen und hohen Streuwahrscheinlichkeiten verlassen haben, und so der mögliche Energieverlust der Elektronen niedrig gehalten. Wenn man nur Winkel bis θ_{max} zuläßt ergibt sich eine auf 1 normierte Transmissionsfunktion T_H^0 .

- Die Parallelenergie $E_{\parallel}$ der Elektronen muß in jedem Punkt des Spektrometers positiv sein.

Die Transmissionsfunktion ist somit eine Funktion von der Startenergie der Elektronen E_S , des Startwinkels θ_S und damit der Magnetfelder zwischen der Quelle und der Spektrometermitte, und auch des Retardierungspotentials U_0 . Definiert man $\varepsilon = E - eU_0$, dann beschreibt dieser Ausdruck die Überschußenergie in der Analysierebene, und es gilt:

$$T_H^0(\varepsilon) = \begin{cases} 0 & \text{für } \varepsilon < 0 \\ \frac{1 - \sqrt{1 - \frac{\varepsilon}{E} \frac{B_S}{B_A}}}{1 - \sqrt{1 - \frac{\Delta E}{E} \frac{B_S}{B_A}}} & \text{für } 0 \leq \varepsilon \leq \Delta E \\ 1 & \text{für } \varepsilon > \Delta E \end{cases} \quad (2.11)$$

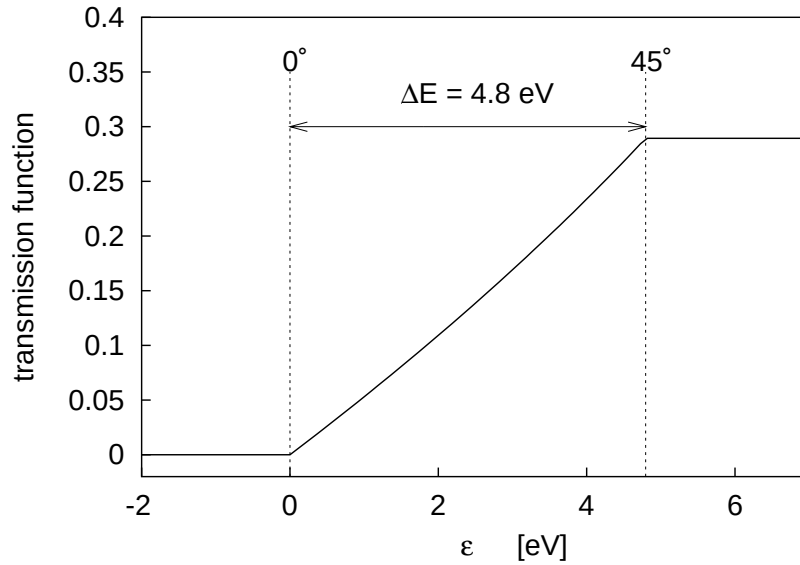


Abbildung 2.3: Die Transmissionsfunktion des MAC-E-Filters für eine isotrop abstrahlende Quelle. Dargestellt ist das Verhältnis der von der Quelle mit einer Energie E_S emittierten Elektronen, die das Analysierpotential überwinden können, zu der Gesamtzahl der in den vorderen Halbraum gestarteten Elektronen. Die Breite des Anstiegs entspricht der Auflösung des Spektrometers. Die hier aufgetragene Funktion wurde nicht auf 1 normiert.

Bei dem bisher beschriebenen Spektrometer handelt es sich um integrierendes Spektrometer, das wie ein Hochpassfilter wirkt, alle Elektronen oberhalb einer Energieschwelle werden im Detektor nachgewiesen. Dieser Operationsmodus erweist sich als ungeeignet, wenn man lokale Strukturen im β -Spektrum weit unterhalb des Endpunktes untersuchen möchte (z.B. Troitskanomalie). Um die Vorteile eines nicht integrierenden Spektrometers zu nutzen, kann die Mainzer Apparatur im Flugzeitmodus (TOF) betrieben werden [Bon99]. Um diesen Ansatz zu verfolgen, geht man vereinfachend davon aus, daß die Potentialbarriere mit dem Potential U_A über eine

Länge L konstant ist, aber sonst keine Retardierung auf die Elektronen wirkt. Für die Flugzeit der Elektronen gilt dann:

$$t_L = \frac{L}{v_{||,A}} = \frac{L}{\sqrt{2(\varepsilon - E_{\perp,A})/m}} \quad (2.12)$$

$E_{||}$ muß Gl.(2.9) erfüllen. Verlangt man eine minimale Driftzeit $t_{L,min}$, so kann man schnelle Teilchen mit

$$E_{||,f} > E_{||,f,max} = \frac{mL^2}{2t_{L,min}^2} \quad (2.13)$$

ausschließen (siehe Abb. (2.4 c)). Wenn man nun $E_{||,f} = \varepsilon = E_{||,f} - E_{\perp,f}$ anwendet, erhält man einen Filter für $T_L(\varepsilon)$, was einem Tiefpaßfilter entspricht.

$$T_L = \begin{cases} 1 & \text{für } \varepsilon < E_{||,f,max} \\ 1 - T_H(\varepsilon - E_{||,f,max}) & \text{für } \text{sonst} \\ 0 & \text{für } \varepsilon < E_{||,f,max} + \Delta E \end{cases} \quad (2.14)$$

Im MAC-E-TOF-Modus operiert das Spektrometer als ein Bandpaßfilter, dessen Transmissionsfunktion durch

$$T_{BP} = T_L \cdot T_H^0 \quad (2.15)$$

beschrieben wird.

2.3 KATRIN

Bei KATRIN, dem KARlsruhe TRItium Neutrinoexperiment handelt es sich um ein geplantes Neutrinomassenexperiment am Forschungszentrum Karlsruhe (FZK). Das Ziel dieses Experiments ist es auf Elektronenneutrinomasse im sub-eV Bereich sensitiv zu werden. Das Prinzip des Experiments beruht auf dem in Abschnitt 2.2 beschriebenen MAC-E-Filter.

2.3.1 Gründe für ein neues großes Experiment

Nach der Bestimmung der Masse des Top-Quarks sind die Neutrinos die einzigen Teilchen des Standardmodells, deren Masse noch nicht experimentell festgelegt werden konnte. Die jüngsten Ergebnisse von solaren und atmosphärischen Neutrinos können praktisch nur durch Neutrinooszillationen erklärt werden und zeigen daher, daß die Neutrinomasse von Null verschieden ist. Die 1998 vorgestellten Evidenzen für Oszillationen atmosphärischer Neutrinos durch das SuperKamiokande- Experiment haben sich weiter verstärkt. Diese Ergebnisse werden auch von anderen Experimenten wie Macro und Soudan 2 unterstützt. Auch das Defizit, das im Fluß der solaren Neutrinos auftritt, scheint durch Neutrinooszillationen erklärbar zu sein. Gerade die

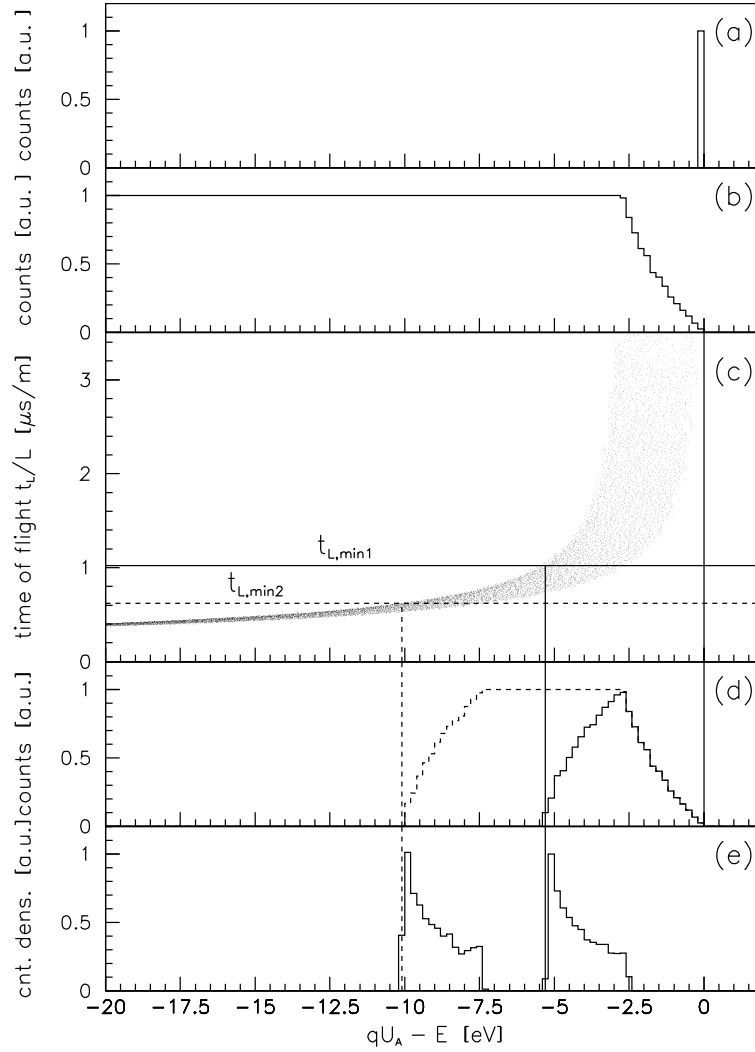


Abbildung 2.4: Monte Carlo Simulation des Time of Flight Modus. Dargestellt ist die MC-Simulation einer monoenergetischen Linie (a). (b) zeigt die Anzahl der durchgelassenen Elektronen im MAC-E-Filter Modus. Abb.(c) zeigt die Verteilung der Ereignisse indem die Flugzeit gegen die Energie aufgetragen wird, hier wird der Energieschnitt durch die Bestimmung einer minimalen Flugzeit deutlich. Das durchgelassene Spektrum im MAC-E-TOF Modus ist in (d) dargestellt. (e) zeigt zwei Spektren transmittierter Elektronen für feste Flugzeiten $t_{L,min1}, t_{L,min2}$, für unsere Anwendung sind diese aber nicht relevant. Diese Abbildung ist [Bon99] entnommen.

neuesten SNO Resultate zeigen im Vergleich zu den SuperKamiokande Daten, daß die Sonnenneutrinos nicht nur als Elektronneutrinos, sondern auch als Tau- bzw. Myonneutrinos auf der Erde ankommen.

Falls Neutrinooszillationen auftreten, ist dies ein Hinweis darauf, das Neutrinos eine endliche Ruhemasse haben. Da Oszillationsexperimente jedoch nur auf Unterschiede im Massenquadrat Δm^2 sensitiv sind kann, aufgrund der vorliegenden Ergebnisse

keine Aussage über die absolute Neutrinomassenskala getroffen werden. In der Theorie gibt es verschiedene Modelle für die Anordnung der Neutrinomassen: hierarchische Modelle ($m_3 \gg m_2 \gg m_1$) und degenerierte Modelle ($m_3 \approx m_2 \approx m_1$). Eine Unterscheidung der Modelle kann nur aufgrund einer direkten Massenbestimmung des Neutrinos getroffen werden, die eine Empfindlichkeit im sub-eV Bereich erreicht.

In Theorien, die über das Standardmodell hinausgehen erhalten Neutrinos meist eine von Null verschiedene Masse, so daß die Beobachtung der Größe der Neutrinomasse ein wichtiges Unterscheidungskriterium zwischen verschiedenen Theorien wäre.

In der Astrophysik und der Kosmologie sind Neutrinos, aufgrund ihrer hohen Anzahl, ebenfalls von großer Bedeutung. Nach der Urknalltheorie sind im Universum, analog zu den Photonen der kosmischen Mikrowellenhintergrundstrahlung, riesige Mengen an Neutrinos verteilt. Das Verhältnis dieser sog. Relic Neutrinos zu baryonischen Teilchen ist $10^9 : 1$. Aufgrund dieser Häufigkeit würden selbst geringe Neutrinomassen einen bedeutenden Beitrag zur Dunklen Materie liefern. Bei Dunkler Materie handelt es sich um Materie, die nicht direkt beobachtet werden kann, ihre Existenz ist aber z.B. durch die Messung von Radialabhängigkeiten der Sternengeschwindigkeit in Galaxien und anderen Beobachtungen indirekt erwiesen. Der Anteil der Dunklen Materie an der gesamten Masse des Universums scheint mehr als die Masse aller leuchtenden Materie (Sterne und Gas) und sogar als die Masse der gesamten baryonischen Materie zu betragen.

Untersuchungen der kosmischen Mikrowellenhintergrundstrahlung deuten auf ein flaches Universum hin. Ein flaches Universum läßt sich durch die relative Massendichte Ω beschreiben, sie gibt das Verhältnis der Massendichte des Universums zu der kritischen Massendichte, die notwendig ist, um ein flaches Universum zu beschreiben.

Abb. (2.5 zeigt die Beiträge der verschiedenen Materiearten zu Ω , wie sie direkt oder indirekt aus Experimenten gewonnen wurden. Der größte Teil, etwa 60 - 70 % liegt in der kosmologischen Konstanten, die schon von Einstein gefordert, jedoch später von ihm selbst wieder verworfen wurde. Der nächst große Anteil liegt mit etwa 30 % bei der dunklen Materie. Hinweise auf den Anteil an baryonischer dunkler Materie, der etwa 5% beträgt, erhält man aus Theorien zur Nukleosynthese während des Urknalls und aus dem Fehlen seltener Wasserstoffisotope wie Deuterium. Der Massenanteil an sichtbarer Materie wird durch Teleskopbeobachtungen gestützt.

Es gibt verschiedene Modelle zur Beschreibung der Strukturbildung des Universums, die abhängig von der Art der Dunklen Materie sind. Man unterscheidet zwischen Heißer Dunkler Materie und Kalter Dunkler Materie, was zur Zeit der Abkoppelung relativistischer, bzw. nichtrelativistischer Teilchen entspricht. Neutrinos endlicher Ruhemasse sind die einzigen bekannten Kandidaten für Heiße Dunkle Materie [Tur99]. Auch hier ist eine Sensitivität auf die Neutrinomasse im sub-eV Bereich notwendig, um die verschiedenen Modelle zu verifizieren.

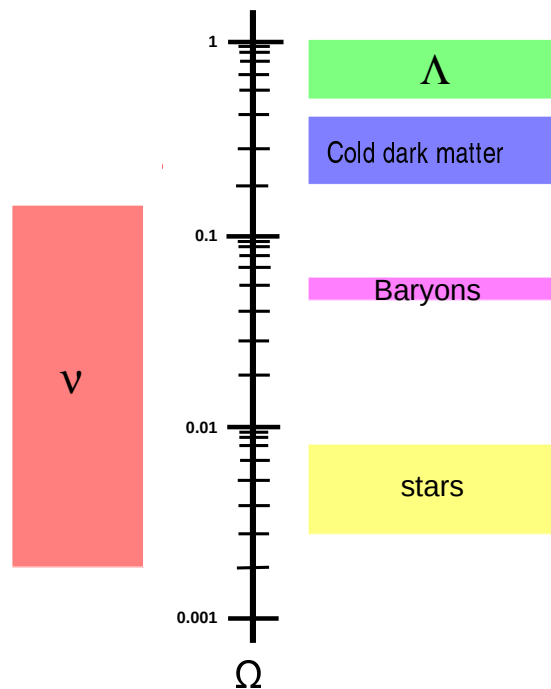


Abbildung 2.5: Omega Es sind die Massenanteile der verschiedenen Materiearten an der Massendichte Ω des Universums aufgetragen. Die Neutrinomasse wird dabei nach unten durch die SuperKamiokande Daten und nach oben durch die Strukturbildung beschränkt.

Eine Möglichkeit, den Anteil der Heißen Dunklen Materie zu bestimmen, ist die Untersuchung der Anisotropien der kosmischen Mikrowellenhintergrundstrahlung mit Ballon- oder Satellitenexperimenten. Die Interpretation der Meßdaten ist aber wiederum modellabhängig und setzt weiterhin die genaue Kenntnis von kosmologischen Parametern wie der Hubblekonstante H_0 und der kosmologischen Konstante Λ voraus. Wäre eine Neutrinomasse, oder eine Obergrenze für die Neutrinomasse im sub-eV Bereich bekannt, wäre dies auch eine wertvolle Information für die Interpretation der Ergebnisse der Satellitenexperimente der nächsten Generation wie MAP und PLANCK.

Es wird deutlich, daß die Neutrinomasse ein Parameter ist, der sowohl für die Teilchenphysik, als auch für die Kosmologie von Bedeutung ist. Sie spielen nicht nur bei der oben beschriebenen Evolution und Strukturbildung des Universums eine große Rolle, sondern auch bei anderen Prozessen wie Supernovaexplosionen sind Neutrinomassen und Neutrinomischungen von Bedeutung. Der interessante Massenbereich liegt dabei im sub-eV Bereich.

Der direkte Zugang zu Neutrinomassen in diesem Bereich kann durch die Untersuchung des neutrinolosen doppelten β -Zerfall, oder durch die Kinematik schwacher

Zerfälle, wie den β -Zerfall von Tritium, untersucht werden. Die Ergebnisse, die man durch die Untersuchung des neutrinolosen doppelten β -Zerfall erhält, hängen stark von der Mischung der Neutrinos ab, insbesondere von dem Mischungswinkel, der die solaren Neutrinooszillationen bestimmt. [Far01]. Die Ergebnisse von Oszillationsexperimenten mit solaren Neutrinos weisen auf einen großen Mischungswinkel hin. In Abb. (2.6) ist der Zusammenhang zwischen dem Mischungswinkel und der Neutrinomasse dargestellt es wird deutlich, daß für diesen solaren Winkel das beste Limit durch die Untersuchung des Tritium- β -Zerfalls geliefert wird. Durch das neue Experiment, KATRIN, ist eine Verschiebung der Obergrenze der Neutrinomasse um eine Größenordnung möglich.

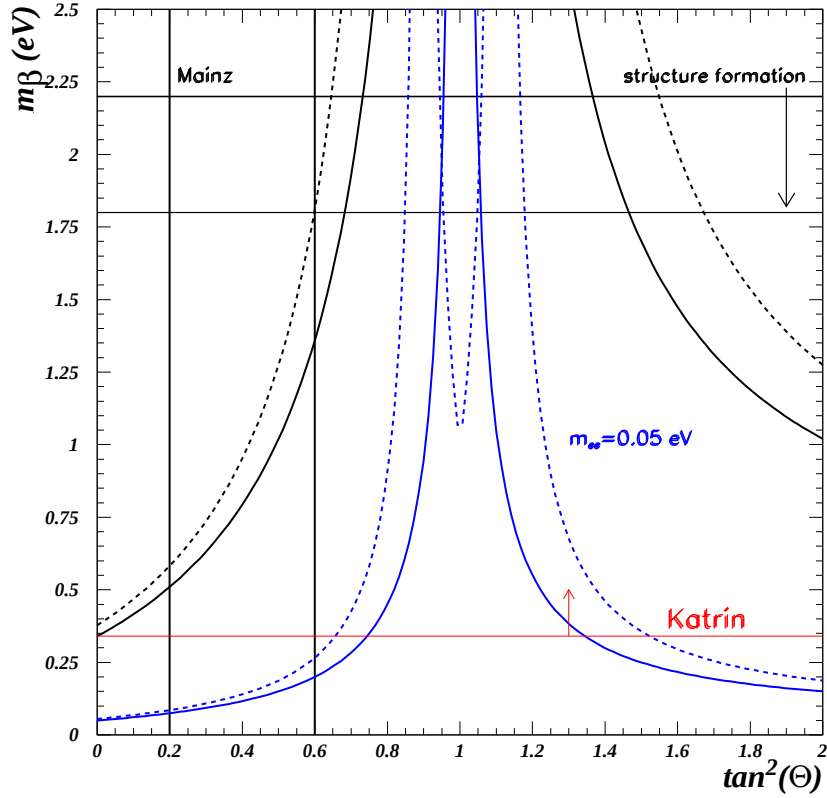


Abbildung 2.6: Unterschiede in der Massensensitivität in der direkten Untersuchung der Elektronneutrinomasse durch den Tritium- β -Zerfall, bzw. den neutrinolosen doppelten β -Zerfall. Die hyperbolischen Kurven geben die Sensitivitätsgrenze auf die Neutrinomasse in Abhängigkeit vom Tangens des Mischungswinkels an. Die oberen Kurven beschreiben den derzeitigen Erkenntnisstand, während die unteren beiden Kurven Simulationen des geplanten GENIUS-Experiments sind. Zusätzlich sind die derzeitigen Neutrinomassengrenzen, die in Mainz, bzw. durch die Strukturbildung ermittelt wurden, als horizontale Balken eingetragen. Man kann erkennen, daß dieses Limit durch KATRIN stark werden verbessert wird.

2.3.2 Die Komponenten von KATRIN

Die Sensitivität auf die Neutrinomasse hängt im wesentlichen von zwei Designparametern ab:

- Der Energieauflösung $\frac{\Delta E}{E}$, die durch das Verhältnis $\frac{B_{min}}{B_{max}}$ des Magnetfeldes B_{min} in der Analysierebene und B_{max} des maximalen Magnetfeldes zwischen Quelle und Analysierebene gegeben ist. Um eine Sensitivität auf die Elektronen-neutrinomasse im sub-eV Bereich zu erlangen sollte die Energieauflösung bei einer Endpunktsenergie von 18.6 keV $\Delta E = 1\text{eV}$ betragen. Daher ergibt sich ein Magnetfeldverhältnis $\frac{B_{min}}{B_{max}} = 5 \cdot 10^{-5}$ [Kat01].
- Der Luminosität $\mathcal{L}$ der β -Quelle, die proportional der Quellfläche A_S und dem maximal erlaubten Startwinkel θ_{max} ist. Da der Einfluß einer endlichen Neutrinomasse auf die Zählrate mit m_ν^2 skaliert, sollte die Luminosität um annähernd zwei Größenordnungen gegenüber den Quellen in Mainz und Troitsk erhöht werden.

Die Güte eines MAC-E-Filters nimmt mit der Fläche seiner Analysierebene zu, daher wächst sie quadratisch mit der Erhöhung der linearen Dimensionen. Um den erwarteten Sensitivitätsbereich zu erreichen, ist ein Spektrometer mit 7m Durchmesser und eine Quelle mit einem akzeptierten Startwinkel von etwa $\theta_{max} = 51^\circ$ geplant.

Eine weitere wichtige Voraussetzung ist das Erreichen einer geringen Untergrundzählrate, wodurch sich Anforderungen an das Vakuumsystem und den Detektor ergeben.

Schließlich sind zwei komplementäre Quellsysteme, eine Gasquelle (wie in Troitsk) und eine Festkörperquelle (wie im Mainz), geplant. Der Vorteil des Betriebes mit zwei Elektronenquellen liegt in der unterschiedlichen Systematik der beiden Quellsysteme, die ein Auffinden experimenteller Fehler erleichtert.

In Abb. (2.7) ist ein schematischer Aufbau des Experiments dargestellt. Im folgenden wird auf die einzelnen Komponenten eingegangen.

- Gasquelle: Die fensterlose Gasquelle besteht aus einem 10 m langen Rohr, in das durch eine dünne Kapillare hochreines Tritiumgas einströmt und an den Enden abgepumpt wird. Diese Kapillare befindet sich in der Mitte des Quellrohrs, so daß die Gaskonzentration zu den Rohrenden hin linear abnimmt. Das Quellrohr befindet sich in einer Reihe von Solenoiden, die die Zerfallselektronen adiabatisch zu den Rohrenden führen. Die Magnetfelder sind so gewählt, daß der bereits erwähnte Startwinkel von $\theta_{max} = 51^\circ$ erreicht wird.
- Transportsystem: Durch das Transportsystem werden die β -Elektronen adiabatisch aus dem Quellbereich zum Spektrometer geführt. Jedes Transportelement hat eine Länge von etwa einem Meter und ist in einem Solenoiden untergebracht. Die unterschiedlichen Transportelemente sind gegeneinander gekippt.

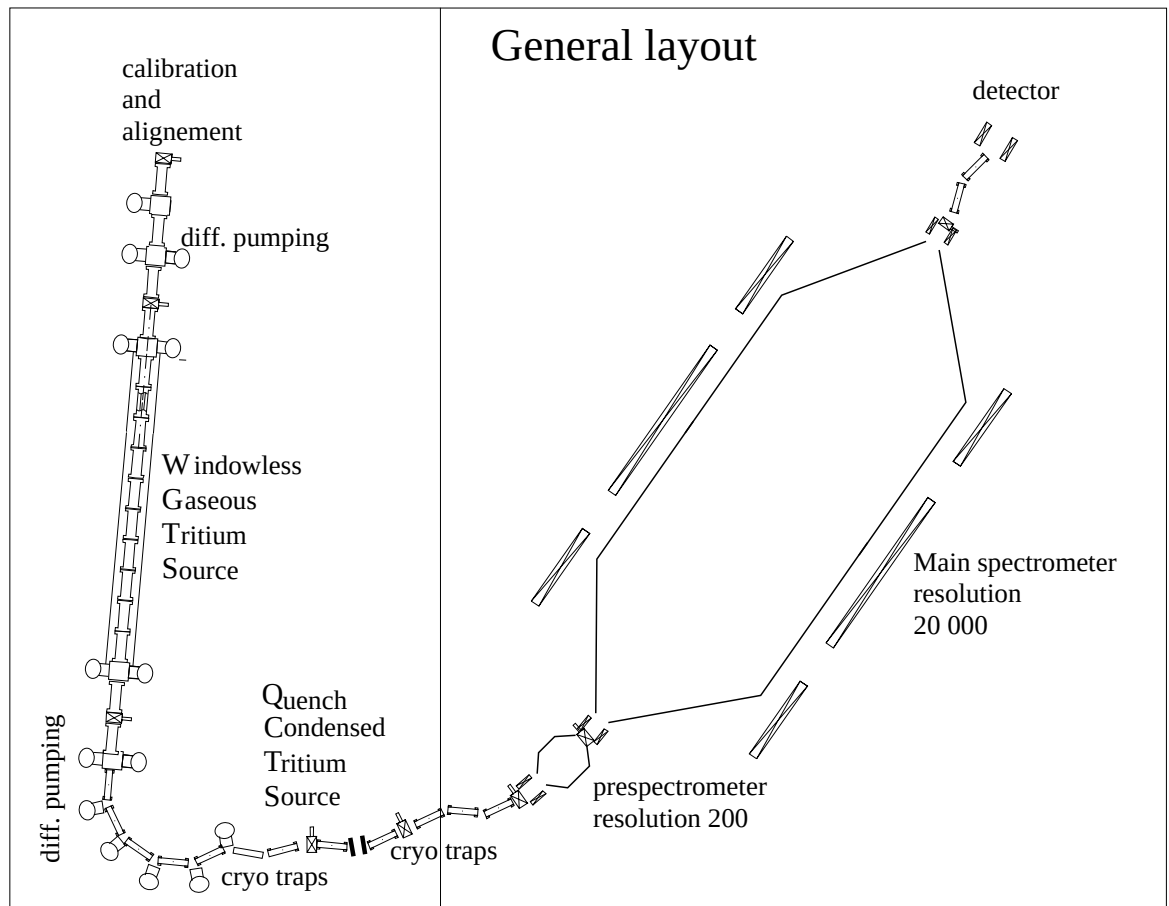


Abbildung 2.7: Layout von KATRIN Dargestellt ist ein schematischer Aufbau des Experiments, es sind alle im Text erklärten Komponenten zu erkennen. Das Layout wurde in dieser Form entworfen, daß der Aufbau in einer bereits vorhandenen Halle in FZK untergebracht werden kann.

Dies ist auch eine differentielle Pumpstrecke in der jeweils zwischen zwei Transportelementen eine Turbomolekularpumpe angebracht ist. Weiteres differentielles Pumpen wird durch heliumkalte Kryoflächen erreicht. Diese differentielle Pumpstrecke sorgt dafür, daß das Spektrometer frei von Tritium bleibt, was sowohl aus sicherheitstechnischen Gründen, als auch zur Reduktion des Untergrundes notwendig ist.

- **Festkörperquelle:** Die Festkörperquelle, die aus einem schockkondensierten Tritiumfilm besteht und einen Durchmesser von etwa 9 cm bei einer Dicke von ungefähr 100 Monolagen Tritium haben wird, erlaubt Messungen des Tritiumspektrums mit einer zu der Gasquelle komplementären Systematik.
- **Pinchmagnet:** Der Pinchmagnet definiert das stärkste Magnetfeld der Apparatur, das hier erzeugte Feld ist etwa doppelt so groß, wie das der Spektrometersolenoides. Auf diese Weise wird der maximale Startwinkel der Elektronen (s. Gl. 2.10), sowie die Energieauflösung (s. Gl. 2.8) des Experiments bestimmt.

- Vorspektrometer: Das Vorspektrometer dient als erster Energiefilter. Es handelt sich hierbei um ein Spektrometer der Größe, wie sie derzeit in Mainz und Troitsk betrieben werden. Die Retardierungsspannung wird etwa bei 18 kV liegen um β -Elektronen mit geringerer Energie schon vor dem Hauptspektrometer auszufiltern, und so zu verhindern, daß diese Elektronen durch Reaktionen mit dem Restgas oder den Spektrometerwänden Untergrundzählrate erzeugen. Die Energieauflösung soll im Bereich von $\frac{\Delta E}{E} \approx 200$ liegen.
- Hauptspektrometer: Bei dem Hauptspektrometer handelt es sich um die Schlüsselkomponente des Experiments. Es wird einen Durchmesser von 7m und eine Länge von etwa 20 m haben. Die angestrebte Energieauflösung beträgt 2 eV. Diese ergibt sich aus dem Verhältnis $\frac{B_{min}}{B_{max}}$ und der untersuchten Elektronenenergie von etwa 20 keV. Um Untergrundprozesse durch Stöße mit Restgas zu vermeiden, wird ein Vakuum angestrebt, das einem Restgasdruck von $p = 10^{-11}$ mbar entspricht.
- Detektor: An den Detektor herrschen hohe Anforderungen bezüglich der Effizienz des Elektronennachweises bei gleichzeitig geringem γ -Untergrund. Er sollte eine Energieauflösung $<5\%$ für 18.6 keV Elektronen haben und über eine gute Ortsauflösung verfügen, die es sowohl erlaubt das Quellprofil abzubilden, als auch eine Unterdrückung von Untergrundereignissen, die auf der Elektrodenoberfläche erzeugt wurden, erlaubt. Für die Flugzeitmessungen ist auch eine gute Zeitauflösung von Nöten.

Der von der Gesamtapparatur erzeugte magnetische Flußschlauch ist in Abb. (2.8) schematisch dargestellt. Es sind die Magnetfelder an den Quellen, das maximale Feld an der Stelle des Pinchmagneten und das minimale Feld in der Analysierebene dargestellt.

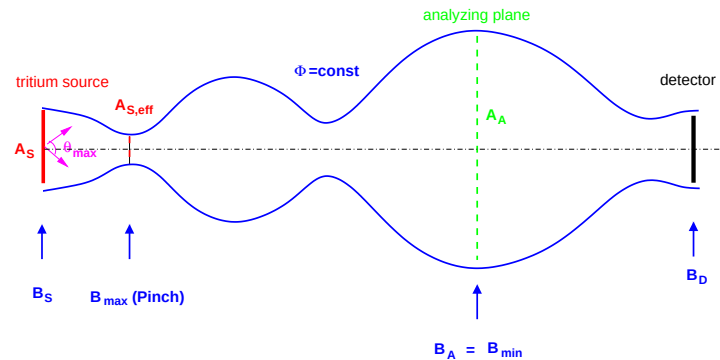


Abbildung 2.8: Der magnetische Flußschlauch ist in dieser Abbildung schematisch dargestellt. Das stärkste Magnetfeld herrscht im Pinchmagneten (B_{Pinch}), das schwächste in der Analysierebene des Hauptspektrometers. Das Verhältniss der Magnetfelder zwischen diesen Punkten beträgt $1/20000$.

Für eine genauere Beschreibung der einzelnen Komponenten sei auf [Kat01] verwiesen.

Kapitel 3

Anforderungen und Werkzeuge

Das Ziel dieser Arbeit ist es Werkzeuge und Methoden zur Simulation und Bewertung der Komponenten von KATRIN zu entwickeln und anzuwenden. In diesem Kapitel werden die theoretischen Grundlagen der verschiedenen Experimentkomponenten beschrieben, sowie die Anforderungen an die Designs gestellt. Der zweite Teil des Kapitels beschäftigt sich mit den Werkzeugen, mit deren Hilfe die Designs realisiert und simuliert wurden. Abschließend werden die durch die Simulationen untersuchten Größen erläutert. In den folgenden Kapiteln werden die Simulationen zum Transportsystem und zum Hauptspektrometer erörtert.

3.1 Designanforderungen

3.1.1 Das Spektrometerdesign

In Kap. 2 wurde bereits das Prinzip des MAC-E-Filters beschrieben. Als wesentliche Bestandteile wurden die adiabatische Führung und die Transformation im Magnetfeld sowie die Retardierung durch das elektrostatische Gegenpotential genannt. Es entstehen also Anforderungen an das Magnetdesign und an das Elektrodendesign. Das Elektrodensystem wird in einem Ultrahochvakuumtank untergebracht. Dieser Tank sollte einen Restgasdruck von $p < 10^{-11}$ mbar ermöglichen und bis etwa 400°C ausheizbar sein. Durch diesen geringen Druck wird die Wahrscheinlichkeit der Restgasionisation durch Elektronen reduziert, was zu einer niedrigen Untergrundzählrate beiträgt. Bei den angestrebten Dimensionen von einer Länge von 20m und einem Durchmesser von 7m handelt es sich bei der Konstruktion um eine große technische Herausforderung.

3.1.1.1 Magnetdesign

Grundsätzlich benötigt man zur Erzeugung des geforderten Flußschlauchs von

$$\int B(r) 2\pi r dr = \Phi = \pi r_A^2 \cdot B_A = (3.5 \text{ m})^2 \cdot \pi \cdot f \cdot 5 \text{ G} = 192 \cdot f \cdot \text{Tcm}^2 \quad (3.1)$$

zwei Spektrometersolenoiden, die in einem Abstand L voneinander entfernt sind. f ist dabei ein noch nicht festgelegter Skalierungsfaktor, der im Bereich von $0.6 \leq f \leq 1$ liegt. Durch diesen Faktor wird das Magnetfeld in den Spektrometersolenoiden zwischen 5 und 3 T festgelegt. Die Stelle des minimalen Magnetfeldes zwischen diesen Solenoiden definiert die magnetische Analysierebene B_A . Um eine konstante Analyse über den gesamten Querschnitt dieser Ebene treffen zu können, sollte diese Ebene eine hohe Homogenität aufweisen. Das Ziel ist es die Abweichungen der Magnetfeldstärke in der Analysierebene kleiner als 10% zu halten. Da dies nicht durch nur zwei Solenoiden erreicht werden kann, werden zur Formung des Flußschlauches zusätzliche Luftspulen angebracht. Dabei ist darauf zu achten, daß durch die Verwendung der Zusatzfelder keine lokalen Magnetfeldminima auftreten, die zu der Speicherung von Teilchen führen können, was wiederum zu einer erhöhten Untergrundzählrate beitragen kann [Ulr00].

Ein weiteres Kriterium beim Entwurf der Spektrometersolenoiden entsteht durch die Anforderungen an den Vakuumtank an sich. Wie bereits erwähnt, soll in dem Tank ein Druck $p < 10^{-11}$ mbar herrschen. Daraus folgt, daß der Tank durch UHV-Ventile abschließbar sein muß. Der gesamte magnetische Flußschlauch sollte also durch ein solches Ventil geführt werden können. Der maximal erhältliche Durchmesser ist DN 200 CF.

3.1.1.2 Elektrostatisches Design

In Kapitel 2 wurde bereits die Aufgabe des elektrischen Gegenpotentials beschrieben:

- Gewährleistung der adiabatischen Bewegung der Elektronen
- Analyse der Parallelenergie der Elektronen

Eine Voraussetzung für die adiabatische Bewegung der Elektronen ist eine geringe Änderung der Magnetfeldes $\frac{\Delta B}{B}$ während eines Zyklotronumlaufes. Da diese Änderung proportional zur Parallelkomponente der Elektronengeschwindigkeit ist, ist eine möglichst frühe Retardierung der Elektronen anzustreben.

Damit eine genaue Analyse der Parallelenergie der Elektronen möglich ist, muß die elektrostatische Analyse in dem Bereich stattfinden, in dem fast die gesamte Energie der Elektronen in Parallelenergie vorliegt, also in der magnetischen Analysierebene B_A . Die elektrische und die magnetische Analysierebenen müssen also identisch sein. Dies wird am leichtesten durch ein symmetrisches Elektrodendesign realisiert.

Ebenso wie für die magnetische Analysierebene wird auch für die elektrische Analysierebene eine möglichst hohe Homogenität angestrebt, die sich durch einen geringen Potentialdurchgriff äußert. Bei dem Potentialdurchgriff handelt es sich um die radiale Potentialdifferenz. Er verursacht eine Verbreiterung der Transmissionsfunktion, die durch einen radial auflösenden Detektor kompensiert werden muß. Der erwünschte

Wert des maximalen Potentialdurchgriffs liegt unterhalb von 1V.

Auch beim Entwurf der Elektroden muß auf die Vermeidung von möglichen Teilchenspeichern geachtet werden. Das betrifft im wesentlichen die Formgebung, die im ungünstigen Fall zur Bildung lokaler Feldminima führen kann.

Im allgemeinen werden geladene Teilchen auf Magnetfeldlinien geführt, so daß sich Teilchenfallen nur durch die axialen Speicherbedingungen unterscheiden.

- Elektrische Fallen: Diese Art der Teilchenspeicherung beruht auf der Speicherung in einem elektrischen Potentialminimum, so daß nur positiv geladene Teilchen (Ionen) gespeichert werden können. Diese Situation herrscht in der Spektrometernähe vor. Ein Spezialfall dieser Art der Speicherung tritt in den Ecken des Spektrometers auf, wo magnetische Feldlinien senkrecht über Sattelflächen des Potentials verlaufen können. D.h. daß entlang der Feldlinie Minima im elektrischen Potential auftreten, so daß Elektronen magnetisch geführt werden, aber durch den Potentialanstieg elektrisch reflektiert werden.
- Magnetische Fallen: Im Fall der magnetischen Speicherung wird die Reflexion durch ein hohes Magnetfeld bewirkt. Auf diese Weise können sowohl positiv, als auch negativ geladene Teilchen gespeichert werden. Entsteht ein Teilchen an einem Ort, an dem ein Magnetfeld der Größe B_S herrscht, mit einer bestimmten Energie, so wird die Parallelenergie dieses Teilchens in Transversalenergie umgewandelt, wenn sich das Teilchen in Richtung eines höheren Magnetfeldes ($B > B_S$) bewegt. Dieser Vorgang führt zur Reflexion des Teilchens. Diese Speicherung kann zwischen den Spektrometersolenoiden auftreten.
- Elektrisch / magnetische Fallen: Treten bei einer Teilchenfalle sowohl elektrische, als auch magnetische Speicherung auf, so wird von elektrisch-magnetischen Speicherung gesprochen. Speicherung dieser Art kann durch die Reflexion eines Teilchens am Retardierungspotential, auf der einen Seite, und Reflexion am magnetischen Gradienten, auf der anderen Seite erfolgen.

3.1.2 Das Transportsystem

Die Aufgabe des Transportsystems ist der adiabatische Transport der Elektronen von den Quellen zum Spektrometer. Um kein Elektron zu 'verlieren' muß auch hier der in Gl. (3.1) geforderte gesamte magnetische Flußschlauch von 192 Tcm^2 transportiert werden können. Das maximale Magnetfeld sollte im Bereich von 5 liegen, das minimale Magnetfeld in dem Spalt zwischen den einzelnen Transportelementen, laut Vorschlag der Troitsker Gruppe, nicht unter 0.5 T, da diese Werte den Parametern des Aufbaus in Troitsk entsprechen. Diese Angaben werden im Rahmen dieser Arbeit näher untersucht. Die einzelnen Transportelemente haben eine Länge von 1m und sind um 20° gegeneinander gekippt, um eine direkte Durchsicht von der Quelle zum Spektrometer zu verhindern. Auf diese Weise wird verhindert, daß eventuell von der Quelle kommende Tritiummoleküle auf geradem Weg in den Vakuumentank gelangen.

3.2 Werkzeuge

Ein wesentlicher Punkt dieser Arbeit lag in der Entwicklung von Werkzeugen, die es erlauben, die Magnetfelder von gegeneinander gekippten Solenoiden zu berechnen. Das Hauptproblem, das dabei auftrat, war die Brechung der Zylindersymmetrie des Systems, die von den Vorgängerprogrammen wie `bfield11` [Fic00] noch ausgenutzt werden konnte. Die Notwendigkeit zur Lösung dieser Problemstellung entsteht aus den Anforderungen an das Transportsystem. Ebenso wird in Erwägung gezogen, das Vor- und das Hauptspektrometer nicht linear, sondern gegeneinander gekippt zu installieren, um die Speicherzeit möglicherweise gespeicherter Teilchen aufgrund der $\vec{R} \times \vec{B}$ -Drift (s. Gl. 4.1) zu reduzieren.

3.2.1 `Bfield_3d`

Dieses Programm wurde zur Berechnung von Magnetfeldern gegeneinander gekippter Spulen geschrieben. Bei Spulenkonfigurationen dieser Art ist die Symmetrie um die Längsachse des Systems (im folgenden die x-Achse) gebrochen. Die Berechnung des Feldes kann also nicht mehr in Zylinderkoordinaten durchgeführt werden, auf die sich die bisher verwendeten Programme beschränken. Ein weiterer Schwachpunkt der bisher verwendeten Programme war der große Bedarf an Rechenzeit. So wurde z.B. das für SIMION (s. Kap. 3.2.2) benötigte magnetische skalare Potential, das sich in wirbelfreien Feldern außerhalb von Stromleitern berechnen läßt, durch die Integration des Magnetfeldes gewonnen, das seinerseits aus der Differentiation des Vektorpotentials berechnet wurde [Fic00]. Das Ziel war also die Entwicklung eines Programms, das Magnetfelder und magnetische Potentiale bei gegebener Spulenkonfiguration in kartesischen Koordinaten mit möglichst geringem Zeitaufwand berechnet. Hierzu wurden neue Reihenentwicklungen für das Magnetfeld und das skalare magnetische Potential langer dünner Spulen, bzw. Leiterschleifen, von F. Glück implementiert [Glu00].

Ein weiteres Designkriterium des Programmes war die Wiederverwendbarkeit, und damit die Modularität der einzelnen Unterprogramme.

Um einen strukturierten Programmaufbau zu gewährleisten, wurden die einzelnen Untereinheiten des Programmes in verschiedene Module gekapselt. Die Struktur des Programmes ist in Abb. (3.1) dargestellt.

Das Hauptprogramm befindet sich im Modul **`Bfield_3d`**, das die Menüführung und die Programmsteuerung beinhaltet. Von hier können die einzelnen Programmfunktionen wie

- Ausgabe der Spulenkonfiguration
- Berechnen von Magnetfeldern
- Berechnen von magnetischen Potentialen (zur Verwendung in SIMION)

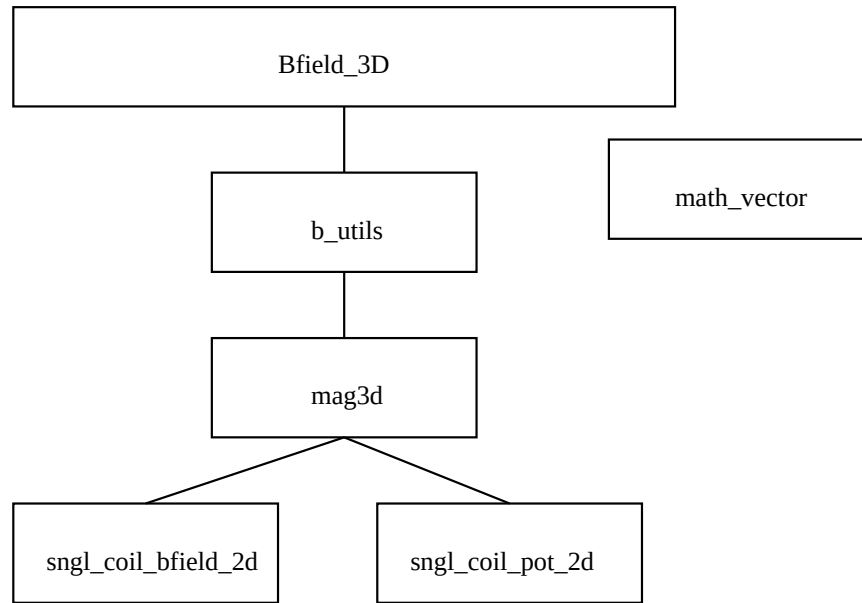


Abbildung 3.1: Struktur von Bfield3D. Es ist die Hierarchie der Module von Bfield3D dargestellt, die Zugriffe sind durch die Linien gekennzeichnet. Eine Ausnahme bildet `math_vector`, das allen Modulen zur Verfügung steht.

- Berechnung der Magnetfeldlinien
- Erzeugung eines kumac-files zur Darstellung der Spulenkonfiguration durch PAW
- Beenden des Programms

aufgerufen werden.

Die Basis des Programmes bilden die Routinen: `sngl_coil_bfield_2d` und `sngl_coil_pot_2d`. Sie beruhen weiterhin auf einer Zylindergeometrie und berechnen das Magnetfeld bzw. das magnetische skalare Potential einer einzelnen Spule. Diese Routinen werden von der Routine `mag_3d` aufgerufen, die im wesentlichen eine Rotation durchführt und die einzelnen Feld- bzw. Potentialkarten zu einer Karte des Gesamtsystems zusammensetzt. In diesem Schritt findet auch die Transformation von Zylinderkoordinaten auf kartesische Koordinaten statt.

`b_util` stellt die Ein- und Ausgabefunktionen, sowie die Routine zur Feldlinienberechnung zur Verfügung, während in `math_vector` häufig benutzte Vektoroperationen verkapselt sind.

Der Aufruf des Programmes erfolgt über die Linux-Kommandozeile:

bfield_3d *Parameterfilename*

Der Parameterfile enthält die Konfigurationsdaten der zu berechnenden Spulengeometrie. Ein Beispiel eines solchen Files ist im Anhang zu finden.

3.2.1.1 snagl_coil_bfield_2d

Die Berechnung des Magnetfeldes beruht wie bei bfield_11 auf der Zerlegung einer Spule in einzelne Leiterschleifen, deren Magnetfeld aus dem Vektorpotential durch das Einsetzen elliptischer Integrale bestimmt werden kann [Fic00]. Bei dieser Methode treten jedoch im Bereich nahe der Symmetrieachse numerische Ungenauigkeiten auf. Bei der bisherigen Anwendung war das kein Problem, da man in der Zylindergeometrie das Feld auf der Achse bestimmen konnte und der Abstand zum nächsten Gitterpunkt groß genug war. Um diese Ungenauigkeiten zu vermeiden, wurde für den Bereich nahe der Achse eine neue Reihenentwicklung zur Bestimmung des Magnetfeldes eingebaut [Glu00].

3.2.1.2 snagl_coil_pot_2d

Da SIMION keine extern erzeugten Magnetfelder benutzt, sondern diese aus dem magnetischen Potential berechnet, wurde eine neue Routine zur Berechnung des skalaren magnetischen Potentials erstellt. Das skalare magnetische Potential läßt sich in wirbelfreien Feldern außerhalb von Stromleitern berechnen. Dazu wird die zu berechnende Spule nicht in einzelne Leiterschleifen zerlegt, sondern durch eine Anzahl dünner langer Spulen approximiert [Glu00]. Auf diese Weise kann der Aufwand an Rechenzeit stark reduziert werden, da lange Spulen durch eine hohe Anzahl von Leiterschleifen in zwei Dimensionen diskretisiert werden müßten, um eine ausreichende Genauigkeit des Magnetfeldes zu erlangen. Die Zerlegung in dünne Spulen verlangt dagegen nur eine eindimensionale Diskretisierung und ist entsprechend schneller.

Das magnetische Potential $U(z, r)$ einer dünnen langen Spule läßt sich durch eine radiale Reihenentwicklung bestimmen. Dabei ist darauf zu achten, daß diese Reihe nur in einem bestimmten Konvergenzradius R_C

$$R_C = \begin{cases} R & \text{für } z_{min} \leq z \leq z_{max} \\ \sqrt{R^2 + (z - z_{max})^2} & \text{für } z > z_{max} \\ \sqrt{R^2 + (z - z_{min})^2} & \text{für } z < z_{min} \end{cases} \quad (3.2)$$

konvergiert (Erläuterung der Größen: s. Abb.(3.2)).

Es gilt für den Fall $r \neq 0$:

$$U(z, r) = \int_0^r d\varrho B_r(z, \varrho) = -\frac{1}{4} R_C^2 \sum_{n=0}^{\infty} S_n(z) \frac{1}{n+1} \left(\frac{r}{R_C}\right)^{2n+2} \quad (3.3)$$

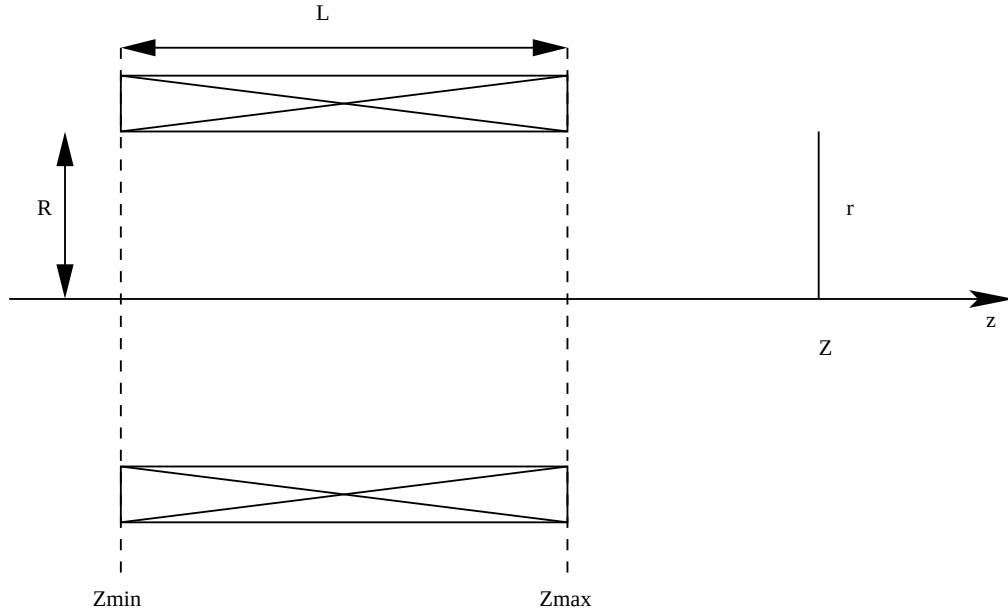


Abbildung 3.2: Konvergenzkriterien für das Vektorpotential einer dünnen Spule. Zur Erläuterung der Größen in Gl. (3.2) sind hier die Variablen einer Spule beschrieben. R ist der Spulenradius, L die Spulenlänge, die sich von Z_{min} bis Z_{max} erstreckt. (z, r) ist der Ort, an dem das Potential, bzw. das Magnetfeld bestimmt werden soll. Die Spulenachse ist hier noch mit z benannt, da das magnetische Potential noch in Zylinderkoordinaten berechnet wird, und erst in einer folgenden Routine in kartesische Koordinaten transformiert wird.

Für diese Entwicklung wurden zwei Abbruchparameter implementiert. Zunächst wird überprüft, ob die Differenz zwischen zwei Reihenelementen einen gewissen Konvergenzwert unterschreitet, sollte dies nicht der Fall sein, wird die Reihe nach einer Anzahl n_{max} von Elementen abgebrochen.

Die genaue Berechnung der Summenelemente S_n wird hier nicht dargestellt, da eine Veröffentlichung durch F. Glück in Vorbereitung ist. Es sei auf den im Anhang wiedergegebenen Quelltext verwiesen.

Das gesamte magnetische Potential berechnet sich aus der Summe

$$V(z, r) = V(z, 0) + U(z, r) \quad (3.4)$$

mit:

$$V(z, 0) = \frac{1}{2}\mu_0 I \frac{G_0(z - Z_{min}) - G_0(z - Z_{max})}{L} - \frac{1}{2}\mu_0 I \quad (3.5)$$

wobei:

$$G_0(x) = \sqrt{R^2 + x^2} \quad (3.6)$$

3.2.1.3 Berechnung der Magnetfeldlinien

Die Berechnung der Magnetfeldlinien wird im Modul `b_util` durchgeführt. Dem Programm wird eine Koordinate übergeben, von der aus die Feldlinie bestimmt werden soll. Der Algorithmus berechnet nun den Magnetfeldvektor an dieser Stelle, dies geschieht unter Verwendung des Moduls `Bfield_3D`. Die Koordinaten des Punktes und der Betrag des Magnetfeldes werden in den Speicher geschrieben. Es folgt der nächste Schritt in Richtung des Magnetfeldvektors, wobei die Schrittweite ebenfalls dem Programm übergeben wird, wo sich diese Prozedur wiederholt. Dieser Vorgang wiederholt sich bis der Rand des zu berechnenden Gebietes erreicht ist. Da die Feldlinien symmetrisch zu der ersten Koordinate berechnet werden, springt das Programm zu dieser zurück und beginnt dieselbe Berechnung in die andere Richtung.

3.2.1.4 Tests von `Bfield_3D`

Zum Test der neuen Routinen wurde das Magnetfeld einer zylindersymmetrischen Konfiguration (Abb. (3.3)) mit Hilfe des bisher verwendeten Programmes `bfield_11` und des neuen Programmes `Bfield_3D` berechnet. Die Ergebnisse sind in Abb. (3.4) dargestellt. Es ist zu erkennen, daß innerhalb der Spule Abweichungen von etwa $2 \cdot 10^{-4}$ auftreten, die dann aber zum Spulenrand hin stark zunehmen. Mit wachsendem Abstand von der Spule gehen die Differenzen der Magnetfeldwerte gegen Null.

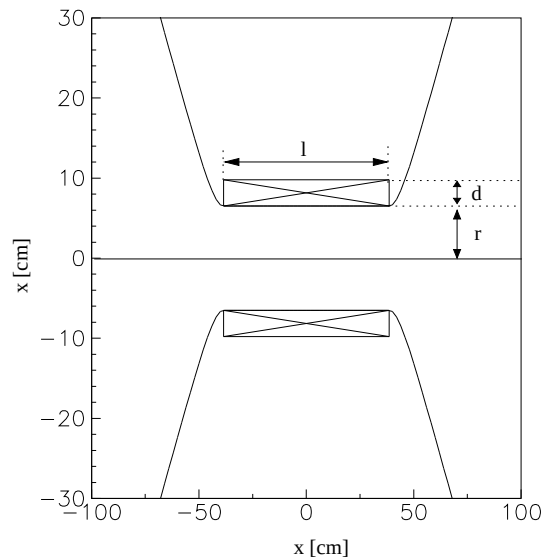


Abbildung 3.3: Einfache Spulenkonfiguration zum Test von `Bfield3D`. Es handelt sich hierbei um den Mittelsolenoiden des Transportelements (s. Tab (4.1) #2). Zusätzlich ist hier der Konvergenzradius R_c der Reihenentwicklung eingezeichnet.

Weiterhin zu Testen war die Genauigkeit des magnetischen Potentials, das durch die Reihenentwicklung berechnet wird. Dazu wurde eine Differentiation des Potentials durchgeführt und das resultierende Magnetfeld mit dem berechneten Magnetfeld

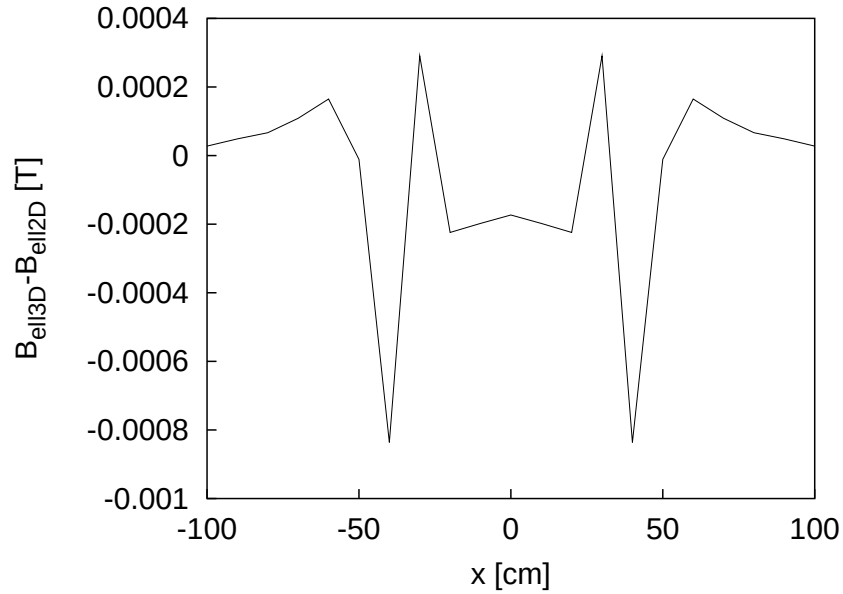


Abbildung 3.4: Vergleich der Magnetfeldberechnung von `bfield_11` und `Bfield3D` Es ist die Differenz des durch `bfield_11` und `Bfield3D` berechneten Magnetfeldes gegen die x-Koordinate dargestellt.

verglichen (s. Abb. (3.5)). Es ist zu erkennen, daß entlang der x-Achse Extrema im Bereich von 10^{-3} T auftreten. Vergleicht man die Position dieser Extrema mit Abb. (3.3), so sieht man, daß diese starken Abweichungen am Ende der Spule auftreten. Dies ist durch die abschnittsweise Definition des Konvergenzradius R_c (Gl. (3.2) zu erklären: Die Definition ändert sich genau am Spulenende, so daß zunächst eine Abweichung in negative Richtungen gefolgt von einer Abweichung in die positive Richtungen auftritt. Innerhalb der Spule und weit außerhalb der Spule sind die Differenzen der Magnetfeldwerte mit Null verträglich. Für die Berechnung entlang der x-Achse ergeben sich Abweichungen im Bereich von $1 \cdot 10^{-5}$. Da die Magnetfelder in unmittelbarer Nähe der Spule sehr stark sind (≈ 5 T) können die Abweichungen von 10^{-3} T in diesem Bereich vernachlässigt werden. Es ist also festzustellen, daß die Magnetfelder, die durch zwei unabhängige Methoden berechnet wurden, gut in Einklang miteinander stehen und so die Benutzung von `Bfield_3D` gerechtfertigt ist.

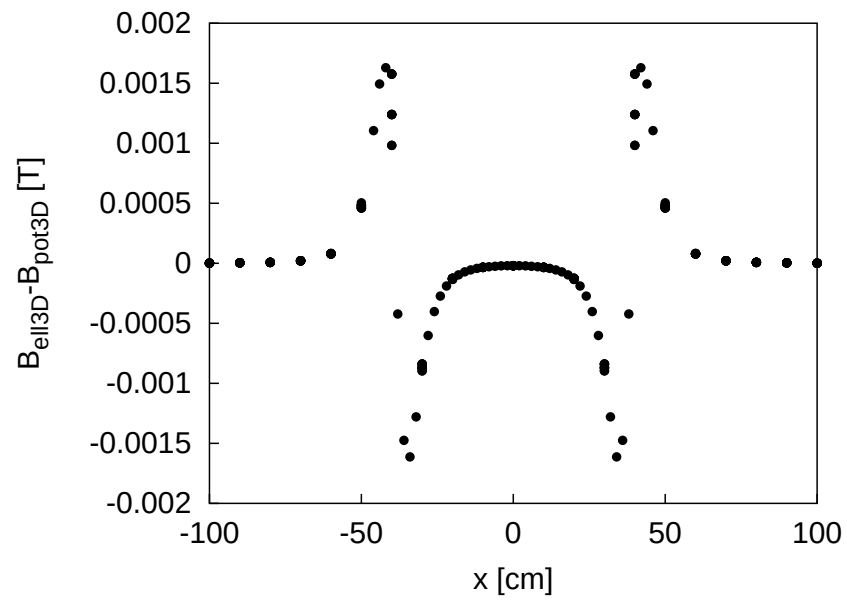


Abbildung 3.5: Test der Berechnung des magnetischen Potentials Dargestellt ist die Differenz der durch Differentiation des magnetischen Potentials und durch die Lösung der elliptischen Integrale berechneten Magnetfelder entlang der x-Achse.

3.2.2 SIMION 7.0

SIMION 3D [SIM] ist ein kommerzielles Programm, das in der verwendeten Version 7.0 unter dem Betriebssystem Windows NT 4.0 läuft. Es wurde zur Berechnung der elektrostatischen Potentiale und zur Bahnverfolgung von geladenen relativistischen Teilchen in elektromagnetischen Potentialen eingesetzt.

- Die Berechnung der Potentialkarte erfolgt, nach der Eingabe einer Elektroden-geometrie über einen graphischen Editor, durch die Relaxations-Methode. Die eingegebene Geometrie definiert ein Potential an der Stelle der Elektroden, so daß das Potential an jedem Punkt im Raum bestimmt werden kann. Dazu bedient sich SIMION einer Iteration, die abbricht, wenn die Differenz der Potentialwerte benachbarter Punkte einen vom Benutzer anzugebenden Konvergenzwert unterschreitet (s. Abb (3.6)).

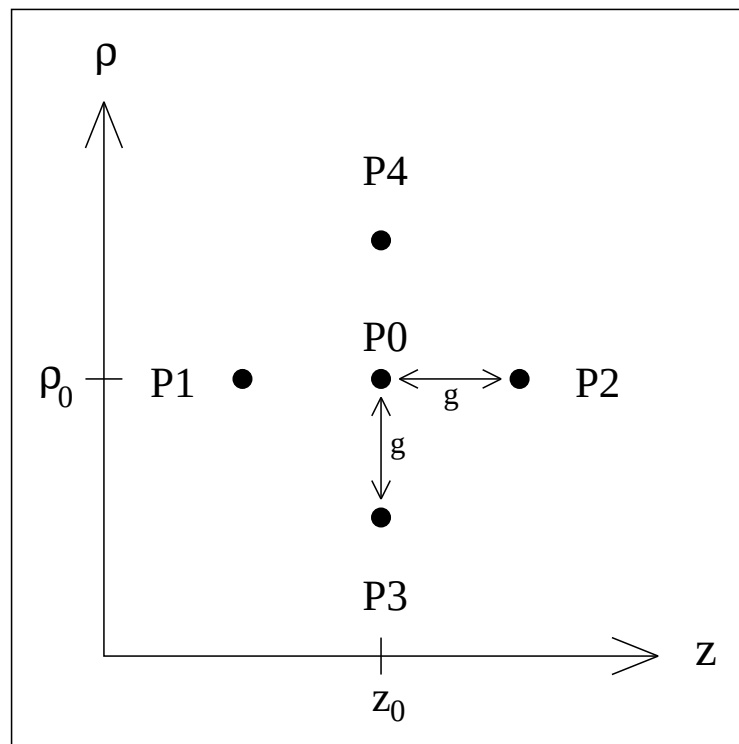


Abbildung 3.6: Schema der Relaxation in SIMION Das Potential des Punktes P0 wird solange iterativ berechnet, bis die Differenz zu den Potentialwerten der Nachbarpunkte (P1 - P4) einen Konvergenzwert unterschreitet. g ist die Gitterkonstante.

Die Relaxationsformel hängt nun von der Gitterkonstanten g ab, mit der das Potential bestimmt werden soll.

$$P_0 = \frac{1}{4}(P_1 + P_2 + P_3 + P_4) + \frac{g}{8\epsilon_0}(P_3 - P_4) \quad (3.7)$$

Dieses Verfahren ist allerdings nur für die Berechnung in zwei Dimensionen und gegebener Zylindersymmetrie gültig. Da diese Voraussetzungen für das Spektrometer gegeben sind, und in den Transportelementen keine elektrischen Felder benutzt werden, ist diese Einschränkung für diese Untersuchungen nicht relevant.

- SIMION bedient sich bei der Berechnung des elektrischen Potentials der Additivität der Lösungen der Laplace-Gleichung. So wird das Potentialfeld jeder Elektrode einzeln berechnet, während die Potentiale der übrigen Elektroden auf Null gesetzt werden. Das Gesamtpotential ergibt sich aus der Superposition der berechneten Einzelkarten. Auf diese Weise wird eine Änderung der Potentiale an einzelnen Elektroden vereinfacht.
- Zur Bahnverfolgung werden die elektrischen und magnetischen Kräfte getrennt voneinander berechnet. Die Bewegungsgleichung wird durch das Runge-Kutta-Verfahren 4. Ordnung numerisch integriert.

Da dieses Verfahren abhängig von der Schrittweite und somit von der Gitterkonstante g , mit der das Potentialfeld relaxiert wurde, ist, verfügt SIMION über eine dynamische Schrittweitensteuerung, die durch den Benutzer über den sog. 'computation quality factor' Q bestimmt werden kann. In [Fic00] ist die Wirkungsweise von Q beschrieben. Im Laufe der Simulationen dieser Arbeit wurde $Q=110$ benutzt, was sich in früheren Tests als geeigneter Kompromiss zwischen Genauigkeit und Rechenaufwand heraus gestellt hat (s. auch Abb. (4.2)).

3.3 Die untersuchten Größen

Das Ziel der Simulationen ist es, verschiedene Entwürfe miteinander zu vergleichen und ihre Eignung zur Verwendung im Experiment zu testen. Dies gilt sowohl für das Transportsystem, als auch für das Spektrometer. In beiden Fällen ist es wichtig die adiabatischen Eigenschaften der geplanten Komponenten zu testen. Desweiteren sind die Abbildungseigenschaften von Transportelementen und Spektrometer von Interesse. Ein weiterer Parameter, der zu testen war, war die Flugzeit der Teilchen im System. Außerdem darf eine Untersuchung der numerischen Genauigkeit nicht fehlen. Für das Transportsystem wurde auch der Phasenraum einer möglichen Teilchenfalle in einem der Transportelemente untersucht.

- **Untersuchung der adiabatischen Eigenschaften**

Die Bedingung eines adiabatischen Transports ist durch Gl.(2.6) gegeben. Es ist üblich einen Parameter Δ einzuführen, der ein Maß für die Verletzung der Adiabasie ist.

$$\Delta = \frac{\gamma_1 |\vec{\mu}_1|}{\gamma_2 |\vec{\mu}_2|} - 1 = \frac{p_{\perp 1}^2}{B_1} \frac{B_2}{p_{\perp 1}^2} - 1 \quad (3.8)$$

Herrschen vollkommen adiabatische Bedingungen, so ist nach Gl.(2.6) das Verhältnis von $\gamma_1 |\vec{\mu}_1|$ an einer Stelle der Elektronenbahn zu $\gamma_2 |\vec{\mu}_2|$ an einer anderen

Stelle der Bahn genau 1. Damit ist Δ im Fall des adiabatischen Transportes gleich Null. Etwaige Abweichungen lassen auf Fehler in der Transformation der Transversalenergie in Parallelenergie schließen. Eine äquivalente Größe zur Untersuchung der adiabatischen Eigenschaften ist die Differenz der Transversalkomponente der Energie am Start und am Ende des Elektronenfluges.

$$\Delta E_{\perp} = \gamma_1 \cdot E_{\perp 1} - \gamma_2 \cdot E_{\perp 2} \frac{B_1}{B_2} \quad (3.9)$$

Diese Größe gibt die Änderung der Transversalenergie zwischen Start- und Endpunkt der Flugbahn an.

- **Untersuchung der Abbildungseigenschaften**

Es wird die Differenz zwischen dem erwarteten Ort des Teilchens (ausgehend von einer Punkt-zu-Punkt-Abbildung) und dem tatsächlichen Ort des Teilchens ermittelt. Bei den Simulationen zu den Transportsystemen werden die Positionen in der Mitte der Solenoiden bestimmt, bei denen zum Spektrometer in der Analysierebene. Es wird erwartet, daß sich das Elektron innerhalb eines Kreises mit dem Radius $R = R_c$ um die erwartete Position aufhält. In eine Richtung wird eine weitere Verschiebung um einen Zyklotronradius erwartet, da das Elektron am Start diesen Abstand von der Führungsfeldlinie hat.

- **Simulationen der Flugzeit**

Diese Simulationen wurden zur Untersuchung des MAC-E-TOF Modus durchgeführt.

- **Untersuchung der numerischen Genauigkeit**

Um physikalische Aussagen treffen zu können, ist es wichtig zu wissen, wie die Simulationsergebnisse zu bewerten sind. Die numerische Genauigkeit ist dabei eine ausschlaggebende Größe. Sie wurde durch die Erhaltung der Gesamtenergie überprüft, so daß wir als Meßgröße die Verletzung der Gesamtenergie erhalten:

$$\delta E = (E + eU)_1 - (E + eU)_2 \quad (3.10)$$

Der erwartete Wert für δE beträgt aufgrund der Energieerhaltung Null.

- **Das Phasenraumvolumen einer Falle**

Bei der Untersuchung der Transportelemente stellt sich die Frage, was mit den Tritiummolekülen geschieht, die sich von der Quelle lösen. Die differentielle Pumpstrecke sorgt dafür, daß der Tritiumdruck kontinuierlich abnimmt, dennoch werden Tritiumzerfälle entlang der Transportstrecke auftreten. Es ist wichtig zu wissen, welchen Prozessen die dabei entstehenden Elektronen unterliegen. Ein Teil der Zerfallselektronen wird gespeichert, während ein anderer Teil zu Spektrometer gelangt, wo sie analysiert werden. Da die gespeicherten Elektronen jedoch Untergrundprozesse bedingen können, ist es von Interesse in welchen Phasenraumvolumen eine Speicherung möglich ist.

Dazu wird jedes Volumenelement ΔV des abgebildeten Flußschlauches gewichtet mit dem gespeicherten Raumwinkel $\Delta \Omega_{Trap}$ betrachtet. Ein Elektron, das unter einem Winkel θ_S aus einem Zerfallsvolumen der Magnetfeldstärke

B_S startet wird gespeichert, wenn auf beiden Seiten ein stärkeres Magnetfeld $B_{max} > B_S$ magnetische Spiegelung für $\theta_{max} < \theta_S < \pi - \theta_{max}$ bewirkt. θ_{max} ergibt sich dabei aus Gl. 2.10. Daraus ergibt sich

$$\Delta\Omega_{trap} = 2\pi \int_{\theta_{max}}^{\pi-\theta_{max}} \sin\theta d\theta = 2\pi [-\cos\theta]_{\theta_{max}}^{\pi-\theta_{max}} = 4\pi \cos\theta_{max} = \sqrt{1 - \left(\frac{B}{B_{max}}\right)^2} \quad (3.11)$$

Für das Phasenraumvolumen der Teilchenfalle gilt dann

$$V = \int \frac{\Delta\Omega_{trap}}{4\pi} dV \quad (3.12)$$

wobei

$$dV = A \cdot ds \sim \frac{ds}{B} \quad (3.13)$$

Dieser letzte Zusammenhang folgt aus $B \cdot A = \text{const.}$, wobei A die Grundfläche des betrachteten Volumenelements ist.

Die Ergebnisse der Untersuchungen der hier vorgestellten Größen werden in den Kapiteln 4 und 5 für das Transportsystem, bzw. für das Spektrometer erörtert.

Kapitel 4

Das Transportsystem

In diesem Kapitel werden die bis jetzt besprochenen Programme und Methoden angewendet und Simulationen zur Untersuchung verschiedener Transportsysteme durchgeführt. Es handelt sich dabei um einen Entwurf der Firma ACCEL aus Bergisch Gladbach, der im Auftrag des Forschungszentrums Karlsruhe entwickelt wurde und um einen Entwurf von Dr. Oleg Kazachenko aus der Arbeitsgruppe in Troitsk.

Das Ziel der Simulationen ist eine Bewertung des adiabatischen Transports der Elektronen durch das System, sowie eine Überprüfung des Abbildungsverhaltens des Systems. Für alle Simulationen in diesem Kapitel wurde der Qualitätsfaktor $Q = 110$ in SIMION eingestellt (s. Kapitel 3).

4.1 Entwurf der Firma ACCEL

Ein Transportelement besteht in diesem Entwurf aus drei Spulen, deren Eigenschaften aus Tabelle (4.1) zu entnehmen sind. Aufgrund eines Mißverständnisses wurde dieses Design, entgegen den Designspezifikationen, für einen Winkel von 10° zwischen den Transportelementen entworfen. Im Zentrum der Transportspule wird ein Magnetfeld von 5 T erzeugt. In der Lücke zwischen zwei Transportelementen herrscht ein Feld von etwa 0.7 T.

Das gesamte Transportsystem besteht aus etwa 10 dieser Transportelemente (s. Abb. (2.5)). Kritisch ist aber zunächst der Übergang zwischen den einzelnen Elementen, so daß hier nur dieser Übergang untersucht wird. Es ist davon auszugehen, daß ein Aneinanderreihen von mehreren Transportelementen zu einer Addition etwaiger Effekte führt, dies wird in dieser Arbeit jedoch nicht mehr betrachtet und bedarf weiterer Untersuchungen.

In Abb. (4.1) sind zwei aufeinander folgende Transportelemente dargestellt, die um den Winkel von 10° gegeneinander gekippt sind. Zunächst war zu überprüfen, ob diese Konfiguration mit den vorhandenen Werkzeugen berechnet werden kann. Dazu

Tabelle 4.1: ACCEL-Design eines Transportelements. Es sind die Parameter eines Transportelements des Entwurfes von ACCEL aufgelistet. x und y geben dabei die Koordinaten relativ zum Zentrum der Transportspule an. Durch θ wird der relative Kippwinkel der Spulen ausgedrückt, während r , d , l den Radius, die Dicke und die Länge der einzelnen Spulen beschreiben. nI gibt das Produkt der Windungszahl und des Spulenstroms wieder.

#	x [cm]	y [cm]	θ [°]	r [cm]	d [cm]	l [cm]	nI [A]
1	-45.1	0.0	0.0	8.5	5.68	9.78	666604.8
2	0.0	0.0	0.0	6.5	3.29	77.14	3045487.2
3	45.1	0.0	0.0	8.5	5.68	9.78	666604.8

wurde der Konvergenzradius der Reihenentwicklung von Bfield3D (s. Kap. 3) in Abb. (4.1) dargestellt. Es ist zu erkennen, daß der zu berechnende Flußschlauch an jeder Stelle des Systems innerhalb des Konvergenzradius liegt.

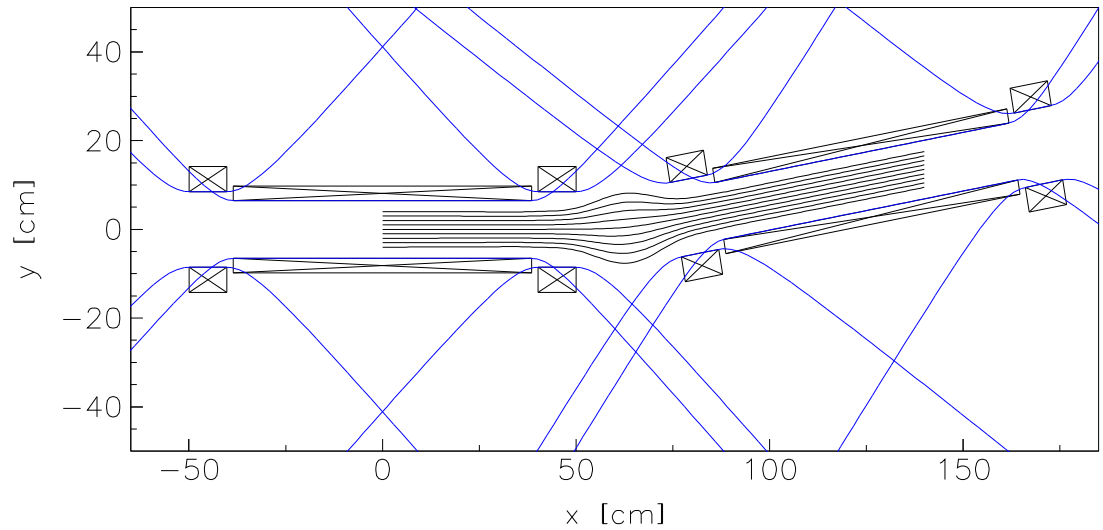


Abbildung 4.1: Das ACCEL-System Es sind zwei Transportelemente dargestellt, die um 10° gegeneinander gekippt sind. Desweiteren ist der berechnete Flußschlauch und die Konvergenzradien der Reihenentwicklung von Bfield3D für die einzelnen Spulen dargestellt. Die Einhüllende dieser Konvergenzradien umfaßt den berechneten Flußschlauch, so daß eine Simulation des Systems mit SIMION möglich ist.

Auf dieser Basis wurde das skalare magnetische Potential der Konfiguration mit Hilfe von Bfield3D berechnet und zur Verwendung durch SIMION bereit gestellt. Die einzelnen Spulen wurden jeweils durch fünf dünne Spulen approximiert, womit die in Kap. 3.2.1.3 beschriebene Genauigkeit erreicht wurde.

Durch die Simulationen mit SIMION wurden die numerische Genauigkeit, die Verlet-

zung der Adiabasie, und die Abbildungseigenschaften des Transportsystems untersucht. Die Berechnungen von SIMION wurden vollständig relativistisch durchgeführt, die Energie der Elektronen war in sämtlichen Simulationen 18601 eV, was einer Überschußenergie von 1 eV gegenüber dem Retardierungspotential in der Analysierebene des Spektrometers entspricht. Weiterhin haben alle Berechnungen gemein, daß die Elektronen in der Mitte des linken Transportelements gestartet wurden und in der Mitte des rechten Transportelementes gestoppt wurden. An beiden Punkten wurden die Koordinaten, die momentanen Geschwindigkeits- und Magnetfeldvektoren, die kinetische Energie, der Fehler der kinetischen Energie und die Flugzeit der Elektronen aufgenommen. Aus diesen Daten lassen sich die in Kap. 3.3 beschriebenen Größen zur Bewertung des Systems bestimmen.

Es wurden Simulationen für eine Vielzahl von Elektronen mit verschiedenen Startradien r und Startwinkeln θ bezüglich der x-Achse durchgeführt. Desweiteren wurde der Azimutwinkel φ variiert. Die Ergebnisse dieser Berechnungen sind in den Abbildungen (4.3 - 4.9) dargestellt.

Betrachtungen zur numerischen Genauigkeit

Wie bereits in Kap. 3.3 erwähnt wird zur Untersuchung der numerischen Genauigkeit die Energieerhaltung überprüft. Da ein Verlust oder ein Gewinn an Energie unphysikalisch ist, muß ein solcher Effekt mit numerischen Fehlern im Simulationsprogramm einhergehen. Die Untersuchung der Gesamtenergie liefert so eine Güte für die Berechnungen durch SIMION.

In den Abbildungen (4.3 - 4.9) (a) ist zu erkennen, daß die Verletzung der Energieerhaltung mit zunehmendem Startwinkel ansteigt. Dies ist insofern zu verstehen, daß Elektronen, die unter steilen Winkeln zur Achse starten einen höheren Anteil an Transversalenergie haben und daher eine geringere Zyklotronschrittweite haben, als Elektronen, die unter flachen Winkeln gestartet sind. Die höhere Anzahl an Zyklotronschritten bedeutet eine Zunahme der Flugzeit (in den Abb. (b) zu erkennen) und somit eine höhere Anzahl an Integrationsschritten, was zu einer Anhäufung der numerischen Fehler führt. Eine Bestätigung dieses Zusammenhanges liefern auch die Abb. (c), die die Verletzung der Energieerhaltung in Abhängigkeit der Flugzeit darstellen. Eine weitere Bestätigung des Zusammenhanges zwischen Flugzeit und numerischem Fehler wird in Abb. (4.2) deutlich. Der Bereich, in dem δE konstant ist entspricht der Zeitdauer, in der das Elektron das niedrige Magnetfeld zwischen den Transportelementen passiert. An dieser Stelle sinkt die Anzahl der Zyklotronschritte und somit die Anzahl der Integrationsschritte, was den geringeren Anstieg des Fehlers erklärt. Bei Eintritt in das zweite Transportelement und damit in ein hohes Magnetfeld steigt die Anzahl der Integrationsschritte wieder, was einen Anstieg des numerischen Fehlers bedingt.

Im Allgemeinen ist aus Abb. (4.3 - 4.9 (a)-(c)) zu erkennen, daß der Startradius keinen Einfluß auf die Verletzung der Energieerhaltung hat, die obere Grenze der

Energieverletzung liegt bei allen simulierten Radien (-32 mm, -16 mm, -8 mm, 0 mm, 8 mm, 16 mm, 32 mm) bei etwa 0.065 eV. Diese Werte wurden für Elektronen mit Startwinkeln von etwa 80° berechnet, von Interesse für das tatsächliche Experiment sind jedoch nur Elektronen mit Startwinkeln bis zu 51° (s. Kap 2.2). In diesem Winkelbereich beträgt die Verletzung der Energieerhaltung etwa 10 meV für alle simulierten Startradien.

Den Abbildungen (4.3 - 4.9) (a-c) ist ebenso zu entnehmen, daß der Phasenwinkel φ keinen Einfluß auf die Numerik hat.

Zieht man in Betracht, daß die untersuchten Elektronen eine Energie von 18.601 keV hatten ist eine Verletzung der Energie von 10 meV, also eine Verletzung der Größe 10^{-6} sehr gering. Auf dieser Grundlage konnten weitere Untersuchungen des Systems durchgeführt werden.

Betrachtungen zu den Abbildungseigenschaften

Die Untersuchung der Abbildungseigenschaften ist insofern von Interesse, als das sicher gestellt werden kann, daß alle Elektronen, die im magnetischen Flußschlauch geführt werden auch von der Quelle zum Spektrometer gelangen. Bestehen zu große Driften in den Transportelementen, so können Elektronen verloren werden.

Für den Fall daß keine Driften auftreten sollte eine Punkt zu Kreis Abbildung erfolgen, genauer erwartet man eine Abbildung der Elektronen innerhalb eines Kreises dessen Radius zwei Zyklotronradien der Elektronen entspricht. Die Ursache für diesen

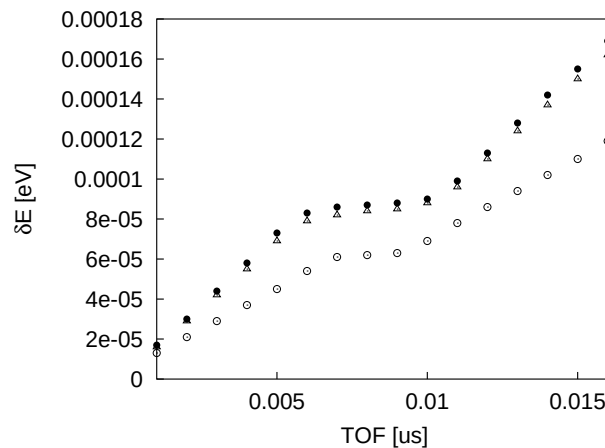


Abbildung 4.2: Der numerische Fehler entlang einer Elektronenbahn. Die einzelnen Punkte wurden in einem zeitlichen Abstand von $10^{-3} \mu s$ aufgenommen. Die Simulation wurde mit verschiedenen Q -Werten von 150 (Kreise), 110 (Dreiecke) und 101 (volle Kreise) durchgeführt. Es ist zu erkennen, daß der Unterschied zwischen $Q=150$ und 110 sehr gering ist.

Effekt liegt darin, daß die Elektronen von ihrem Startort von einer Magnetfeldlinie geführt werden, die einen Zyklotronradius von der eigentlichen Teilchenposition entfernt ist. Trifft das Elektron gerade zu dem Zeitpunkt am Endpunkt der Bahn ein, zu dem es sich auf der gegenüber liegenden Seite der Zyklotronbahn befindet, so ergibt sich der Radius des abgebildeten Kreises zu zwei Zyklotronradien.

In den Abb.(4.3 - 4.9) (d) ist die Differenz der tatsächlichen von der erwarteten Elektronenposition in Einheiten des Zyklotronradius aufgetragen. Es ist zu erkennen, daß sowohl in der y-, als auch in der z-Richtung leichte Abweichungen von der Erwartung auftreten. Es handelt sich dabei um Werte von etwa einem halben Zyklotronradius, dies entspricht $15\mu m$. Eine Überprüfung der berechneten Magnetfeldlinien ergab eine Abweichung von etwa $7\mu m$ von der Sollposition, diese Werte liegen innerhalb einer Größenordnung und können so diesen Effekt erklären.

Betrachtungen zur Adiabasierletzung

Zur Betrachtung der Adiabasie stehen zwei Größen zur Verfügung: der Δ -Parameter und die Änderung der Transversalenergie während eins Durchfluges durch das Transportelement $\Delta E_{\perp}$. Sie sind jeweils in den Abb. (4.3 - 4.9) (e) und (f) dargestellt.

Der Verlauf des Δ -Parameters zeigt für alle Startradien eine betragsmäßige Überhöhung für kleine Startwinkel ($\theta < 30^\circ$) und im weiteren Verlauf ein etwa konstantes Verhalten. Dieses Verhalten ist darauf zurückzuführen, daß für kleine Startwinkel der Anteil der Zyklotronenergie an der Gesamtenergie geringer ist als für große Startwinkel. Das Elektron erfährt also ein größere Änderung des Magnetfeldes pro Zyklotronumlauf, was dazu führt, daß die Bedingung für den adiabatischen Transport (s. Gl. 2.6) verletzt wird. Dies wird durch einen großen Wert für Δ ausgedrückt. Ein weiterer Grund für dieses Verhalten liegt darin, daß Energieänderungen bei Elektronen mit kleinen Startwinkeln besonders stark in Δ deutlich werden.

Betrachten wir andererseits die Differenz der Transversalenergie zwischen Start- und Endpunkt der Elektronenbahn $\Delta E_{\perp}$, so wird deutlich, daß dieser Wert für wachsende Startwinkel ansteigt. Desweiteren ist eine Abhängigkeit von Startradius der Elektronen zu erkennen. So liegt $\Delta E_{\perp}$ für auf der Achse ($r=0$ mm) gestartete Elektronen im Bereich von betragsmäßig 10-25 eV, während achsenfern gestartete Elektronen Abweichungen der Transversalenergie von bis zu 250 eV zeigen. Für dieses Verhalten kann ebenfalls die stärkere Krümmung der Elektronenbahnen für die großen Startradien verantwortlich sein.

Es kann nicht ausgeschlossen werden, daß es sich bei diesen Ergebnissen um numerische Artefakte handelt.

Das Trappingvolumen

Das Trappingvolumen liegt für diese Konfiguration bei etwa 3070 cm^3 . Dies ist aufgrund des Kippwinkels der Transportelemente von 10° nicht direkt mit dem Entwurf aus Troitsk zu vergleichen, da dieser von einem Kippwinkel von 20° ausgeht, und so ein insgesamt größeres Volumen im niedrigen Magnetfeld hat. Es wurde daher ein weiterer Test des Trappingvolumens durchgeführt, bei dem die Transportelemente um 20° gegeneinander gekippt waren. In dieser Konfiguration beträgt der Phasenraum der Teilchenfalle 3400 cm^3 .

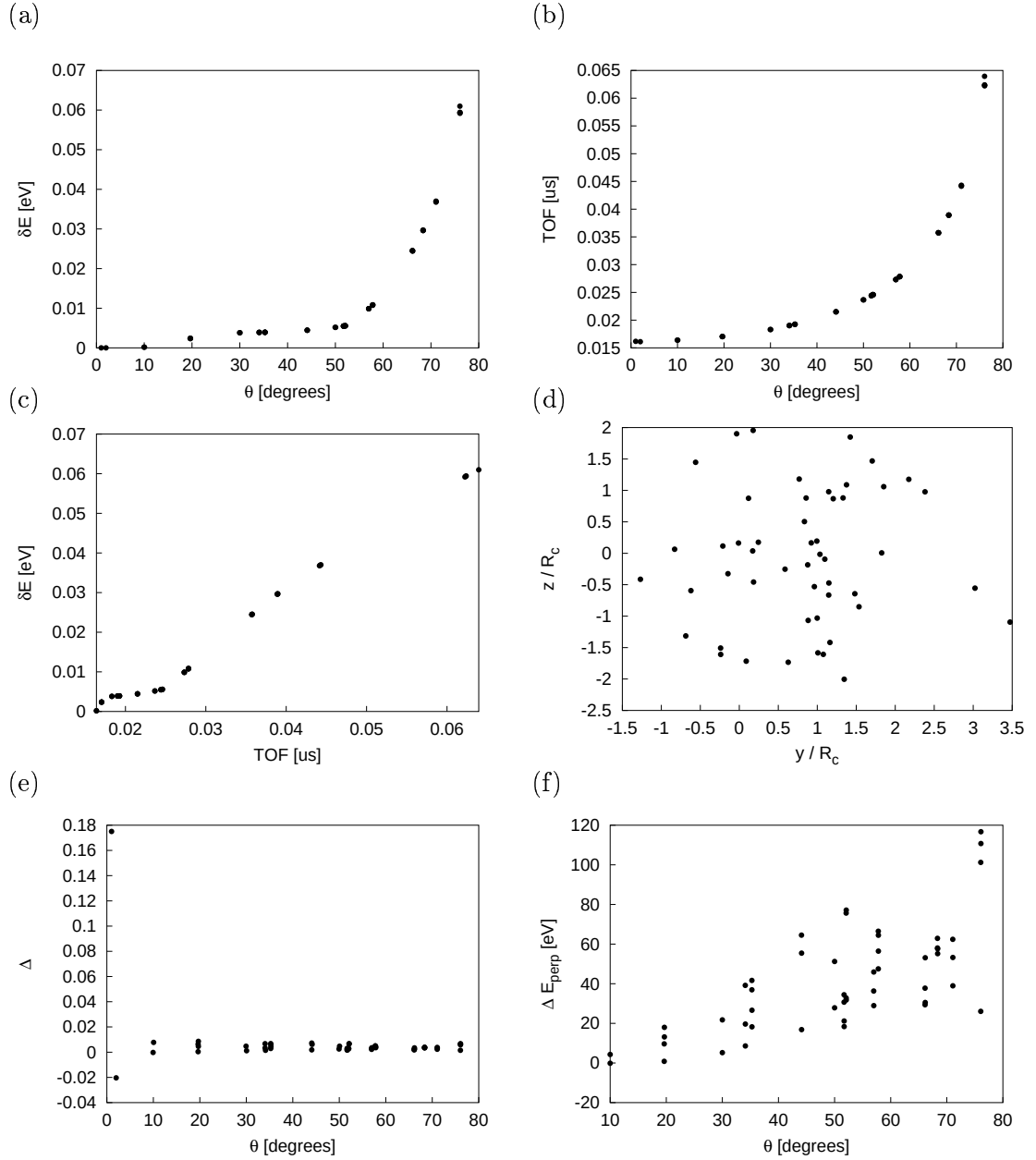


Abbildung 4.3: Das ACCEL-System. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von -32mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. In (a) ist die Energieverletzung δE gegen den Startwinkel θ aufgetragen. (b) zeigt die Flugzeit in Abhängigkeit von θ , (c) die Energieverletzung in Abhängigkeit von der Flugzeit. Die Abweichung von einer Punkt zu Punkt - Abbildung der Elektronen ist in Einheiten des Zyklotronradius in (d) dargestellt. Abb (e) zeigt den Δ -Parameter, ein Maß für die Verletzung der Adiabasie, (f) zeigt die Abweichung der Transversalenergie zwischen Startpunkt und Endpunkt der Elektronenflugbahn und somit auch einen Parameter für die Verletzung der Adiabasie, jeweils in Abhängigkeit von θ .

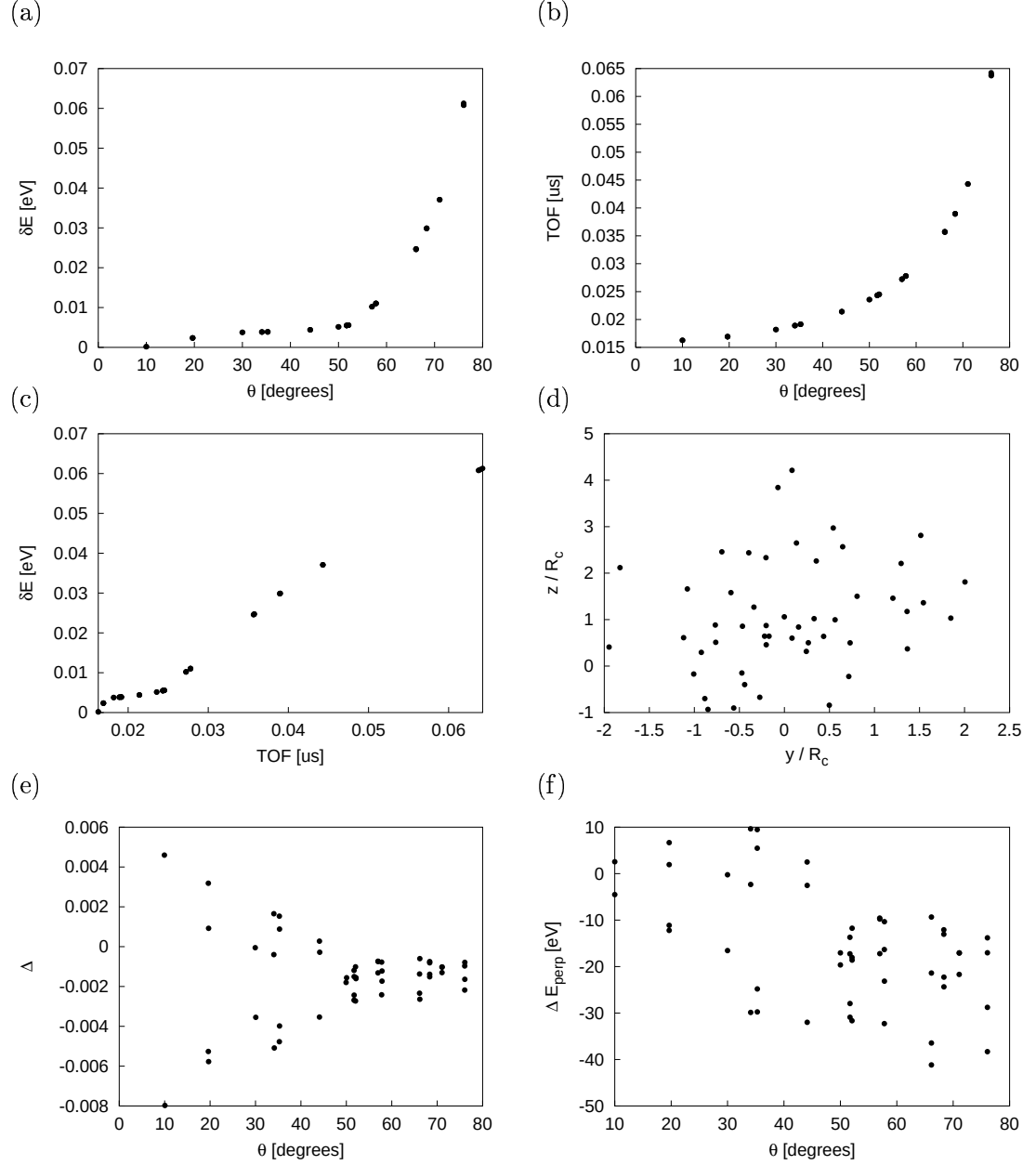


Abbildung 4.4: Das ACCEL-System. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von -16 mm von der Achse dargestellt. Die Plots entsprechen denen aus Abb. 4.3.

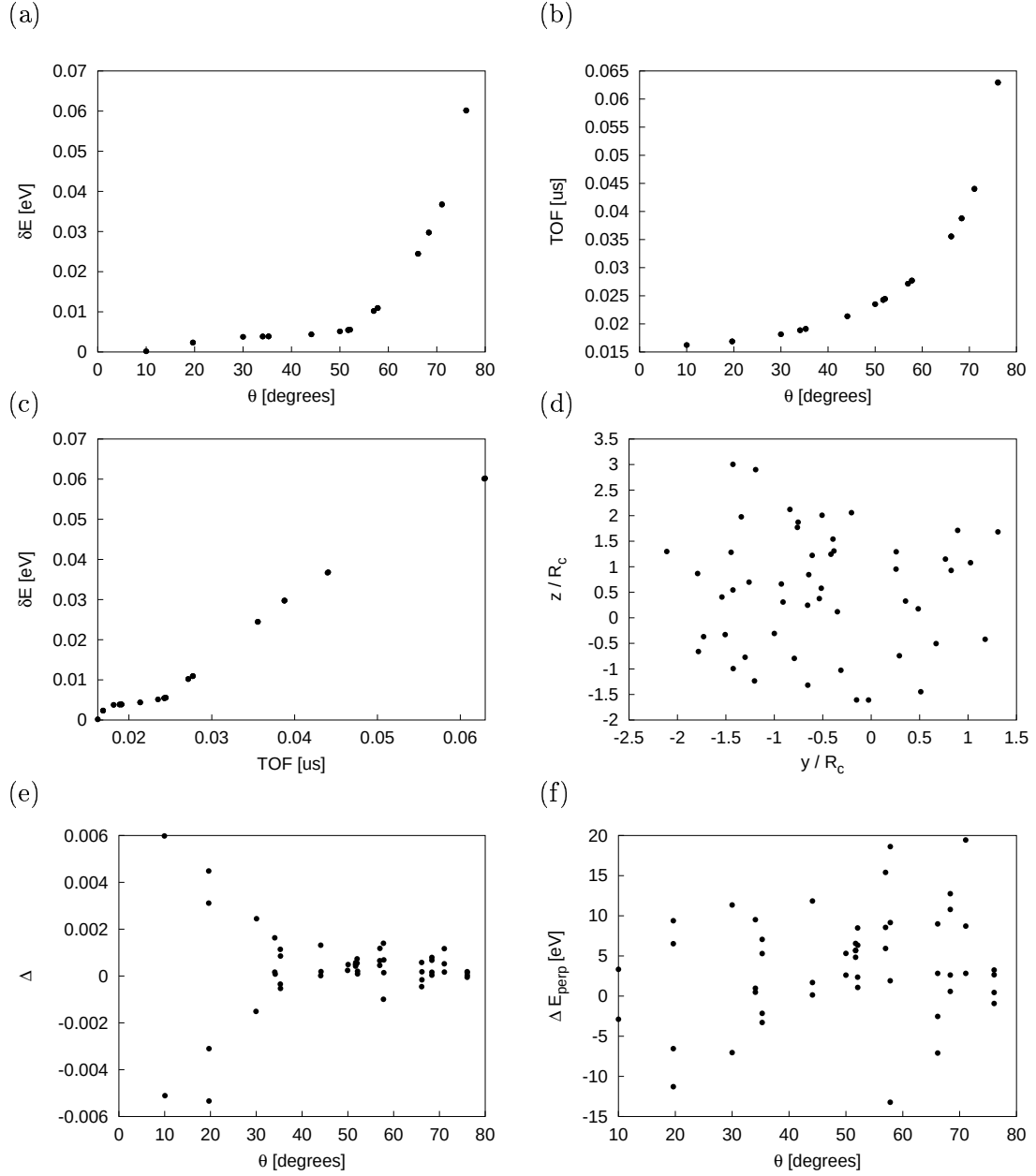


Abbildung 4.5: Das ACCEL-System. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von -8mm von der Achse dargestellt. Die Plots entsprechen denen aus Abb. 4.3.

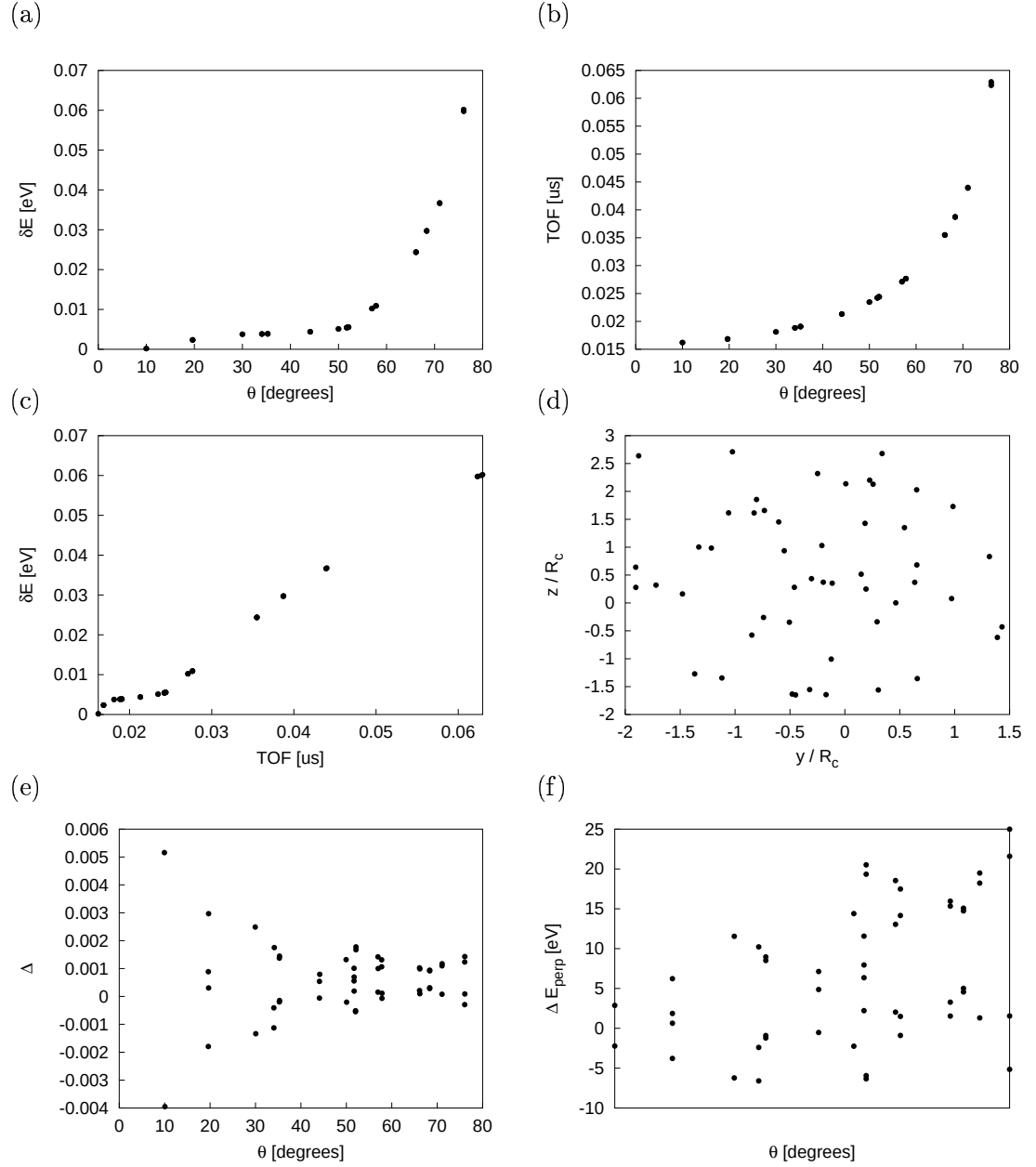


Abbildung 4.6: Das ACCEL-System. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 0mm von der Achse dargestellt. Die Plots entsprechen denen aus Abb. 4.3.

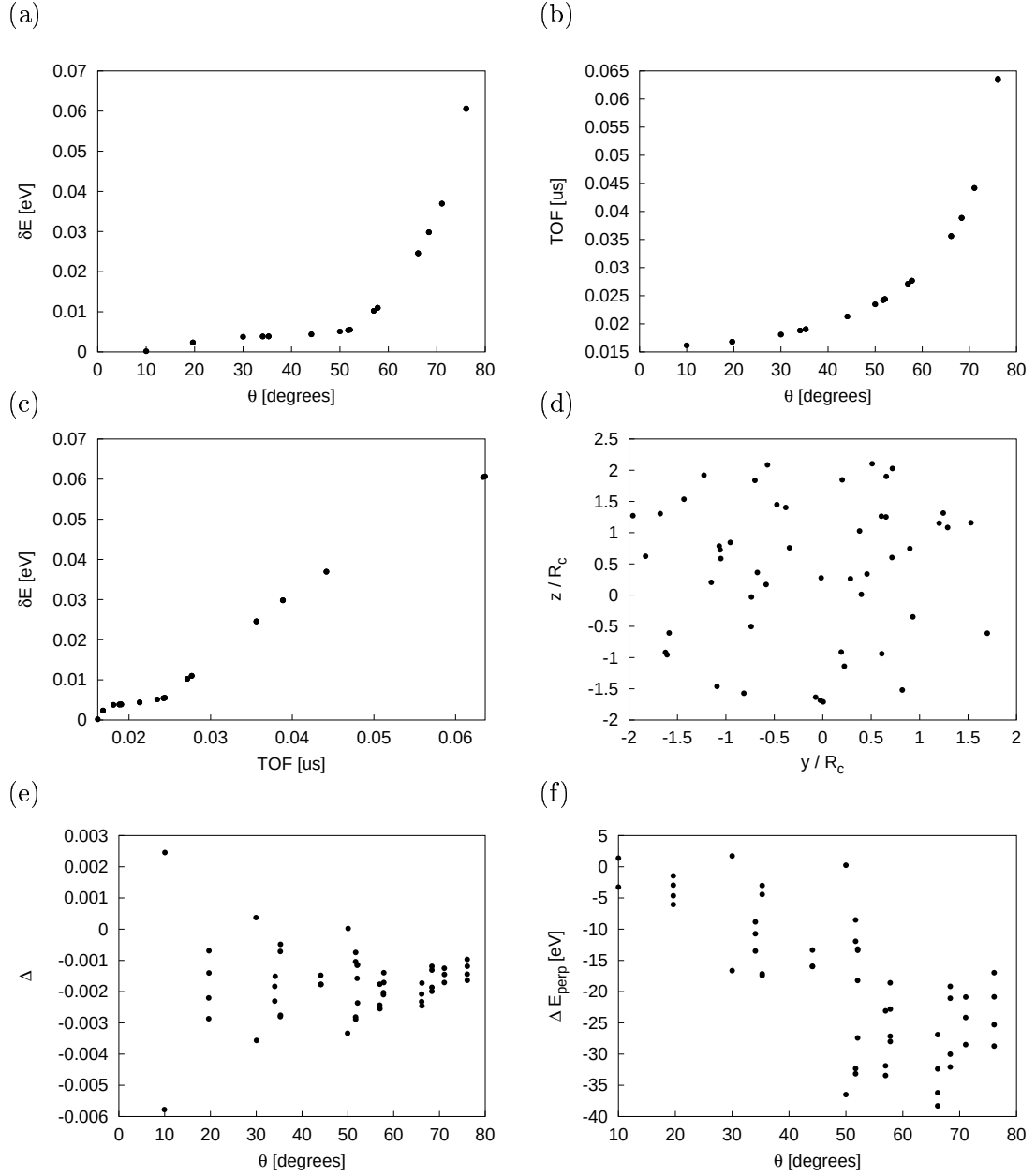


Abbildung 4.7: Das ACCEL-System. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 8mm von der Achse dargestellt. Die Plots entsprechen denen aus Abb. 4.3.

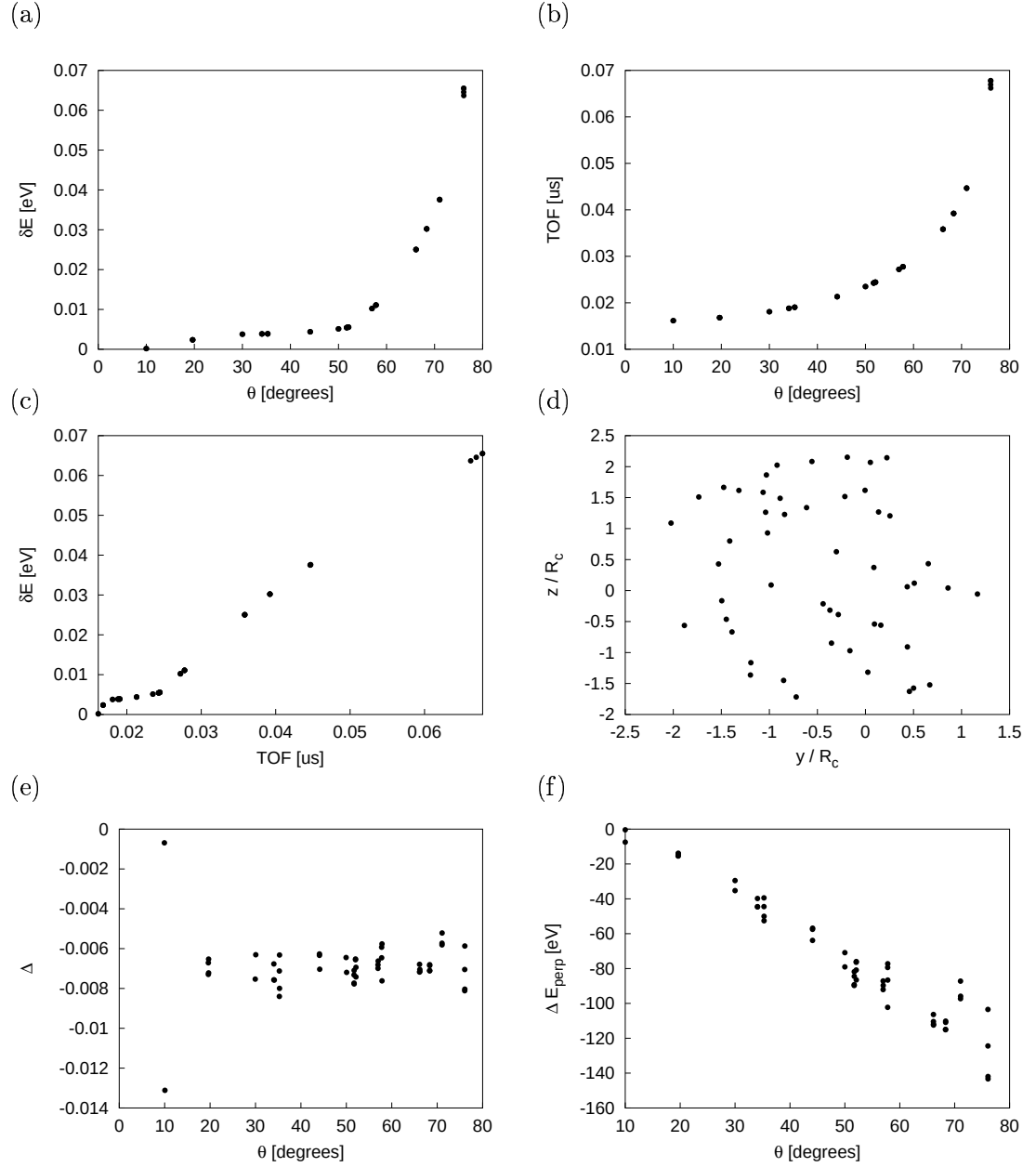


Abbildung 4.8: Das ACCEL-System. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 16mm von der Achse dargestellt. Die Plots entsprechen denen aus Abb. 4.3.

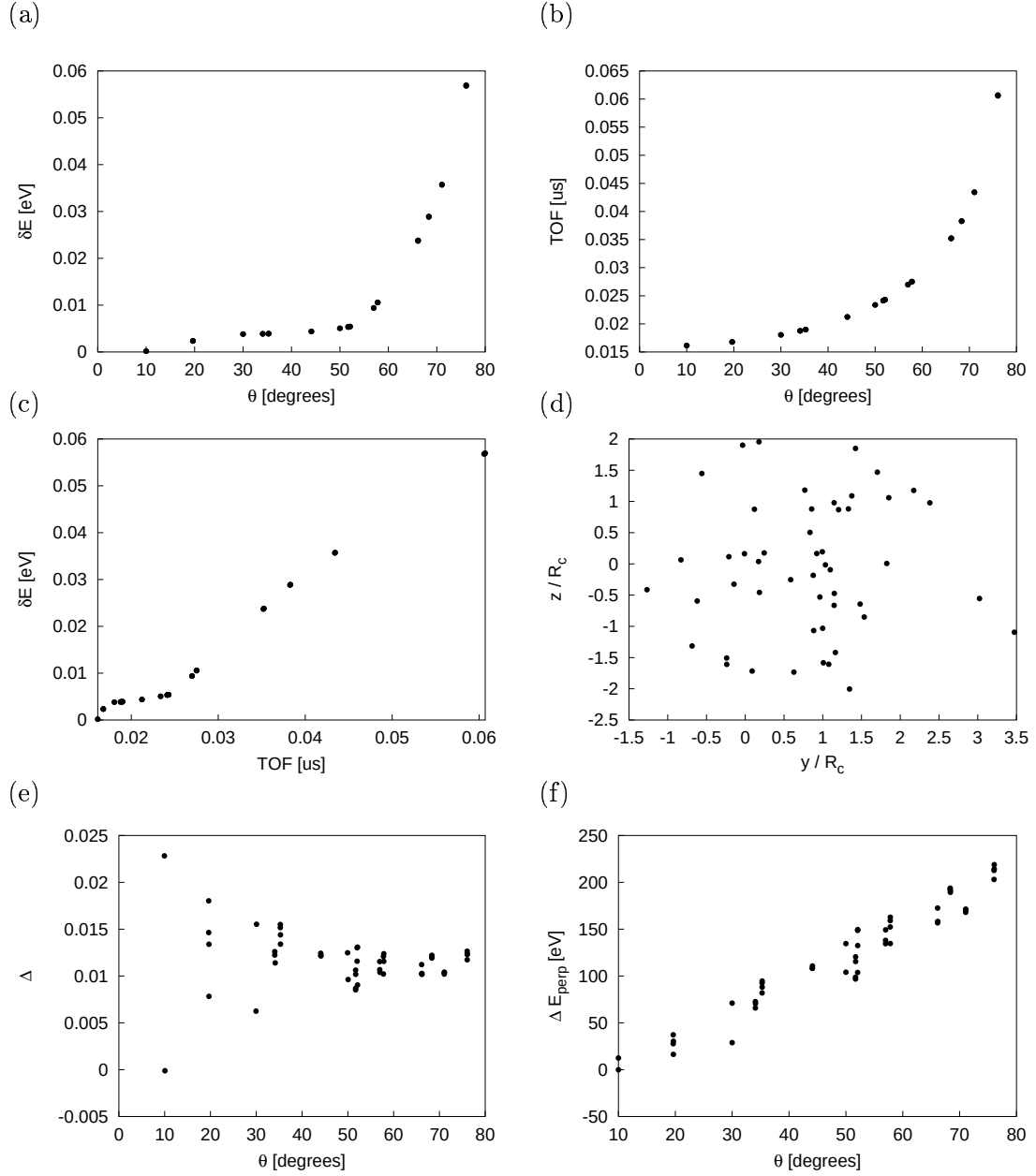


Abbildung 4.9: Das ACCEL-System. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 32mm von der Achse dargestellt. Die Plots entsprechen denen aus Abb. 4.3

4.2 Entwurf der Troitsker Arbeitsgruppe

Ein Transportelement des Entwurfes von Oleg Kazachenko besteht aus fünf Spulen, deren Parameter und Ausrichtung Tabelle 4.2 entnommen werden können.

Auch hier wurde zunächst überprüft, ob eine Berechnung der Magnetfelder mit Hilfe von BField3D gerechtfertigt war. Abb. 4.10 zeigt die Spulenkonfiguration von zwei um 20° gegeneinander gekippten Transportelementen mit den berechneten Konvergenzradien. Es ist zu erkennen, dass auch in dieser Konfiguration der gesamte abgebildete Flußschlauch innerhalb des Konvergenzradius der Reihenentwicklung (Gl. 3.2) liegt. Somit waren auch für dieses System Simulationen mit SIMION möglich.

Tabelle 4.2: Troitsk-Design eines Transportelements. Es sind die Parameter eines Transportelements des Entwurfes von O. Kazachenko aufgelistet. Die Größen entsprechen denen aus Tab. (4.1).

#	x [cm]	y [cm]	θ [°]	r [cm]	d [cm]	l [cm]	nI [A]
1	49.5	-9.84	0.0	20.0	2.0	6.0	288000
2	40.0	-7.7	0.0	8.0	2.0	19.0	912000
3	0	0.0	0.0	8.0	2.0	60.0	2880000
4	-40.0	7.7	0.0	8.0	2.0	19.0	912000
5	-49.5	9.84	0.0	20.0	2.0	6.0	288000

Die zentralen Transportspulen erzeugen ein Feld von 6 Tesla. Das Magnetfeld dieser Konfiguration ist also ein Tesla stärker als das Feld des Entwurfes von Accel. Auch in der Lücke zwischen zwei Transportelementen ist das Magnetfeld höher als in der Accel-Konfiguration. An dieser Stelle beträgt es etwa 1 T gegenüber 0.7 T. Auch diese Abweichung von den ursprünglichen Designforderungen basiert auf einem Mißverständnis, aber es ist möglich die für 6 T ausgelegten Spulen mit niedrigeren Strömen zu fahren und so ein Feld von 5 T zu erreichen. Die hier vorgestellten Simulationen wurden allerdings mit dem Feld von 6 T durchgeführt.

Auch für dieses System wurden die numerische Genauigkeit, die Abbildungseigenschaften und die Verletzung der Adiabasie untersucht. Zu diesem Zweck wurden die selben Verfahren verwendet, die schon in Abschnitt 4.1 beschrieben wurden. Die Resultate der Simulationen sind in den Abbildungen (4.11-4.17) dargestellt.

Betrachtungen zur numerischen Genauigkeit

Anhand der Abb. (4.11-4.17) (a-c) wird deutlich, daß das in 4.1 geschilderte Verhalten auch die numerischen Fehler in diesem System beschreibt. Es bleibt zu erwähnen, daß aufgrund des höheren Magnetfeldes für diesen Entwurf, insgesamt längere Flugzeiten auftreten (Abb. (4.11-4.17) (b)), dies hat aber keinen Einfluß auf die numerische Genauigkeit. Sie liegt auch für dieses System in der Größenordnung von 10^{-6} der

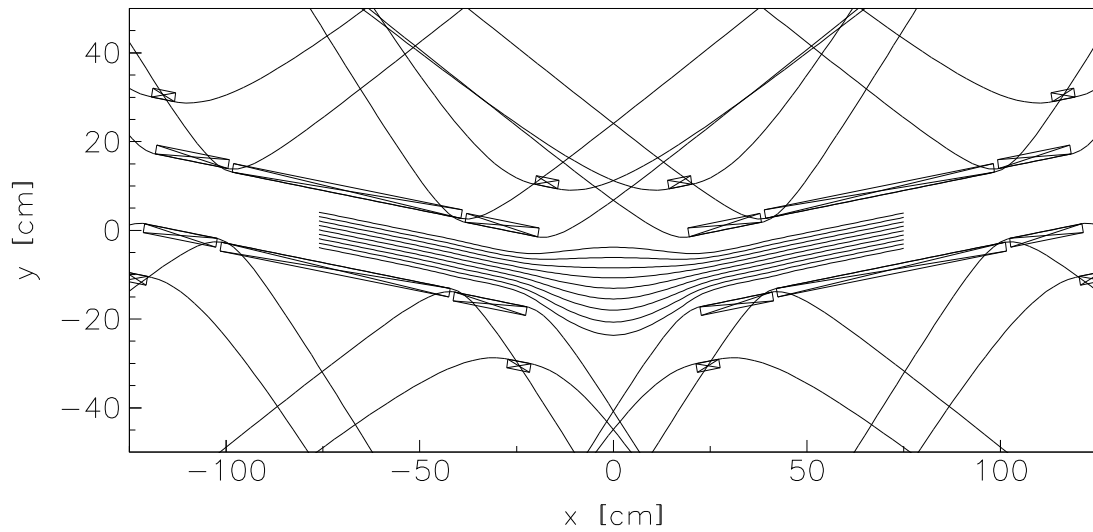


Abbildung 4.10: Der Troitsk-Entwurf Es sind zwei Transportelemente dargestellt, die um 20° gegeneinander gekippt sind. Desweiteren ist der berechnete Flußschlauch und der Konvergenzradius der Reihenentwicklung von BField3D dargestellt.

Elektronenenergie.

Betrachtungen zu den Abbildungseigenschaften

Die Abbildungen (4.11-4.17) (d) zeigen die Abweichung der Elektronenposition vom der erwarteten Position. Hier ist deutlich zu sehen, daß die Erwartung in der y-Richtung für alle Startradien erfüllt ist: Die Zielpositionen liegen im Abstand von zwei Zyklotronradien um den Punkt, der durch eine Punkt zu Punkt Abbildung erreicht würde. Dagegen sind in z-Richtung deutliche Abweichungen von dieser Erwartung zu erkennen. Die Ursache liegt in der sog. $\vec{R} \times \vec{B}$ -Drift, die auftritt, wenn ein geladenes Teilchen durch ein gekrümmtes Magnetfeld fliegt. Sie kann durch einen zusätzlichen Term korrigiert werden:

$$\vec{v} = \frac{2E_{||}}{eR^2B^2} \vec{R} \times \vec{B} \quad (4.1)$$

Wobei R der Krümmungsradius der führenden Feldlinie ist. Dieser Krümmungsradius liegt in unserem Fall ebenso wie der Magnetfeldvektor in der x-y-Ebene, so daß eine Drift in z-Richtung resultiert. Eine Abschätzung dieser Drift liegt bei einer Magnetfeldstärke von 1 T, einem Krümmungsradius von etwa 2 m und einer Parallelenergie von etwa 18 keV bei $80 \mu\text{m}$. Dies entspricht etwa 3 Zyklotronradien. Diese Abschätzung wurde für auf der Achse fliegende Elektronen getroffen. Geht man nun zu den negativen Startradien, so erkennt man eine stärkere Krümmung der führenden Feldlinie, was zu einer größeren Drift führt. In Richtung der positiven Startradien ist ein umgekehrtes Verhalten zu erkennen.

Betrachtungen zur Adiabasierletzung

Der in Abb. (4.11-4.17) (e) dargestellte Δ -Parameter zeigt auch für den Troitsker Entwurf die bereits beschriebene Überhöhung für kleine Startwinkel. Eine Erklärung dieses Verlaufes wurde schon in Abschnitt 4.1 gegeben.

Die Betrachtung von $\Delta E_{\perp}$ (Abb. (4.11-4.17) (f)) zeigt, daß die Differenz der Transversalenergie zwischen Start- und Endpunkt der Elektronenbahn keine Abhängigkeit vom Startwinkel θ aufweist. Es ist aber eine leichte Abhängigkeit vom Startradius zu erkennen: So wächst $\Delta E_{\perp}$ mit dem Betrag von r . Die Abweichungen bleiben aber stets geringer als 60 eV.

Das Trappingvolumen

Das Trappingvolumen beträgt für dieses Design etwa 6000 cm^3 . Dies ist sehr viel größer als das Phasenraumvolumen einer Teilchenfalle im Accel-Entwurf. Der Grund dafür liegt darin, daß in den Designanforderungen ein Pumpport mit 25 cm Durchmesser gefordert wurde, der auch in beiden Designs realisiert ist. Im Accel-Design geht diese Bohrung jedoch in ein Rohr mit 15 cm Durchmesser über, während im Troitsker Design das Rohr mit 25cm bis in das Innere der Lücke geführt wird, dafür wird ein wesentlich größeres Volumen benötigt. Dies führt zu dem größeren Phasenraumvolumen der Teilchenfalle.

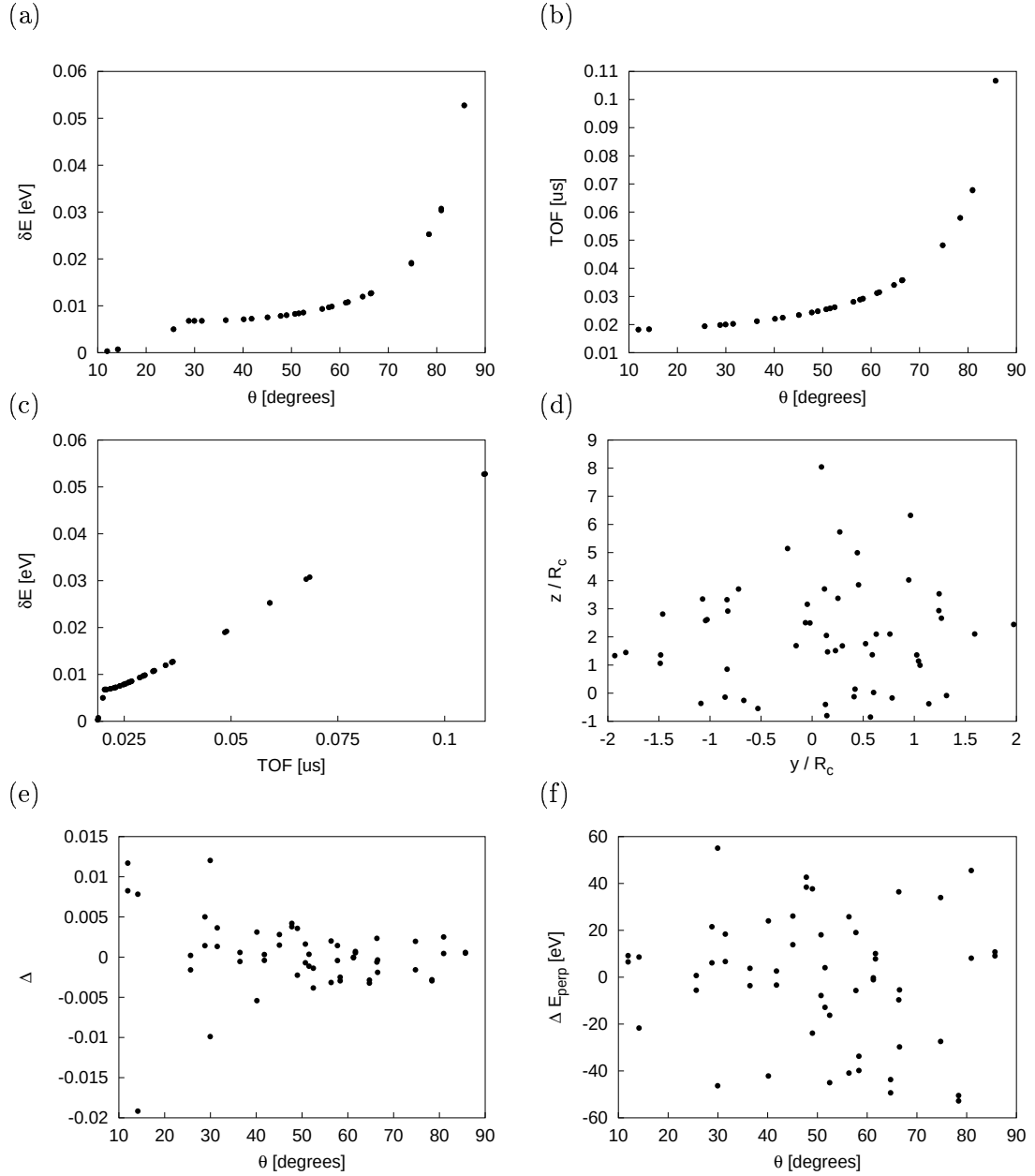


Abbildung 4.11: Der Troitsk-Entwurf. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von -32 mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. In (a) ist die Energieverletzung δE gegen den Startwinkel θ aufgetragen. (b) zeigt die Flugzeit in Abhängigkeit von θ , (c) die Energieverletzung in Abhängigkeit von der Flugzeit. Die Abweichung von einer Punkt zu Punkt - Abbildung der Elektronen ist in Einheiten des Zyklotronradius in (d) dargestellt. Abb (e) zeigt den Δ -Parameter, ein Maß für die Verletzung der Adiabasie, (f) zeigt die Abweichung der Transversalenergie zwischen Startpunkt und Endpunkt der Elektronenflugbahn und somit auch einen Parameter für die Verletzung der Adiabasie, jeweils in Abhängigkeit von θ .

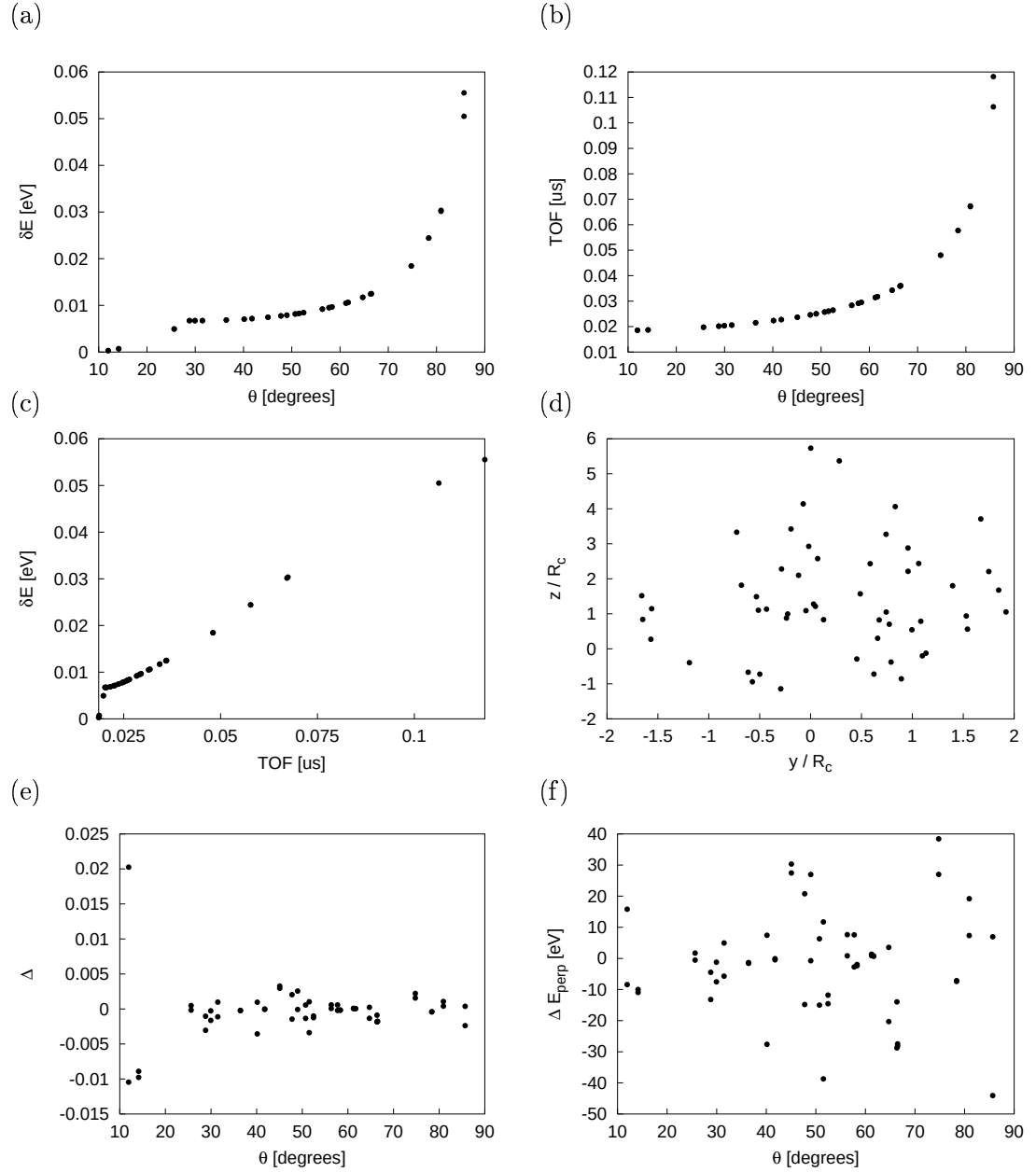


Abbildung 4.12: Der Troitsk-Entwurf. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von -16 mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. Die Plots entsprechen denen aus Abb. (4.11).

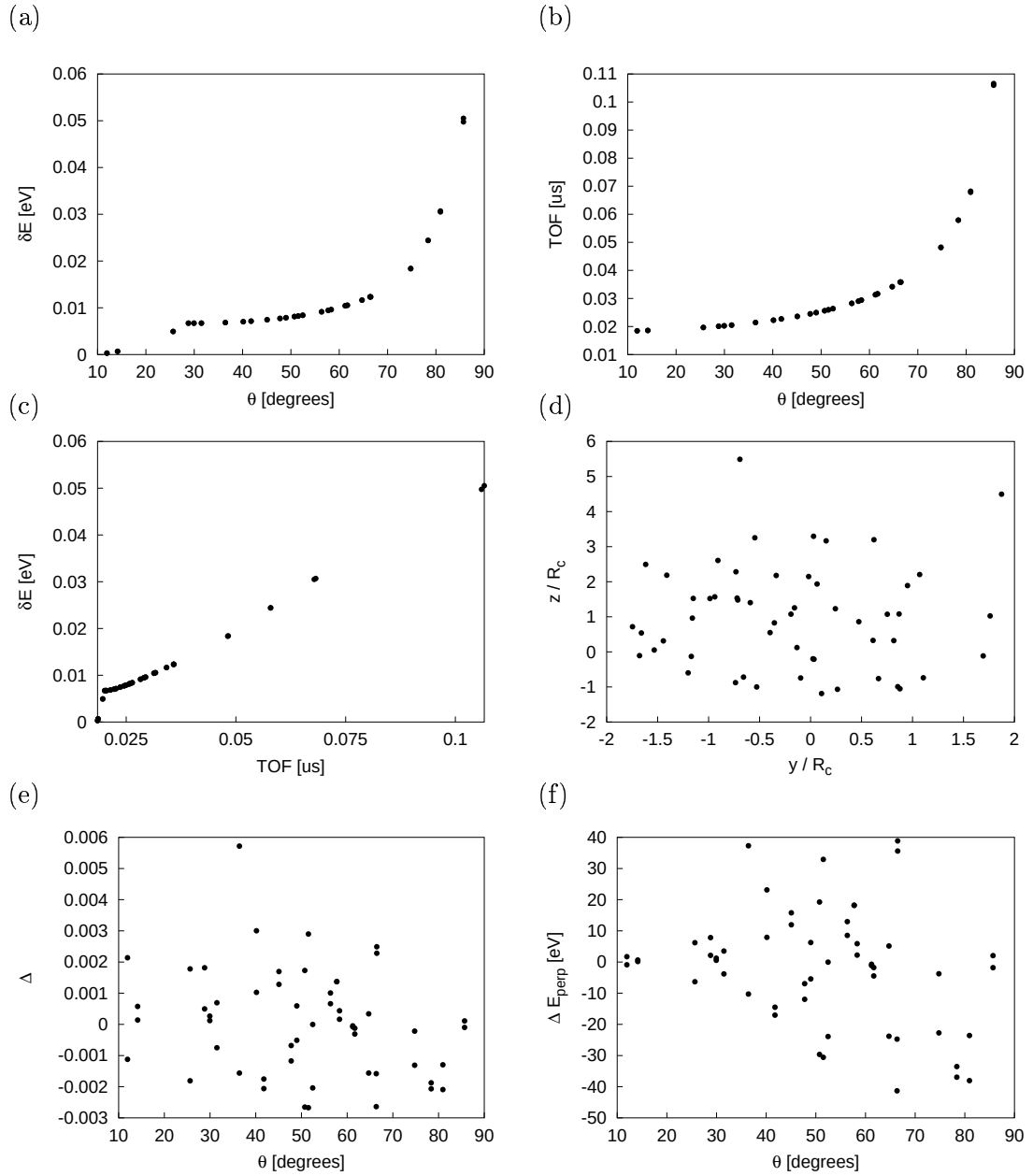


Abbildung 4.13: Der Troitsk-Entwurf. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von -8 mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. Die Plots entsprechen denen aus Abb. (4.11).

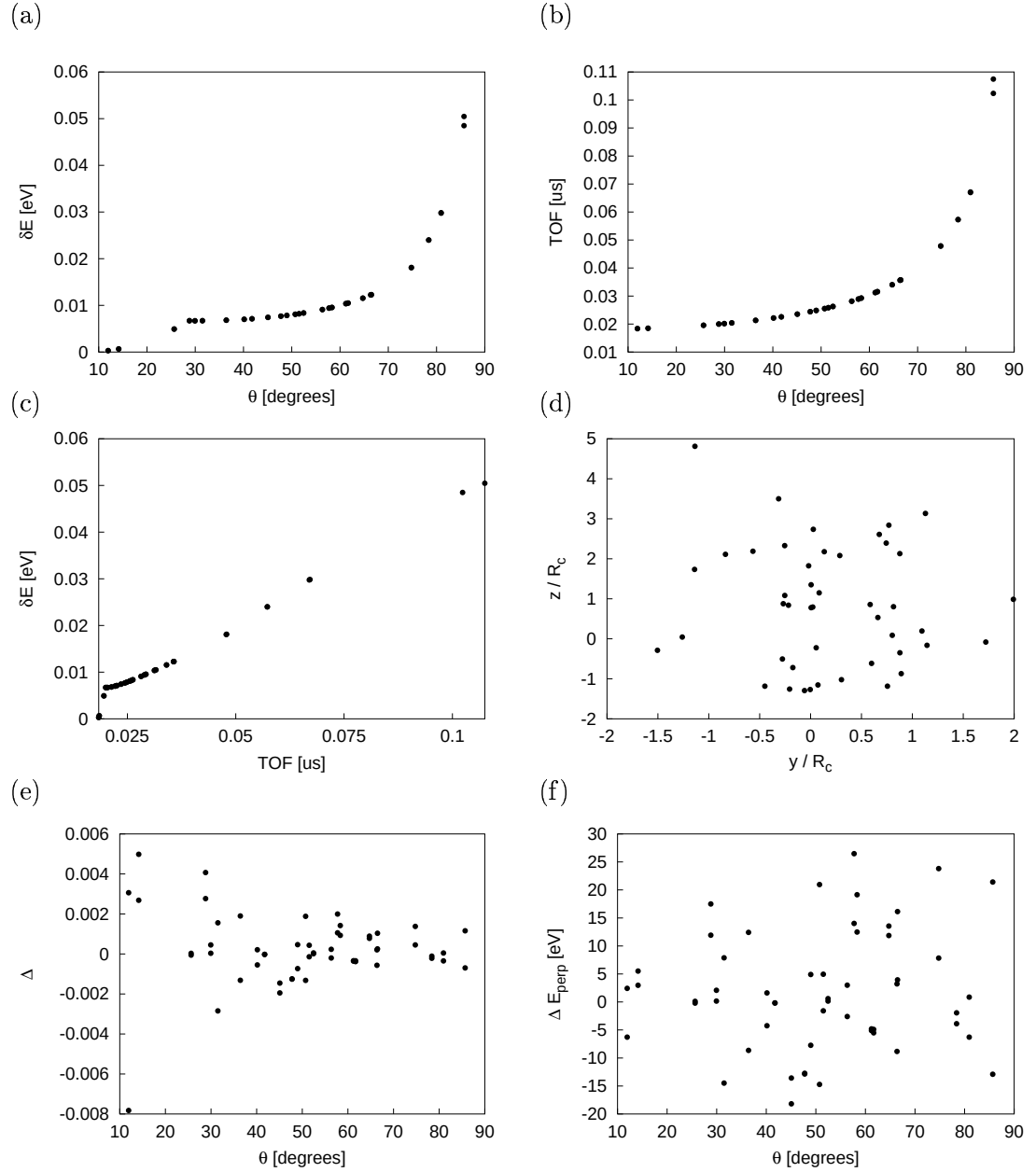


Abbildung 4.14: Der Troitsk-Entwurf. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 0 mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. Die Plots entsprechen denen aus Abb. (4.11).

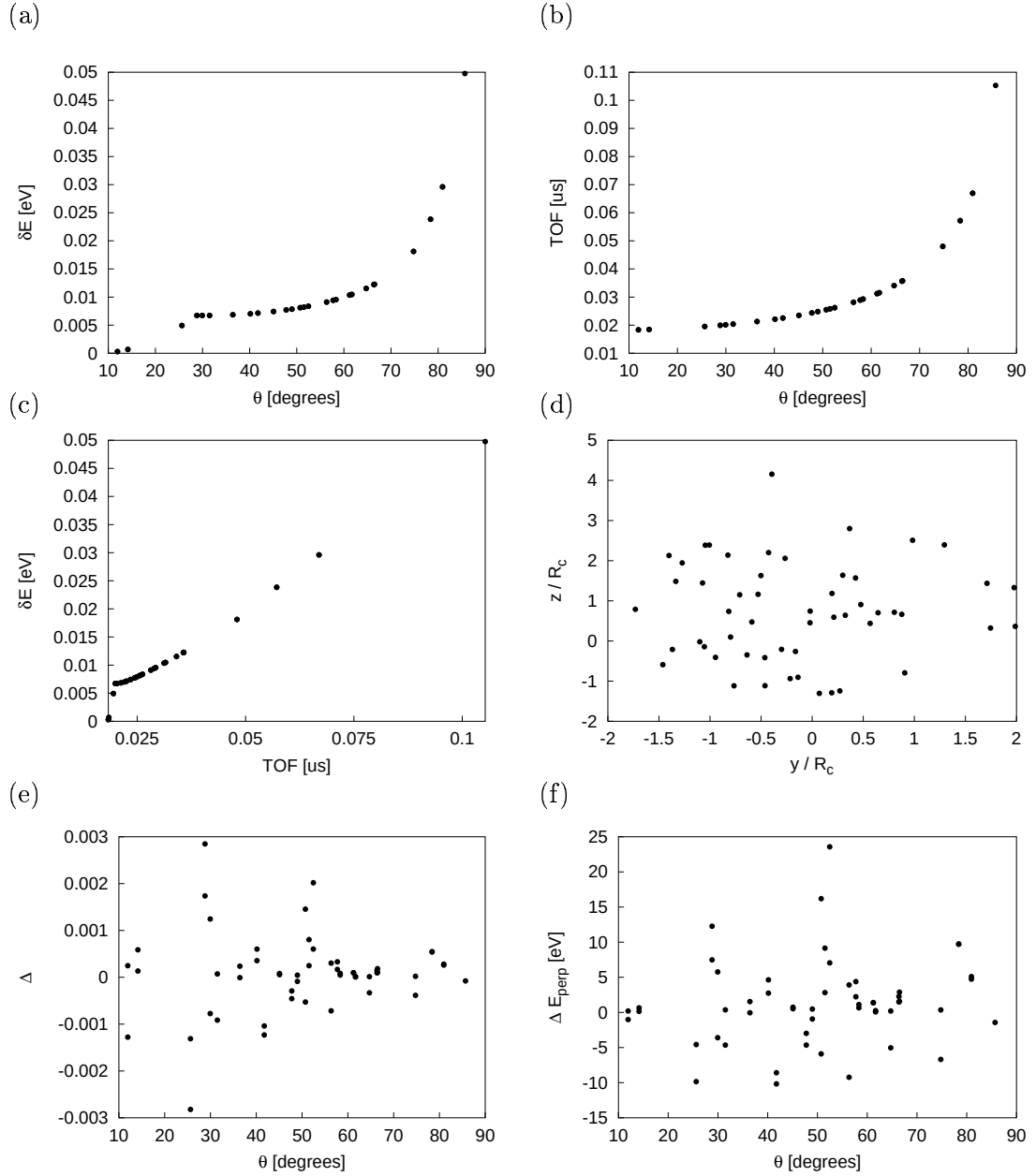


Abbildung 4.15: Der Troitsk-Entwurf. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 8 mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. Die Plots entsprechen denen aus Abb. (4.11).

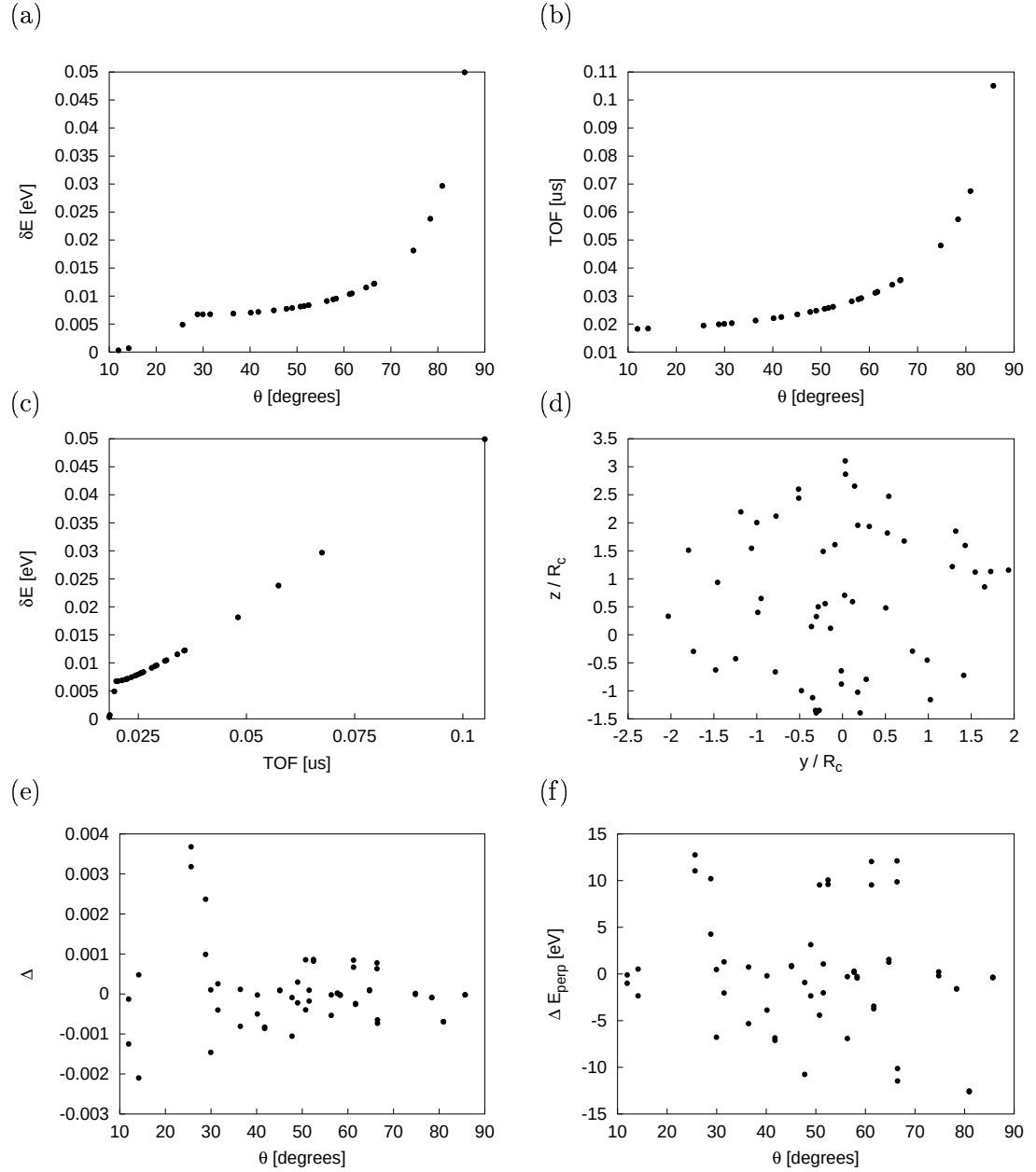


Abbildung 4.16: Der Troitsk-Entwurf. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 16 mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. Die Plots entsprechen denen aus Abb. (4.11).

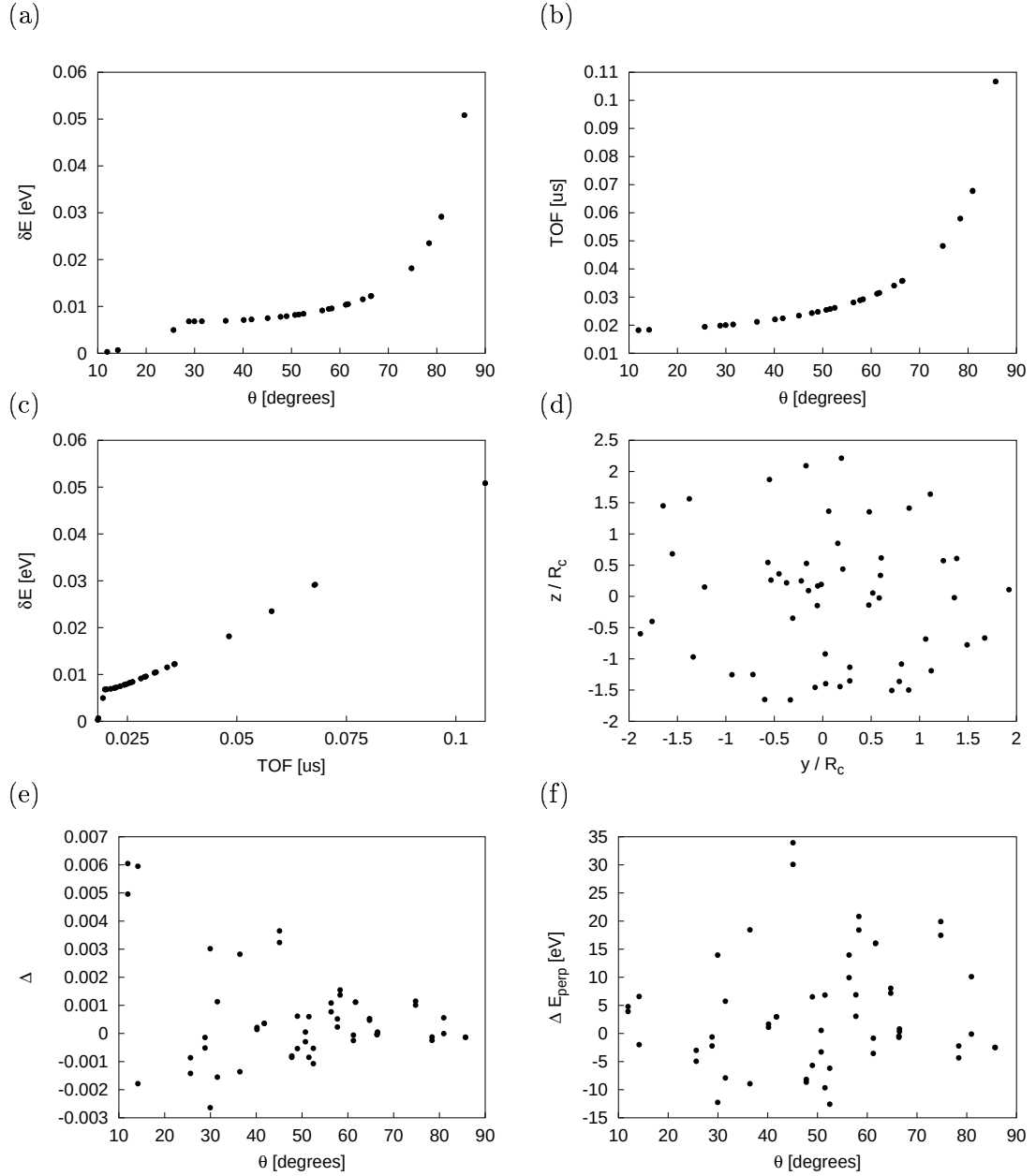


Abbildung 4.17: Der Troitsk-Entwurf. Es sind die Simulationsergebnisse für die Elektronen mit einem Startradius von 32 mm von der Achse dargestellt. Die Energie beträgt für alle Elektronen 18601 eV. Die Plots entsprechen denen aus Abb. (4.11).

4.3 Zusammenfassung und Wertung

Es wurden zwei alternative Vorschläge für ein Transportsystem untersucht. Beide Systeme weichen von der ursprünglichen Designanforderung ab, so ist der Winkel zwischen zwei Transportelementen im Entwurf der Firma Accel nicht wie gefordert 20° sondern 10° . Der Troitsker Entwurf geht von einem maximalen Magnetfeld von 6 T gegenüber geforderten 5 T aus.

Die Untersuchungen der Systeme umfaßte

- Betrachtung der zu erreichenden numerischen Genauigkeit
- Untersuchung der Abbildungseigenschaften
- Untersuchung der adiabatischen Transporteigenschaften
- Untersuchung des Phasenraumes einer möglichen Teilchenfalle

Beide Systeme konnten mit einer numerischen Genauigkeit untersucht werden, die in der Größenordnung von 10^{-6} der Elektronenenergie liegt. In Absolutwerten ausgedrückt entspricht dies einer Energieverletzung von etwa 70 meV im maximalen Fall. Generell ist festzuhalten, daß der numerische Fehler mit der Flugzeit der Elektronen und somit mit dem Startwinkel anwächst.

Die Untersuchung des Accel-Designs zeigt keine größere Abweichung von der erwarteten Punkt zu Kreis Abbildung. Die geringen Abweichungen die auftreten können durch numerische Ungenauigkeiten erklärt werden. Im Entwurf der Troitsker Arbeitsgruppe treten jedoch starke $\vec{R} \times \vec{B}$ -Driften auf, die unter Umständen zu einem Verlust von Elektronen im Transportsystem führen können, wenn man mehrere dieser Transportelemente hintereinander aufbaut. Um diesen Sachverhalt zu klären werden weitere Untersuchungen durchgeführt werden.

Der Vorteil des Entwurfes von O. Kazachenko liegt in den adiabatischen Transporteigenschaften. Verletzung der Transversalenergie ist für dieses System um etwa 50 % geringer als für das von Accel vorgeschlagene Design.

Der letzte untersuchte Punkt, das Phasenraumvolumen einer Teilchenfalle, zeigt einen weiteren Vorteil des Accelsystems auf. Das Trappingvolumen ist hier zur Hälfte kleiner als in dem Alternativvorschlag. Der Grund dafür liegt in der Verwendung eines dünneren Rohres im Inneren des Pumpports. Im Accel-Entwurf ist ein Rohr mit 150mm Durchmesser vorgesehen, während im Entwurf aus Troitsk eins mit 250mm Verwendung findet.

Bevor man eine Entscheidung über den Bau des Transportsystems fällt sollten weitere Untersuchungen dieser Systeme durchgeführt werden. Dazu sollten die Entwürfe

so modifiziert werden, daß sie den geforderten Parametern entsprechen. Insbesondere sollten identische Kippwinkel und Magnetfeldstärken, sowie gleiche Leitwerte des Pumpsystems berücksichtigt werden.

Aufgrund der vorliegenden Ergebnisse kann man sagen, daß das Design von O. Kazachenko bessere adiabatische Eigenschaften hat als der Entwurf von ACCEL. Dies ist durch die großen Spulen zu erklären, die das Magnetfeld am Ende der Transportelemente weniger stark abfallen lassen.

Mit diesen Angaben können aus der Kombination der beiden vorgestellten Designs die Werte der Adiabasierletzung, der Abbildungseigenschaften und des Trappingvolumens für einen Kompromiss abschätzen.

Kapitel 5

Das Hauptspektrometer

In diesem Kapitel werden verschiedene Spektrometerdesigns bezüglich der in Kapitel 3 gestellten Anforderungen getestet. Dazu werden die ebenfalls in Kapitel 3 beschriebenen Programme benutzt.

Auch hier ist es das Ziel den adiabatischen Transport und das Abbildungsverhalten zu testen. Weiterhin wird die Flugzeit der Elektronen untersucht. Zur Bewertung der Ergebnisse wird in diesem System ebenfalls die numerische Genauigkeit untersucht. Die beiden hier vorgestellten Designs wurden in Rahmen dieser Arbeit entworfen.

Bevor die im folgenden vorgestellten Simulationen durchgeführt wurden, fand ein Test der Transmissionseigenschaften der Entwürfe mit Hilfe des Programms *Transmission*, das in einer früheren Arbeit entwickelt wurde [Fic00], statt. Das Programm überprüft, ob die Parallelenergie der Elektronen an jeder Stelle des Spektrometers positiv ist. Dies geschieht mit Hilfe von Gl. (2.9). Im Rahmen dieser Untersuchungen wurde festgestellt, das beide Designs die Transmissionsbedingungen erfüllen.

5.1 Das Einelektroden-Design

In Kapitel 2 wurden die Aufgaben des elektrostatischen Gegenfeldes erläutert. Es dient dazu die Energie der von den Quellen kommenden Elektronen zu analysieren und es hat die Aufgabe dafür zu sorgen, daß die Bewegung der Elektronen adiabatisch bleibt. Daraus ergeben sich die in Kapitel 3 genannten Designanforderungen:

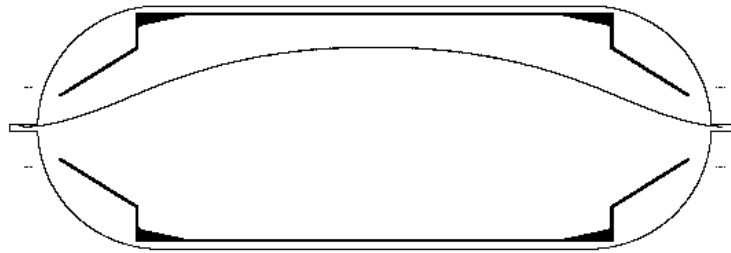
- Hohe Homogenität des elektrischen Feldes in der Analysierebene
- Identität der magnetischen und elektrischen Analysierebene
- Vermeidung von Teilchenspeichern

Es wurden bereits Designs mit mehreren Elektroden simuliert und getestet [Fic00]. In dieser Arbeit werden Systeme, die das Retardierungsfeld mit nur einer Elektrode erzeugen untersucht. Bei diesen Untersuchungen wurde jeweils dasselbe Magnetdesign verwendet. Das maximale Magnetfeld beträgt 3 T, das minimale Magnetfeld in der Analysierebene 3 Gauss, wobei hier eine Homogenität von 7% erreicht wird. Diese Feldwerte entsprechen einem Skalierungsfaktor f von 0.6. Für diese untere Grenze des f -Bereiches werden stärkere Abweichungen erwartet, als für größere Magnetfelder. Eine genaue Beschreibung des Magnetdesigns ist in [Fic00] zu finden.

Es wurden ebenfalls Untersuchungen mit einer Magnetkonfiguration vorgenommen, die ein maximales Feld von 5 T im Spektrometersolenoiden und ein Feld von 5 Gauss in der Analysierebene erzeugt. Auch hier wurde eine Homogenität des Magnetfeldes in der Analysierebene von 7% erreicht. Diese Untersuchungen werden hier jedoch nicht vorgestellt.

Bei dem ersten Entwurf handelt es sich um eine einzelne Elektrode im Inneren des Vakuumtanks (s. Abb. (5.1(a))). An diese Elektrode wurde ein Potential von $U_0 = -18600$ V angelegt. Abb. (5.1(b)) zeigt den Zentralbereich des Spektrometers und den Verlauf zweier Äquipotentiallinien. Der erreichte Potentialdurchgriff liegt bei 1.13 V. Dieser Wert entspricht nicht ganz dem geforderten Wert, der unterhalb von 1 V liegen sollte, was auf die große Öffnung der Elektrode zurückzuführen ist. In Abb. (5.1) ist der Verlauf des Potentials dieser Elektrodenkonfiguration in der Analysierebene dargestellt.

(a)



(b)

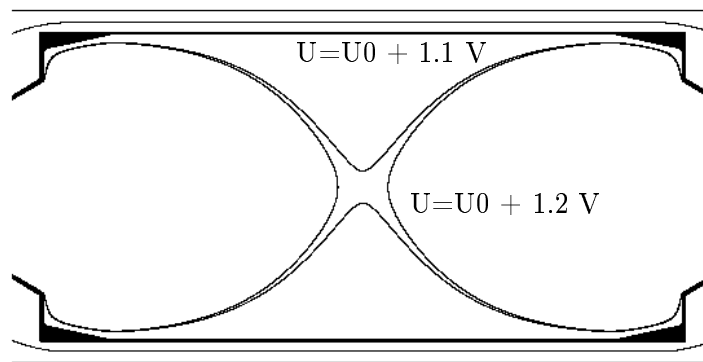


Abbildung 5.1: Das Einelektroden-Design. In (a) ist der Vakuumtank mit einer Elektrode dargestellt. Das gezeigte Elektron wurde bei einem Radius $r = 24\text{mm}$ kurz vor dem Maximum des Magnetfeldes gestartet. (b) zeigt den Zentralbereich des Einelektroden-Designs und den Verlauf der Äquipotentiallinien mit $U_0 + 1.1\text{V}$ und $U_0 + 1.2\text{V}$. Der Potentialdurchgriff beträgt 1.13 V .

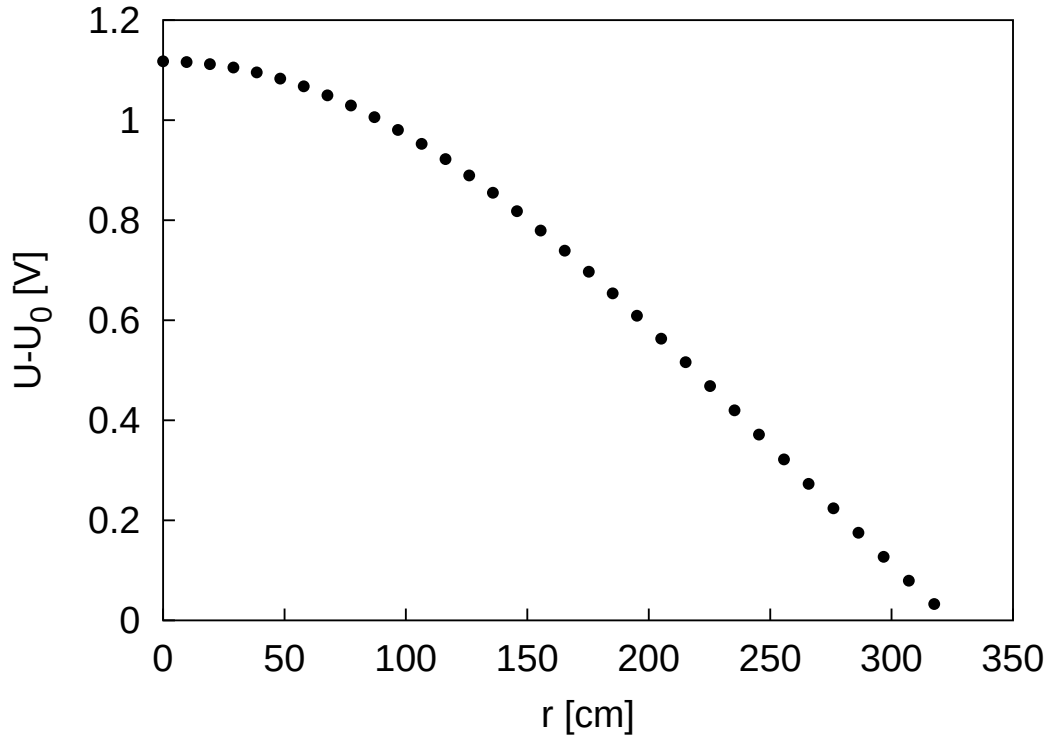


Abbildung 5.2: Einelektrodenesign. Dargestellt ist der Verlauf des elektrischen Potentials U entlang der Analysierebene.

Untersuchung der numerischen Genauigkeit

Zur Untersuchung der numerischen Genauigkeit wurde, wie auch schon bei der Untersuchung der Transportsysteme, die Energieerhaltung überprüft. Eine Abweichung von der Gesamtenergie kann nur durch numerische Artefakte auftreten, da ein solches Verhalten unphysikalisch ist. Im Gegensatz zu der Untersuchung des Transportsystems muß an dieser Stelle noch das elektrische Feld mit in Betracht gezogen werden, so daß sich für die Verletzung der Gesamtenergie ergibt:

$$\delta E = (E + eU)_1 - (E + eU)_2 \quad (5.1)$$

Der Startpunkt (1) liegt dabei kurz vor dem Maximum des Magnetfeldes eines Spektrometersolenoiden und der Endpunkt in der Analysierebene.

Die Abbildungen (5.3-5.7) zeigen die Ergebnisse der Untersuchung der Energieerhaltung. Es wurde die Flugzeit der Elektronen (jeweils in (a)) dargestellt sowie die Verletzung der Gesamtenergie δE (jeweils in (b)) in Abhängigkeit vom Startwinkel θ untersucht. Um eine Korrelation dieser Größen festzustellen wurde in (c) δE gegen die Flugzeit aufgetragen.

Zunächst wurden Elektronen mit einem Energieüberschuß von 1eV und den Startstrahlen von $r = -32$ mm (Abb.(5.3)) und $r = 0$ mm (Abb.(5.4)) betrachtet. Der Startstrahl

us von 32 mm wurde gewählt, da dies der äußersten Elektronenflugbahn entspricht. Für beide Startradien ist zu erkennen, daß mit ansteigenden Startwinkeln die Flugzeit und auch die Verletzung der Gesamtenergie wächst. Dies ist wie schon in den Betrachtungen zu den Transportsystemen auf die steigende Anzahl an Rechenschritten, die mit der Zunahme an Transversalenergie einher gehen, zu erklären. Für den Startradius $r=32$ mm ist eine Aufspaltung von δE zu erkennen. Dies ist auf geringe Potentialdifferenzen in der Analysierebene zurückzuführen, die aufgrund von Abweichungen von der Punkt-zu-Punkt-Abbildung auftreten (s. Abschnitt Untersuchungen der Abbildungseigenschaften). Eine weitere Erklärungsmöglichkeit liegt im Azimutwinkel, unter dem das Elektron der Krümmung der Führungsfeldlinie folgt. Insgesamt ist eine Abweichung von der Gesamtenergie von bis etwa 10 meV aufgetreten.

Die Abb. (5.5 -5.7) zeigen die Resultate für Elektronen mit Startradien von $r=32$ mm und Überschußenergien von 10, 100 und 1000 eV. Wie erwartet nimmt die Flugzeit mit größeren Startenergien ab. Dennoch ist auch in diesen Plots das Ansteigen der Flugzeit und der Energieverletzung mit wachsendem Startwinkel zu beobachten. Da der Betrag der Energieabweichungen trotz der stark unterschiedlichen Flugzeiten in der gleichen Größenordnung liegt, ist festzustellen, daß der Hauptanteil dieser Größe auf die unterschiedlichen Potentiale in der Analysierebene zurückzuführen sind.

Untersuchung der Adiabasie

Die Adiabasie wurde durch den Δ -Parameter (Gl. 3.8) und die Verletzung der Transversalkomponente der Energie untersucht. Auch in diesen Betrachtungen wurde als Startpunkt eine Position kurz vor den Maximum des Magnetfeldes des Spektrometersolenoiden und als Endpunkt die Analysierebene genutzt. Unter adiabatischen Bedingungen wird für $\Delta = 0$ erwartet.

Die Ergebnisse der Simulationen sind in den Abb. (5.8-5.11) dargestellt. In den Abbildungen (a) und (c) sind jeweils die Δ -Parameter und in (b) und (d) die Abweichung der Transversalenergie $\delta E_{\perp}$ in Abhängigkeit von θ dargestellt. Es ist anzumerken, daß Δ für $\theta = 0^{\circ}$ den Wert -1 annimmt, da für diesen Fall die gesamte Energie in der longitudinalen Bewegung vorliegt. In den Diagrammen wird daher auf eine Darstellung von $\theta = 0^{\circ}$ verzichtet, da sie ansonsten unlesbar werden. Die Informationen sind aber weiterhin in den Plots für $\Delta E_{\perp}$ zu entnehmen.

In Abb. (5.8) und (5.9) sind die Resultate für die Startradien $r=32$ mm dargestellt, die Elektronen hatten einen Energieüberschuß von 1 eV ((5.8(a) und (b)), 10eV (5.8(c) und (d)), bzw. 100 eV und 1 keV (5.9). Es wird deutlich, das die Verletzung der Adiabasie mit wachsender Überschußenergie ansteigt. Dieses Verhalten ist auch zu erwarten, da eine Bedingung für die Adiabasie eine geringe relative Änderung des Magnetfeldes $\frac{\Delta B}{B}$ während eines Zyklotronumlaufes ist. Mit steigender Elektronenenergie nimmt jedoch auch die Zyklotronschrittlänge zu, so daß diese Bedingung

verletzt wird. Um diesen Zusammenhang zu verdeutlichen, ist in Abb. (5.12) die Varianz der $\delta E_{\perp}$ gegen die Überschußenergie aufgetragen.

Diese Untersuchung wurde auch für auf der Achse fliegende Elektronen durchgeführt. Die zugehörigen Resultate sind in Abb. (5.10 und 5.11) dargestellt. Insgesamt ist für diese Elektronen eine geringere Verletzung der Adiabasie festzustellen, was auf die nicht gekrümmte Führungsfeldlinie zurückzuführen ist.

Untersuchung der Abbildungseigenschaften

Um im Spektrometer auftretende Driften zu erkennen, wurden die Abbildungseigenschaften des Spektrometers untersucht. Dazu wurden die Koordinaten bei denen die Elektronen die Analysierebene passiert haben mit der erwarteten Position verglichen. Diese Sollposition wurde mit Hilfe der Magnetfeldlinienberechnung von Bfield_3D bestimmt. Abweichungen vom erwarteten Verhalten können in gewissem Maß auch auf eine ungenaue Bestimmung der Magnetfeldlinie zurückgeführt werden. Um diesen Sachverhalt zu prüfen wird im Rahmen der Diplomarbeit von T. Thümmeler eine neue Routine zur Feldlinienberechnung auf der Basis des Runge-Kutta-Verfahrens entwickelt.

In Abb. (5.13) sind die Differenzen zwischen der Sollposition und der tatsächlichen Position in Einheiten des Zyklotronradius aufgetragen. Es wird erwartet, daß sich die Elektronen in einem Kreis mit einem Radius, der zwei Zyklotronradien entspricht, befinden. Diese Erwartung beruht darauf, daß für den adiabatischen Transport das magnetische Moment $\mu = \frac{E_{\perp}}{B}$ konstant ist. Da für den Zyklotronradius der Zusammenhang $r_c \propto \frac{\sqrt{E_{\perp}}}{B}$ gilt, ergibt sich über $r_c \propto \frac{1}{\sqrt{B}}$, daß $r_c^2 \cdot B$ konstant ist.

Abb. (5.13(a)) zeigt die Ergebnisse für auf der Achse fliegende Elektronen der Überschußenergien 1eV, 10eV, 100 eV und 1 keV. Für diese Elektronen ist das erwartete Verhalten sehr gut zu erkennen. Dies ist auch leicht zu verstehen, da die Führungsfeldlinie nicht gekrümmt ist. In Abb. (5.13(b)) werden jedoch starke Abweichungen von der Erwartung deutlich: Die einzelnen Elektronen liegen nun in einem Band der Breite $2R_c$, doch sind größere Abweichungen in der z-Richtung zu erkennen. Diese Abweichungen wachsen mit der Teilchenenergie an. Daraus läßt sich schließen, daß für dieses Verhalten ebenfalls die $\vec{R} \times \vec{B}$ - Drift (Gl. (4.1)) verantwortlich ist. Zusätzlich tritt im Spektrometer auch die $\vec{E} \times \vec{B}$ - Drift auf. Diese Drift entsteht, falls das elektrische Feld $\vec{E}$ und das magnetische Feld $\vec{B}$ nicht vollständig parallel zueinander sind. Die Zyklotronbahn eines geladenen Teilchens verläuft dann zum Teil gegen und zum Teil mit der zu dem Magnetfeld senkrechten Komponente des elektrischen Feldes. Daraus resultiert ein abwechselndes Abbremsen und Beschleunigen des Teilchens, was zu einer seitlichen Driftbewegung führt [Jac82]. Der resultierende Driftterm wird durch

$$\vec{v} = \frac{\vec{E} \times \vec{B}}{B^2} \quad (5.2)$$

beschrieben.

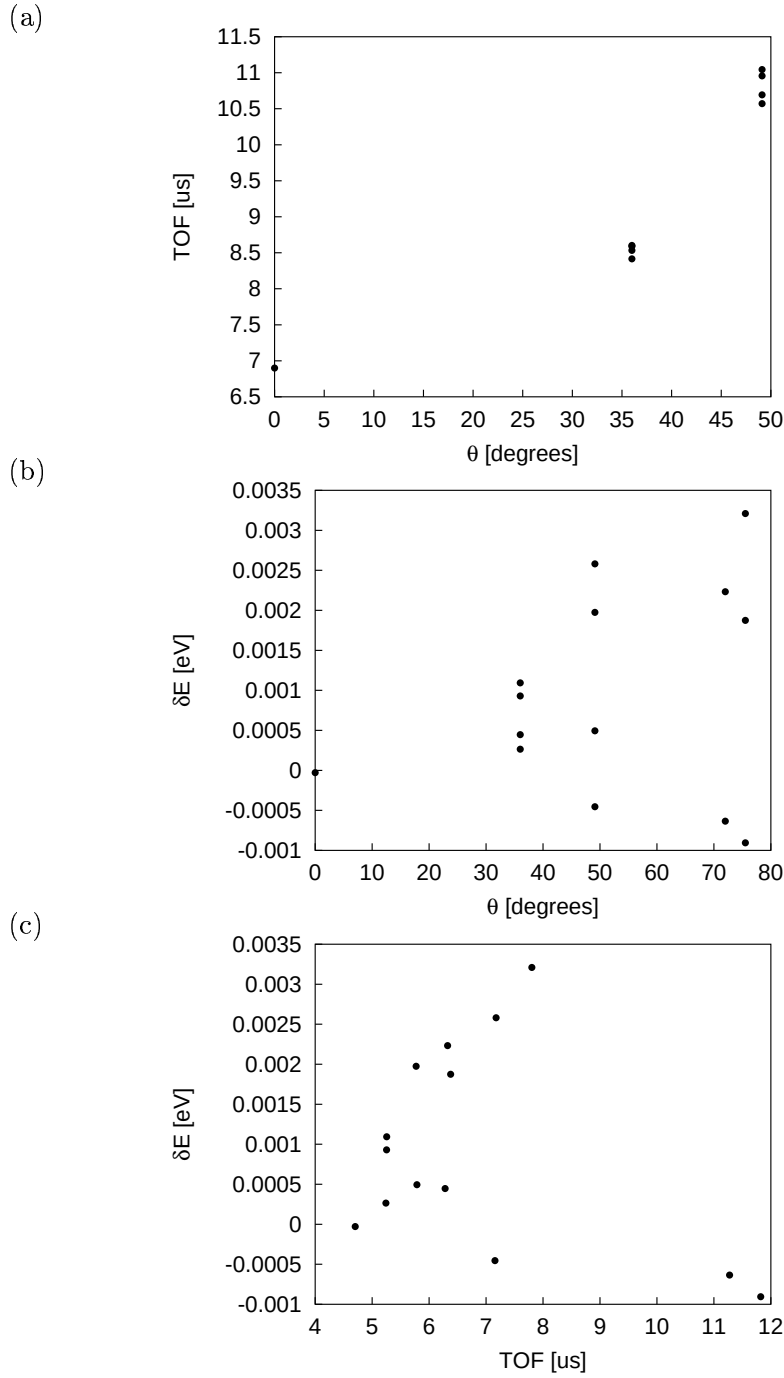


Abbildung 5.3: Das Einelektroden-Design. Dargestellt sind die Ergebnisse der Untersuchungen zur numerischen Genauigkeit der Simulationen des Hauptspektrometers. Das betrachtete Elektron hatte die Energie $U_0 + 1\text{eV}$ und einen Startradius von -32 mm . Abb. (a) zeigt die Flugzeit des Elektrons in Abhängigkeit vom Startwinkel. In (b) ist die Energieverletzung δE gegen den Startwinkel aufgetragen. (c) stellt die numerische Ungenauigkeit in Abhängigkeit der Flugzeit dar.

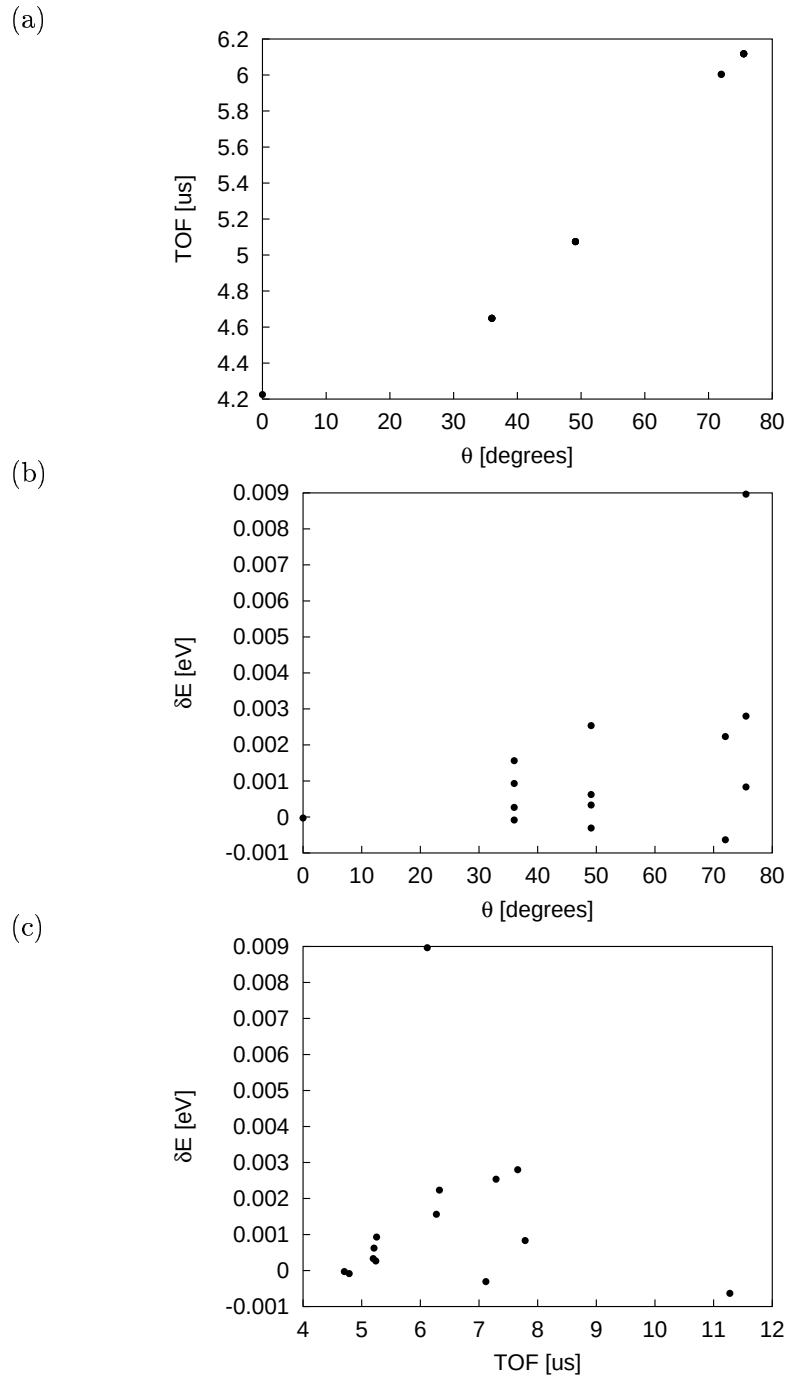


Abbildung 5.4: Das Einelektroden-Design. Die Abbildung entspricht Abb. (5.3) für den Startradius $r=0$ mm.

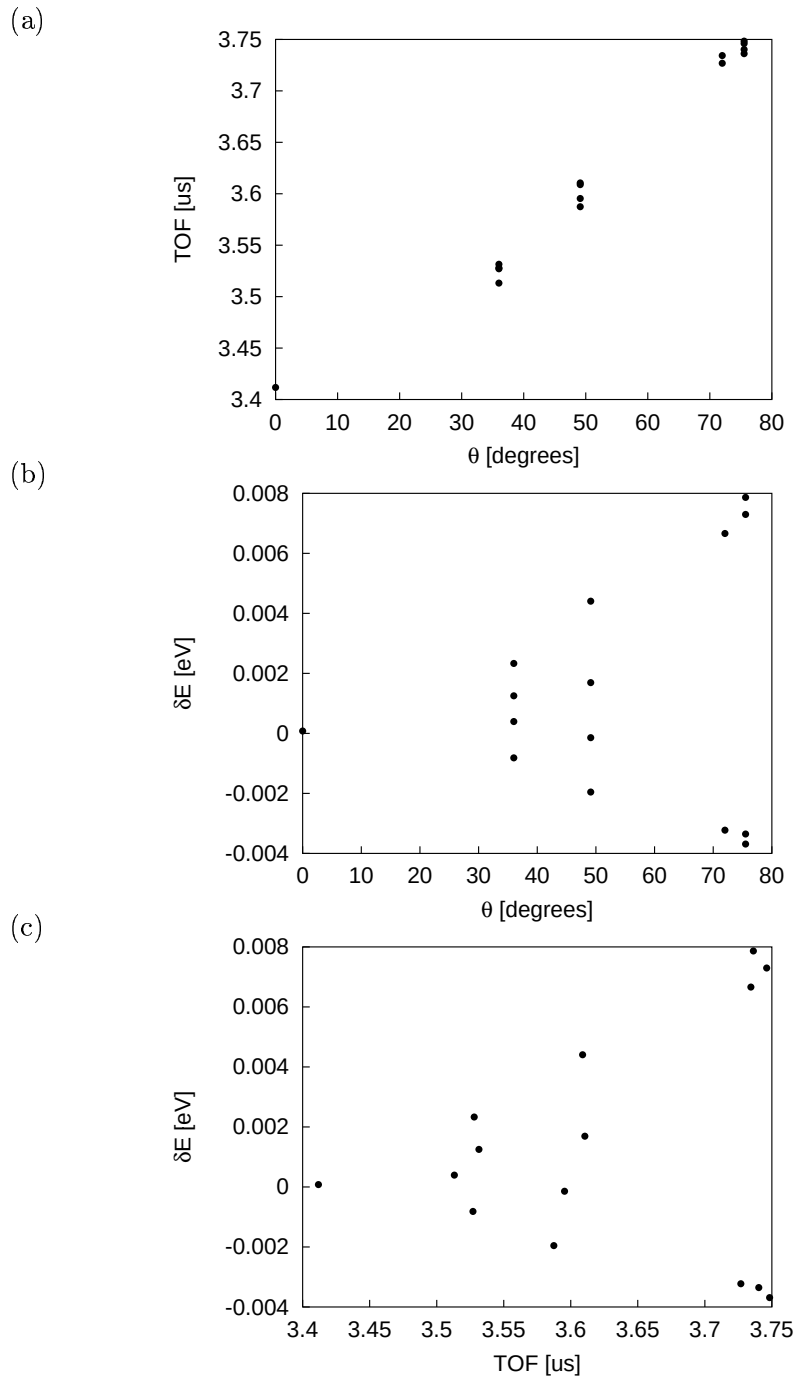


Abbildung 5.5: Das Einelektroden-Design. Die Abbildung entspricht Abb. (5.3) für eine Überschussenergie von 10 eV.

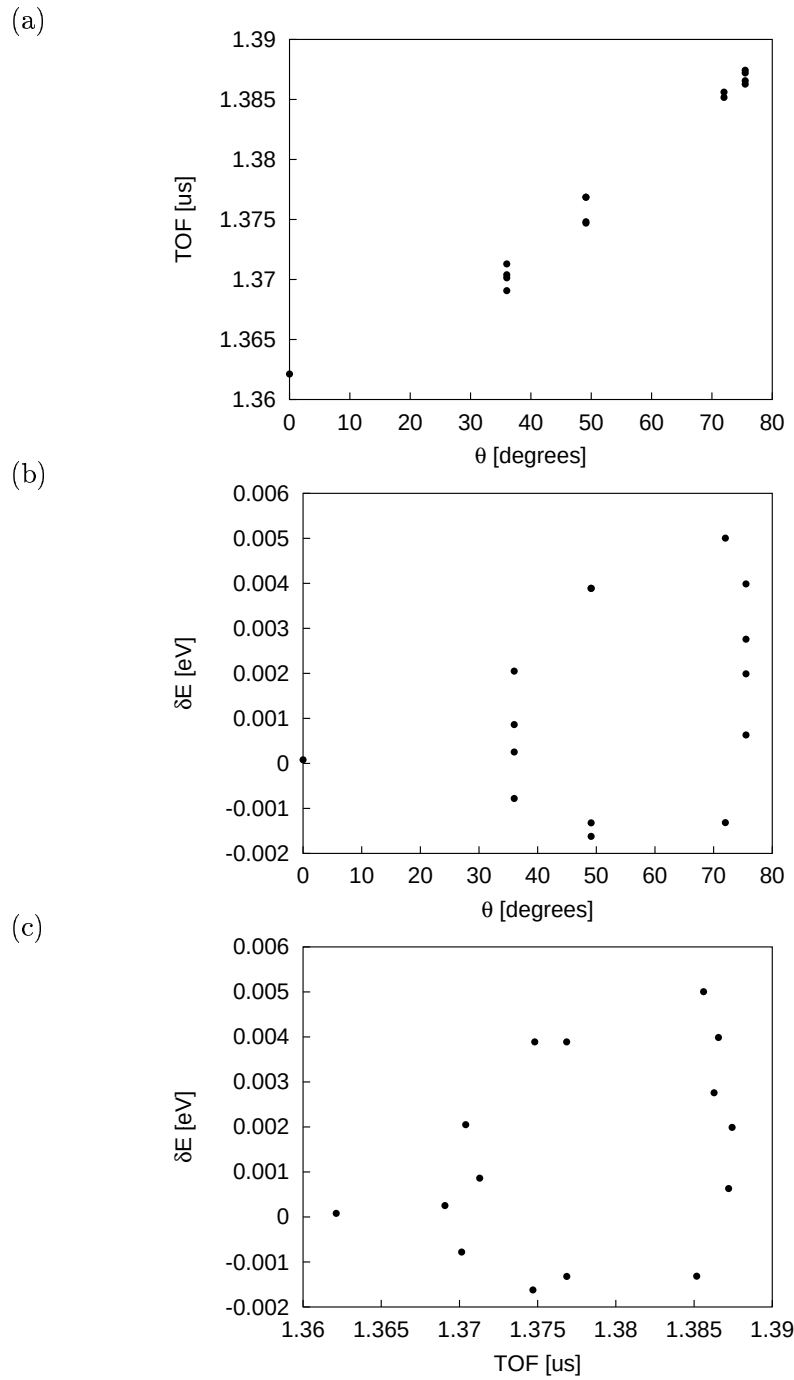


Abbildung 5.6: Das Einelektroden-Design. Die Abbildung entspricht Abb. (5.3) für eine Überschußenergie von 100 eV.

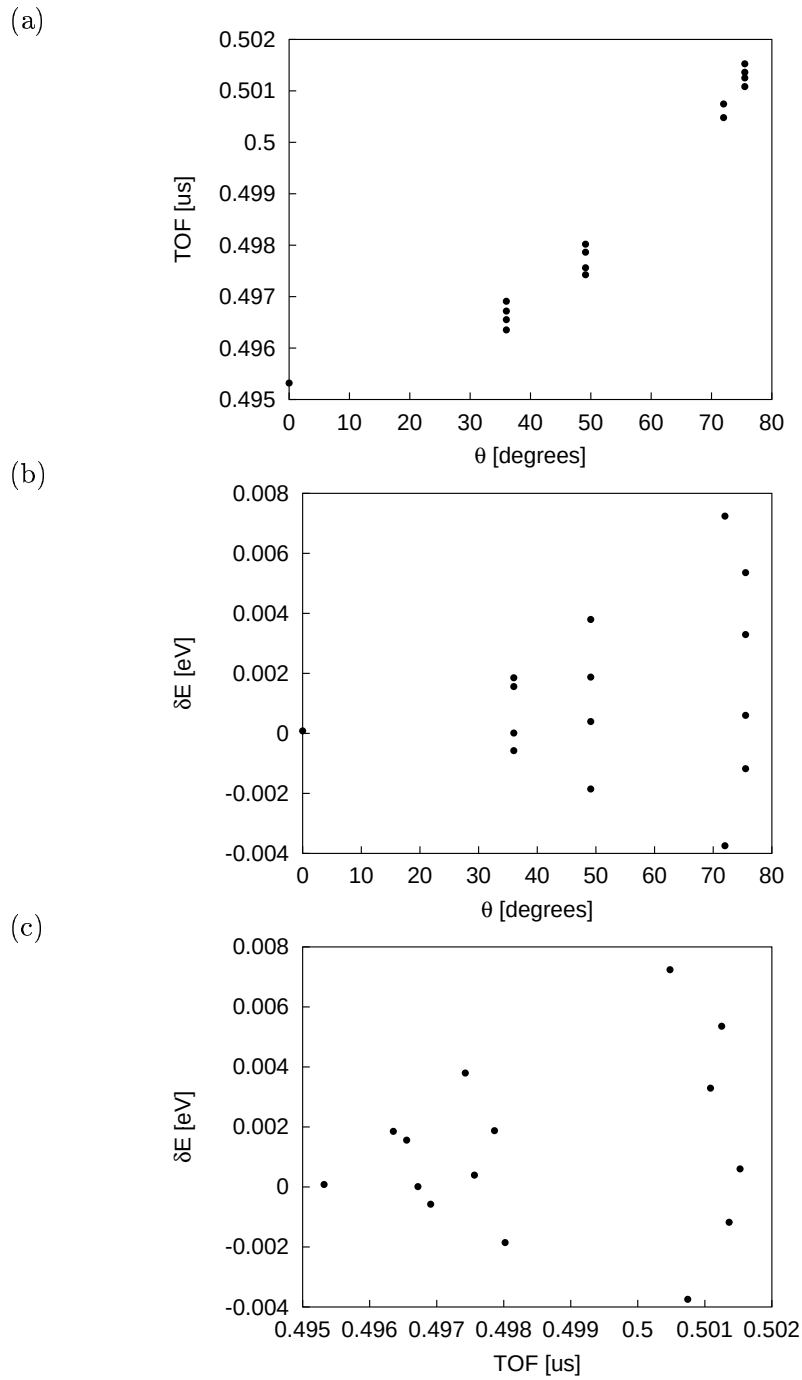


Abbildung 5.7: Das Einelektroden-Design. Die Abbildung entspricht Abb. (5.3) für eine Überschußenergie von 1 keV.

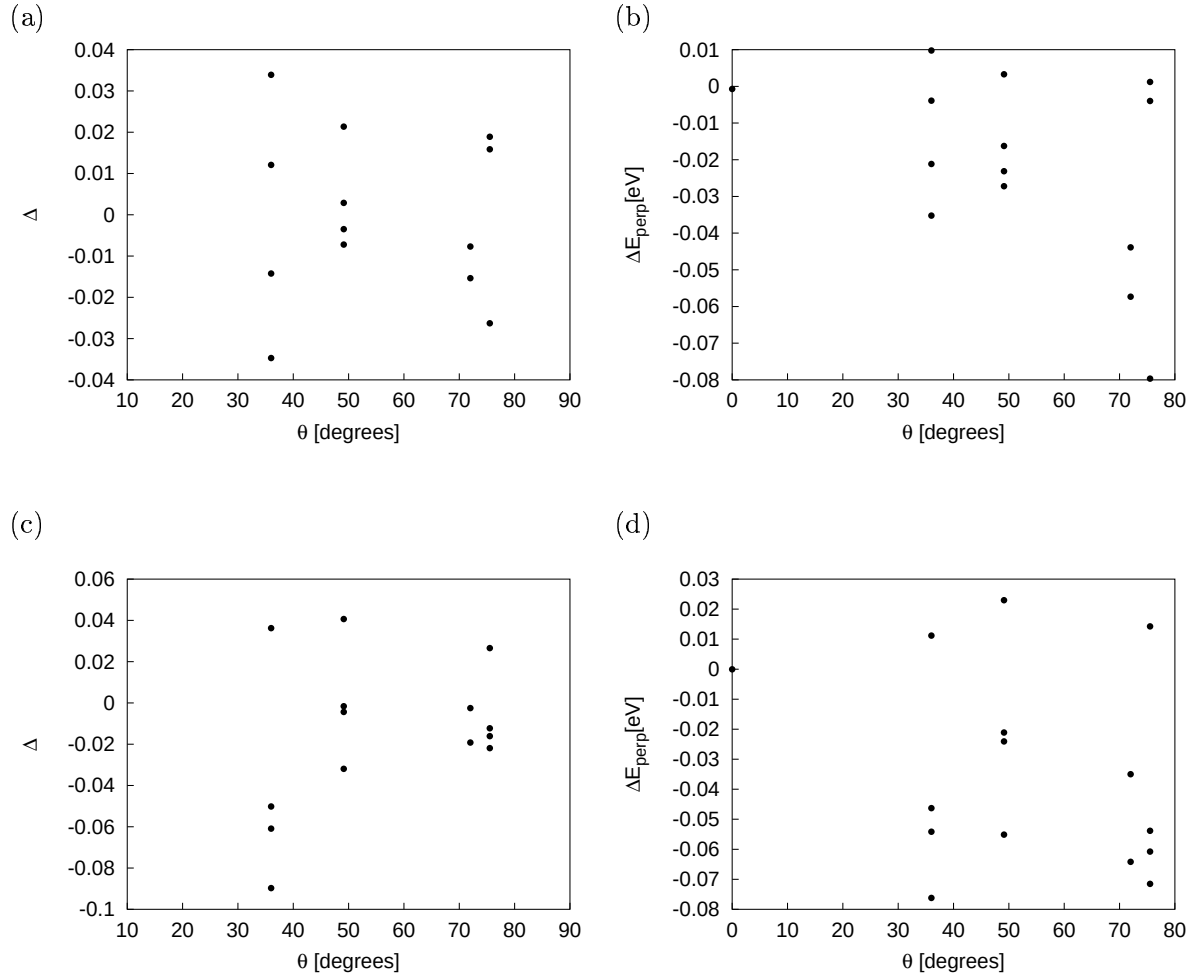


Abbildung 5.8: Das Einelektroden-Design. Abbildung (a) zeigt den Δ -Parameter für Elektronen mit einer Energie von $E = U_0 + 1\text{eV}$ und einem Startradius von -32mm in Abhängigkeit vom Startwinkel θ . In (b) ist die Verletzung der Transversalenergie dieser Elektronen dargestellt. Die Ergebnisse der Simulationen mit Überschußenergien von 10eV sind in (c) und (d) gezeigt.

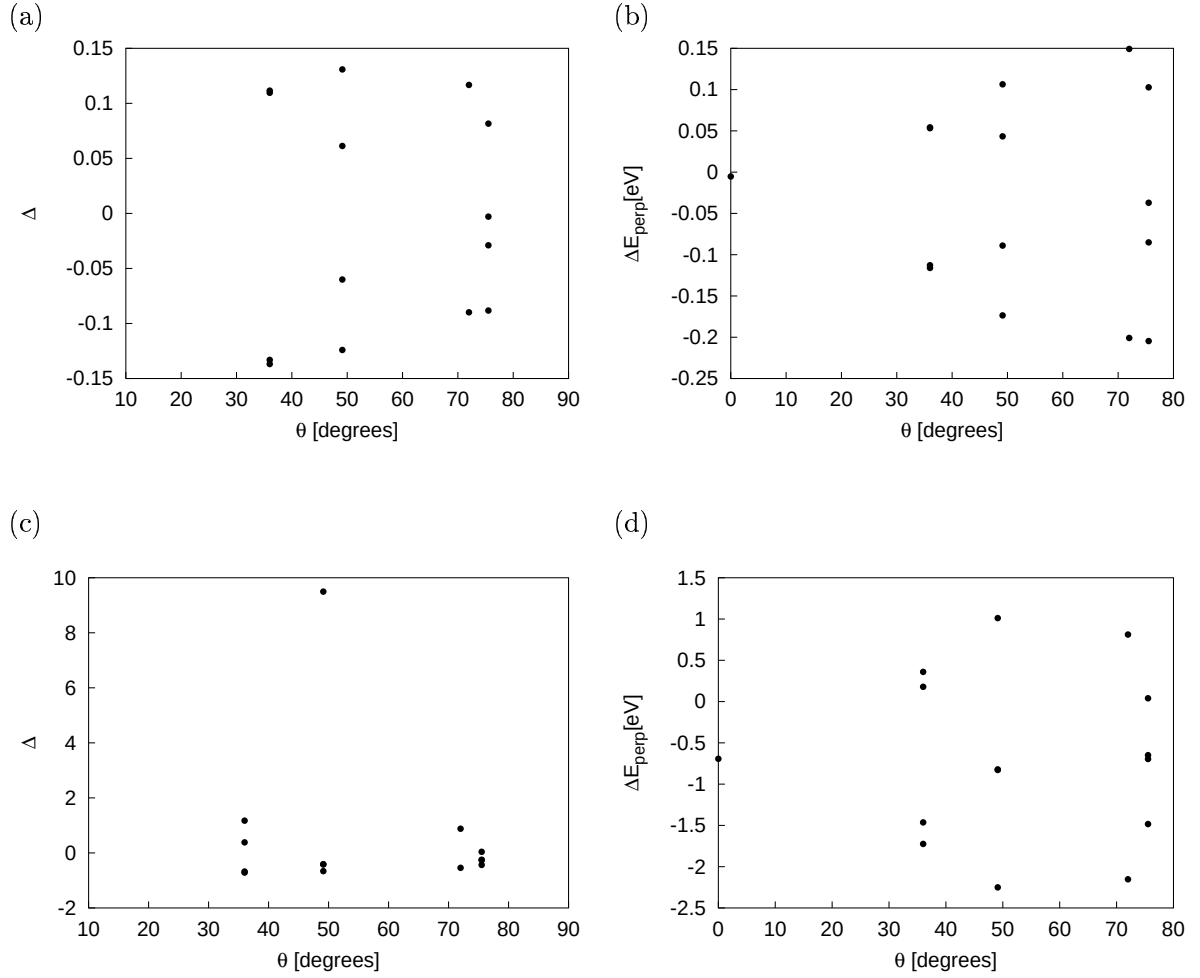


Abbildung 5.9: Das Einelektroden-Design. Abb. (a) und (b) zeigen den Δ -Parameter und die Verletzung der Transversalenergie für Elektronen mit der Überschußenergie von 100eV und dem Startradius von -32mm. (c) und (d) zeigen diese Parameter für Elektronen der Energie $E = U_0 + 1\text{keV}$.

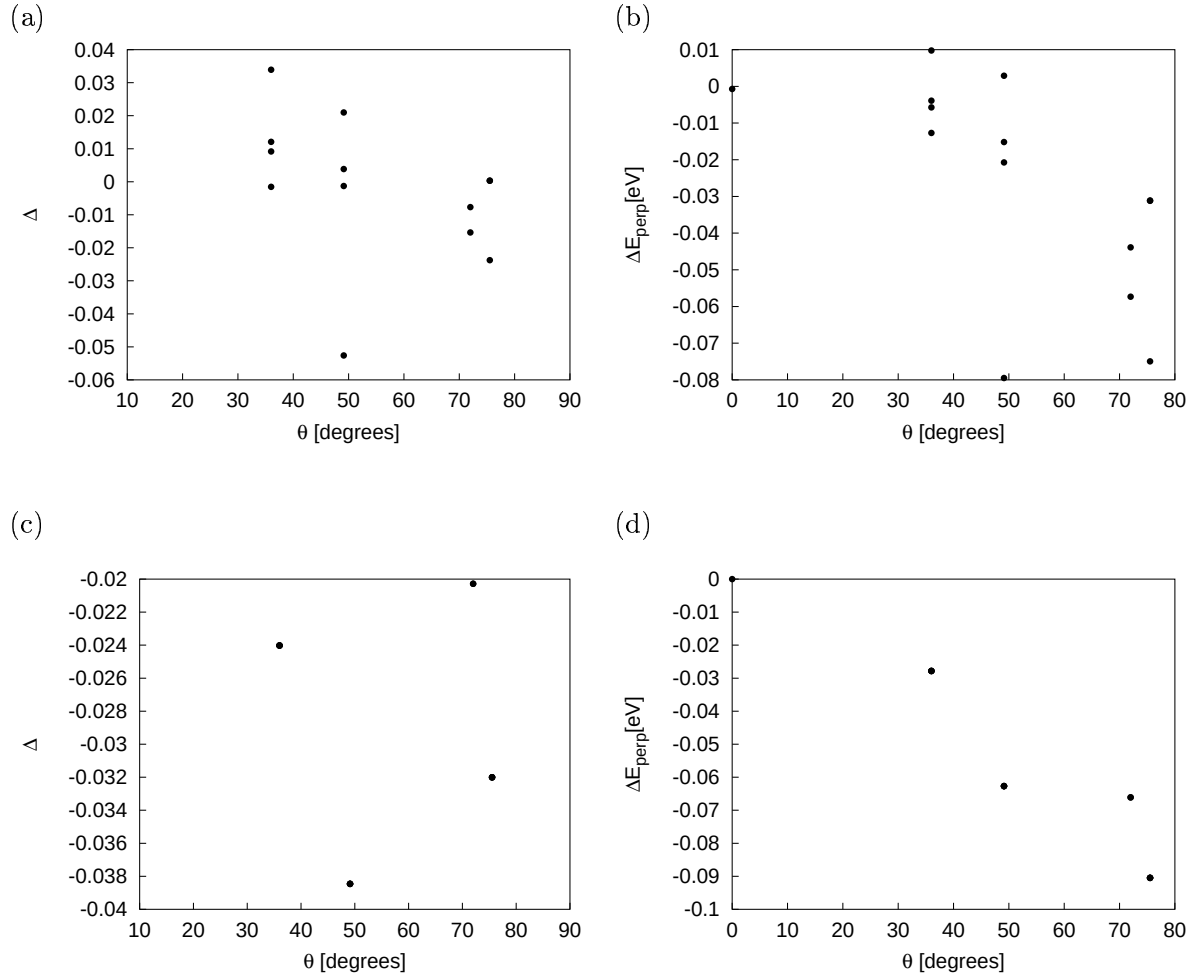


Abbildung 5.10: Das Einelektroden-Design. Abbildung (a) zeigt den Δ -Parameter für Elektronen mit einer Energie von $E = U_0 + 1\text{eV}$ und einem Startradius von 0 mm in Abhängigkeit vom Startwinkel θ . In (b) ist die Verletzung der Transversalenergie dieser Elektronen dargestellt. Die Ergebnisse der Simulationen mit Überschußenergien von 10eV sind in (c) und (d) gezeigt.

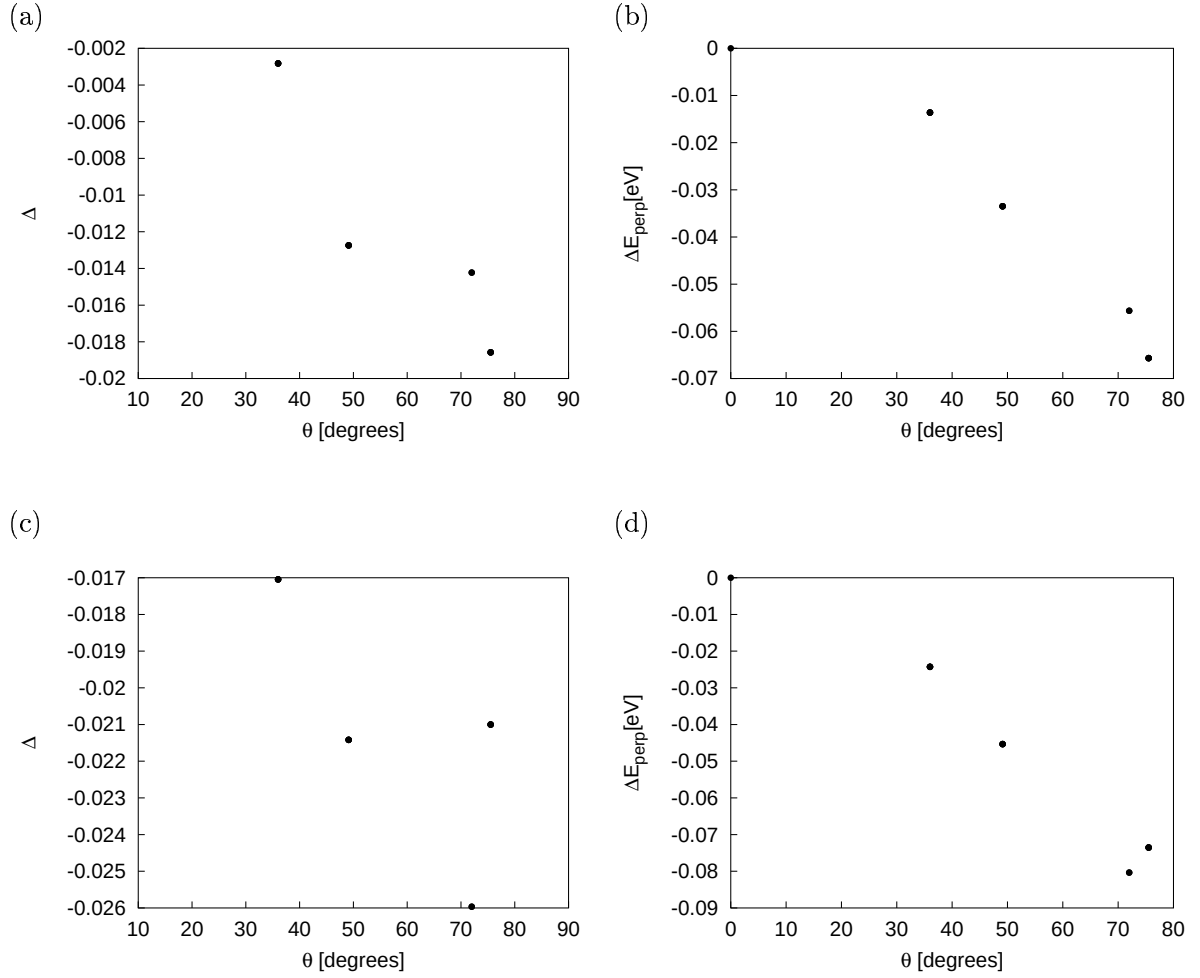


Abbildung 5.11: Das Einelektroden-Design. Abb. (a) und (b) zeigen den Δ -Parameter und die Verletzung der Transversalenergie für Elektronen mit der Überschußenergie von 100eV und dem Startradius von 0 mm. (c) und (d) zeigen diese Parameter für Elektronen der Energie $E = U_0 + 1\text{keV}$.

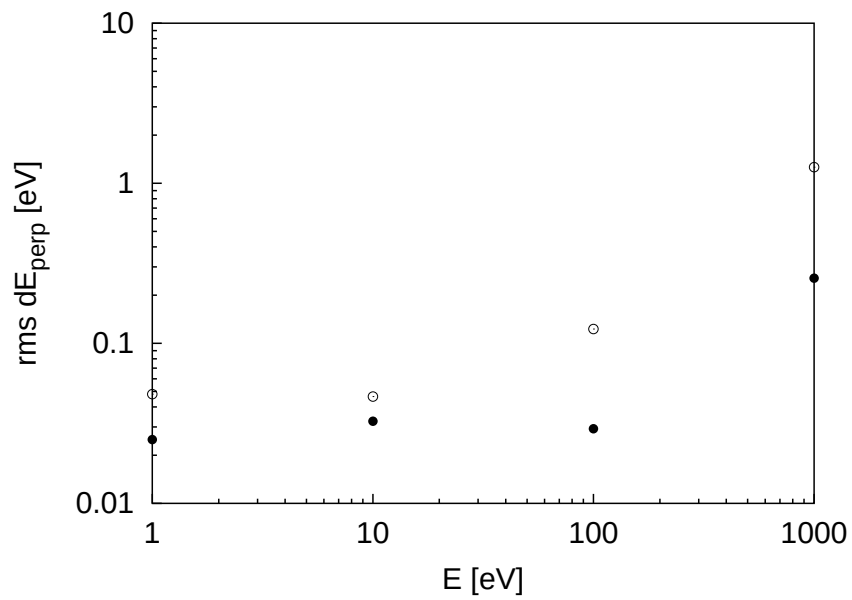
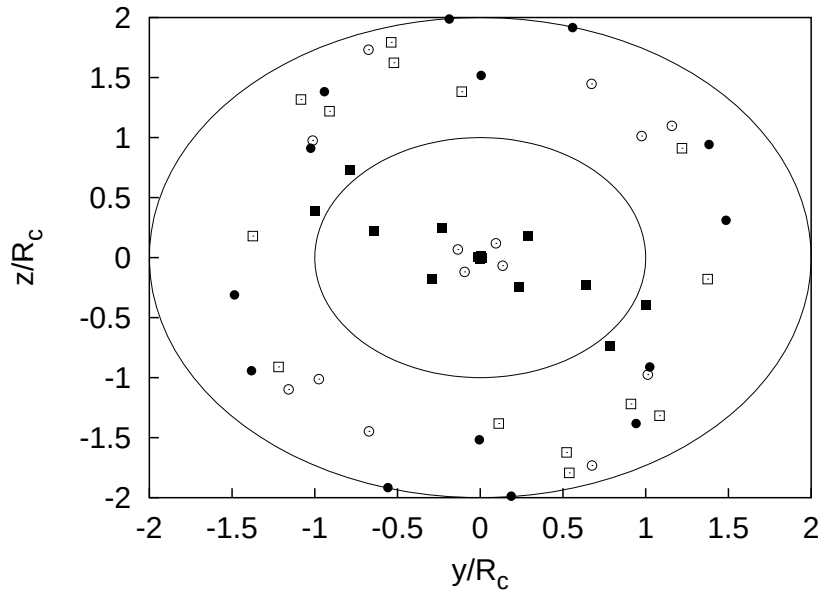


Abbildung 5.12: Der Vakuumtank auf HV. Es ist die Varianz der $\delta E_{\perp}$ gegen die Überschußenergie aufgetragen. Die vollen Kreise beschreiben auf der Achse fliegende Elektronen ($r = 32$ mm), während die Kreise Elektronen auf der äußeren Flugbahn zeigen.

(a)



(b)

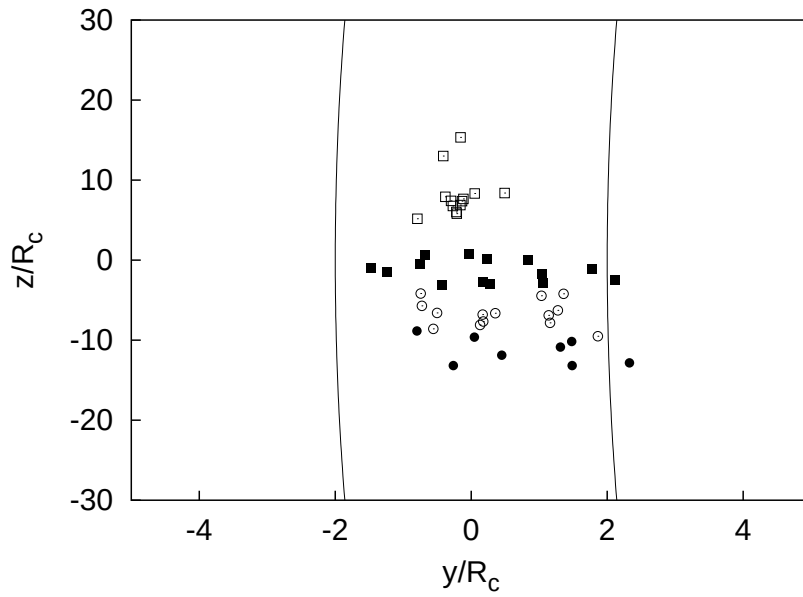


Abbildung 5.13: Das Einelektroden-Design. Es sind die Abweichungen der Teilchenposition von der erwarteten Position in der Analysierebene in Vielfachen des Zyklotronradius dargestellt. Die Plots zeigen die Ergebnisse für auf der Achse fliegende Elektronen ($r=0\text{mm}$) (a) und achsenferne Elektronen ($r=32\text{ mm}$) (b). Die Überschußenergien betragen 1eV (volle Kreise), 10eV (Kreise), 100eV (volle Quadrate) und 1keV (Quadrate).

5.2 Vakuumtank auf Potential

Die konsequente Verfolgung eines möglichst einfachen Aufbau führt dazu, daß man anstatt einer Elektrode innerhalb des Vakuumtanks sofort die Tankhülle als Elektrode verwendet. Auf diese Weise wird die Fläche im UHV-Bereich stark reduziert (um etwa $\frac{2}{3}$) und die technische Komplexität verringert sich, da man kein Aufhängungssystem für die Elektroden benötigt. Weiterhin wird die nutzbare Analysierebene um etwa 10% größer. Diese Zahl ergibt sich, wenn der Flußschlauch bis zu 10 cm an die Elektrode herangeführt wird, und die Elektrode in Einelektroden design selbst etwa 10 cm von der Tankwand entfernt ist. In Abb. (5.14) ist ein Entwurf dieser Idee gezeigt.

Das Potential von $U_0 = -18600$ V wird direkt am Vakuumtank angelegt. Im Quellbereich ist eine Blende vorgesehen, die ein Potential von 0 V trägt. Ein detaillierter Plan dieses Aufbaus ist Abb. (5.16) zu entnehmen. Auf diese Weise wird ein Potentialdurchgriff von 0.41 V erreicht (s. Abb. (5.15)). Der Verlauf des Potentials in der Analysierebene ist in Abb. (5.17) dargestellt. Hier ist liegt ein ähnlicher Verlauf wie beim Einelektroden design vor.

Eine Umsetzung dieses Entwurfes wird mit einem zusätzlichen Abschirmungspotential $U_0 + \epsilon$ realisiert werden, da ein sofortiger Übergang auf das Erdpotential zu Spannungsinstabilitäten führen würde.

Auch für dieses System wurden die numerische Genauigkeit, die Verletzung der Adiabasie und die Abbildungseigenschaften untersucht.

Die Untersuchung der Numerik ergab eine Verletzung der Gesamtenergie, die in der selben Größenordnung liegt, wie bei der Untersuchung für das Einelektroden system. Es ist also möglich die Ergebnisse direkt miteinander zu vergleichen. Die Ergebnisse sind in den Abb. (5.18-5.22) dargestellt.

Auch die Untersuchung der Adiabasierletzung zeigt ein ähnliches Verhalten für dieses Spektrometer design, so wird in den Abbildungen (5.23 - 5.26) ebenfalls deutlich, daß die Verletzung der Adiabasie für große Startradien größer ist, als für auf der Achse fliegende Elektronen. Ebenso ist zu erkennen, daß die Abweichung vom adiabatischen Verhalten für Elektronen höherer Energien größer ist, als für Elektronen geringerer Überschußenergie. Auch für diesen Fall ist in Abb. (5.27) die Varianz der δE_{perp} gegen die Überschußenergie aufgetragen.

In den Untersuchungen zu den Abbildungseigenschaften (Abb.(5.28-??)) konnte ebenfalls das bereits in den Untersuchungen zum Einelektroden system festgestellte Verhalten reproduziert werden. So treten auch hier für auf der Achse fliegende Elektronen keine Driften auf, während diese bei großen Startradien deutlich zu erkennen sind.

Insgesamt ist aber eine Punkt -zu-Kreis-Abbildung festzustellen.

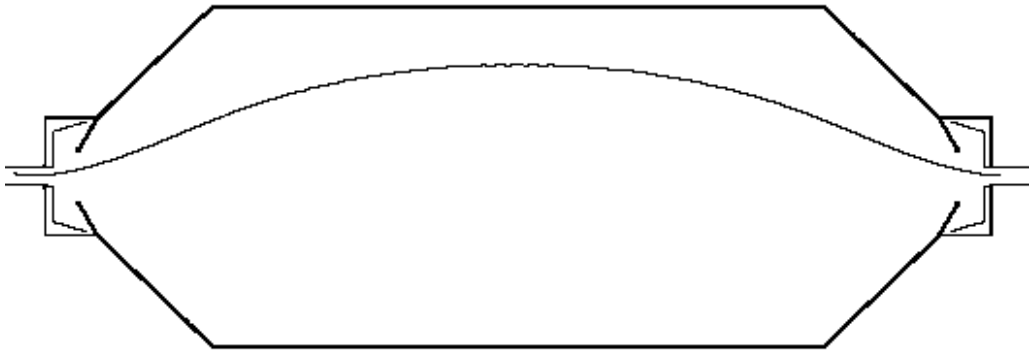


Abbildung 5.14: Vakuumtank auf HVE Ein Entwurf des Vakuumtanks auf Hochpotential mit $U_0 = -18600$ V.

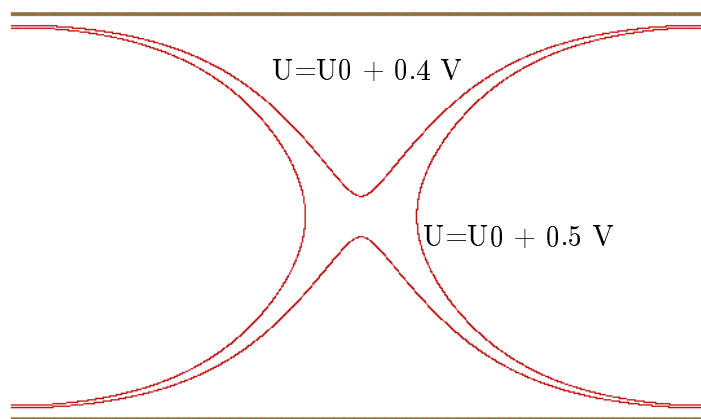


Abbildung 5.15: Potentialdurchgriff des Tanks auf HV Die dargestellten Äquipotentialebenen liegen 0.4, bzw. 0.5 V über den Retardierungspotential. Der Potentialdurchgriff dieses Systems beträgt 0.41 V.

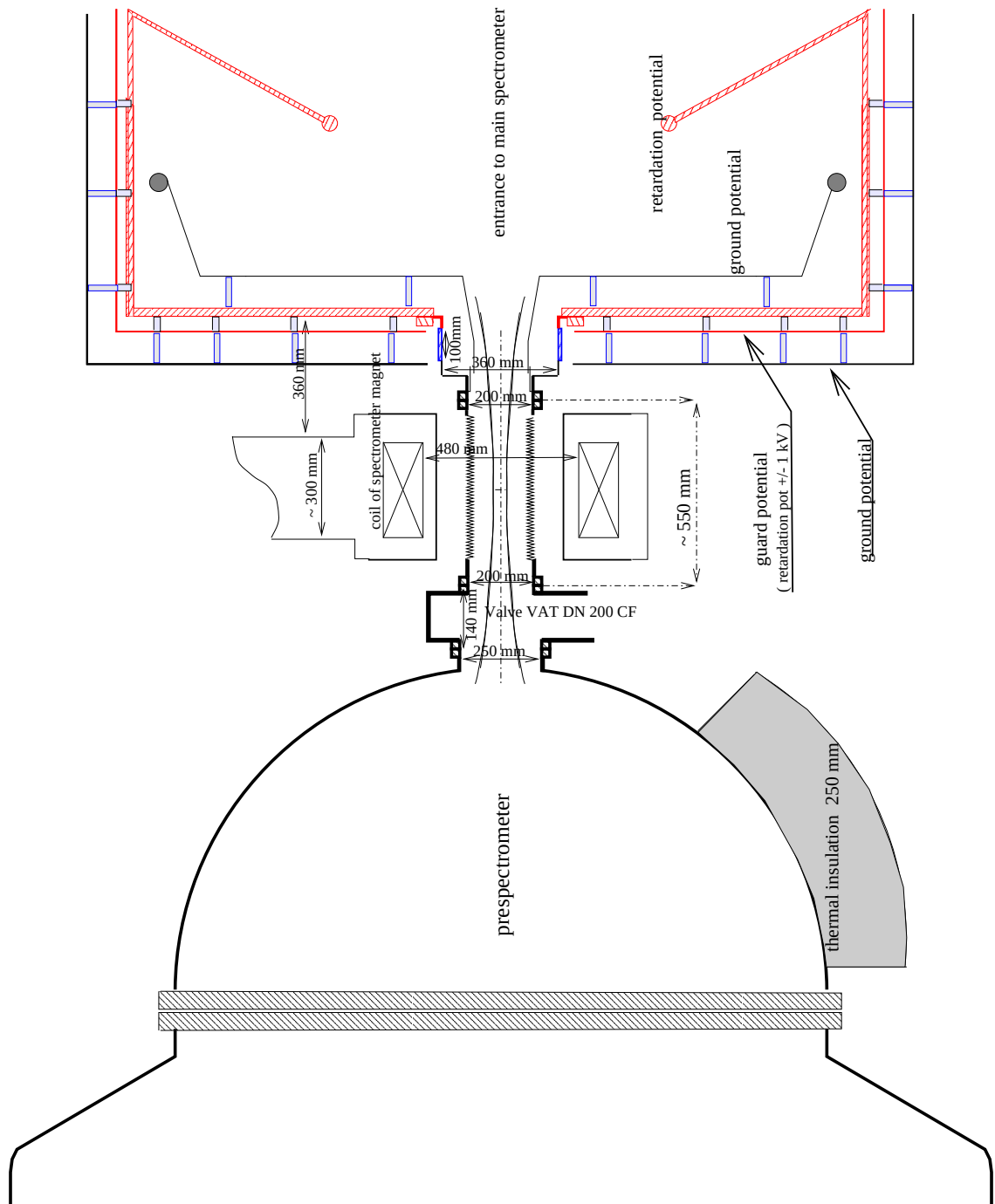


Abbildung 5.16: Übergang zwischen Vor- und Hauptspektrometer. Es ist der Aufbau zwischen dem Vor- und dem Hauptspektrometer gezeigt. Neben den Vakuum-tanks ist ein Spektrometersolenoid, sowie ein Ausschnitt des magnetischen Flußschlauches, und das UHV-Ventil zum Abschließen des Spektrometers dargestellt.

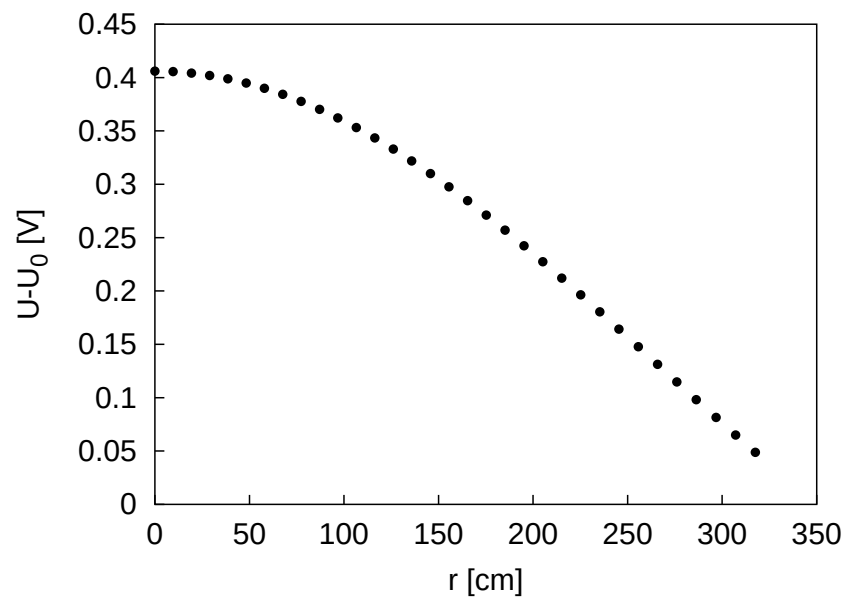


Abbildung 5.17: Vakuumtank auf HV. Dargestellt ist der Verlauf des elektrischen Potentials U entlang der Analysierebene.

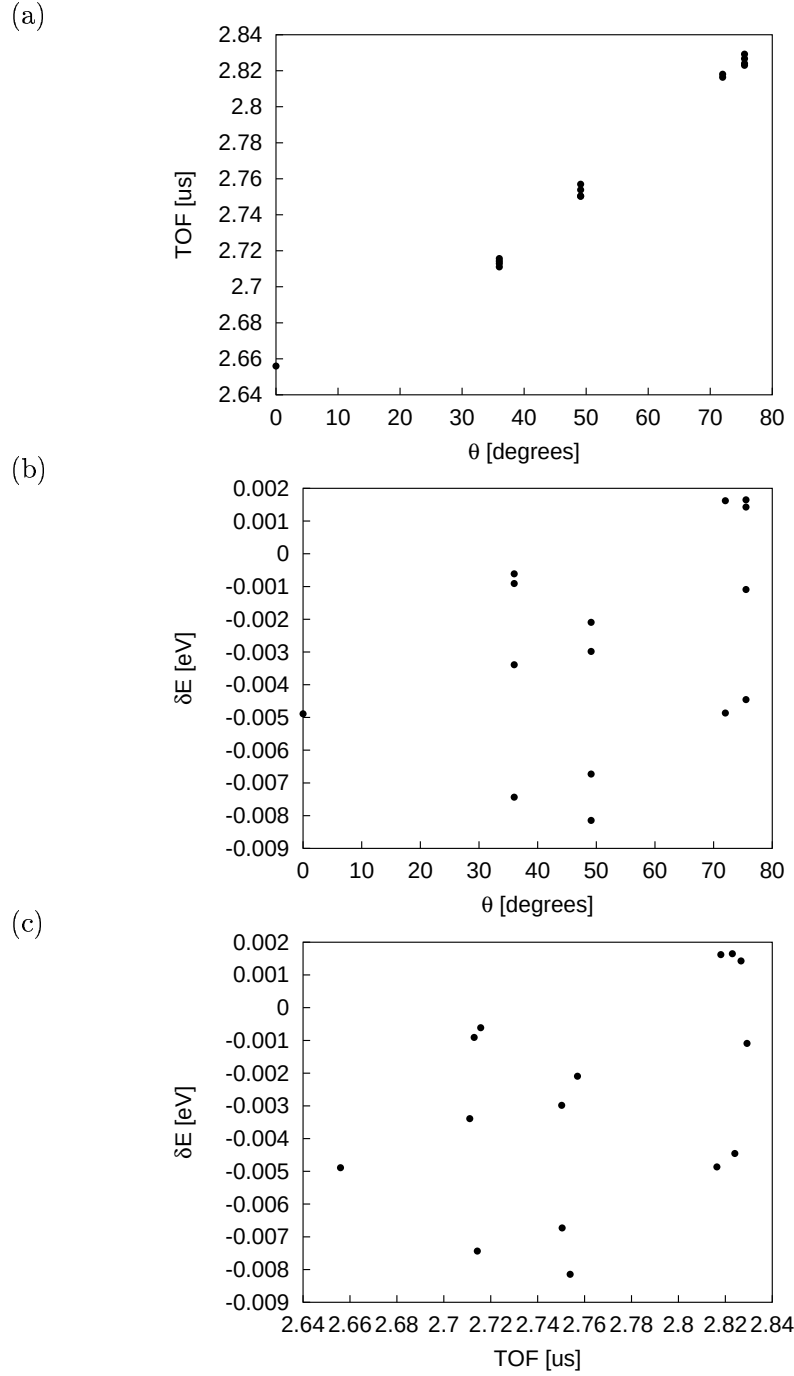


Abbildung 5.18: Das Einelektroden-Design. Dargestellt sind die Ergebnisse der Untersuchungen zur numerischen Genauigkeit der Simulationen des Hauptspektrometers. Das betrachtete Elektron hatte die Energie $U_0 + 1\text{eV}$ und einen Startradius von -32 mm . Abb. (a) zeigt die Flugzeit des Elektrons in Abhängigkeit vom Startwinkel. In (b) ist die Energieverletzung δE gegen den Startwinkel aufgetragen. (c) stellt die numerische Ungenauigkeit in Abhängigkeit der Flugzeit dar.

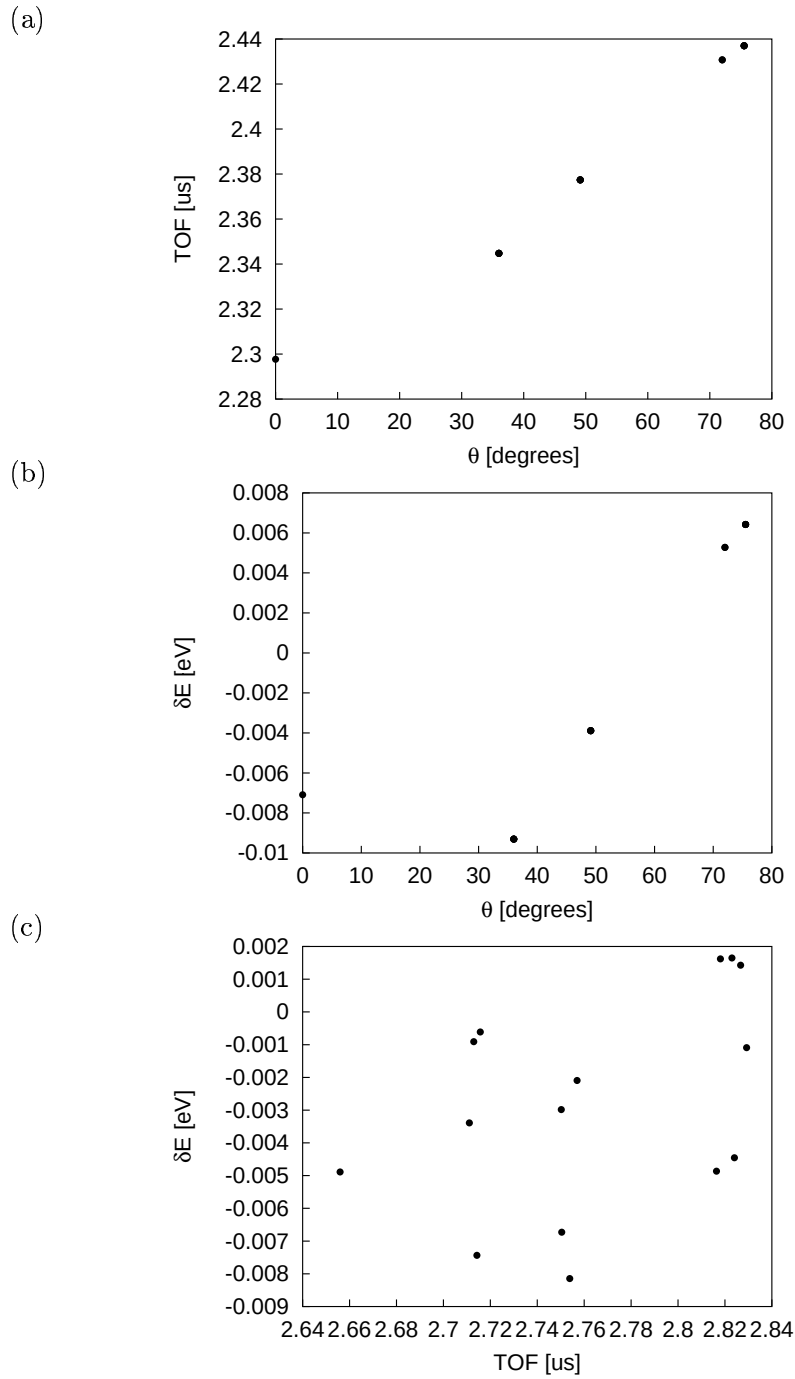


Abbildung 5.19: Der Vakuumtank auf HV. Die Abbildung entspricht Abb. (5.18) für den Startradius $r=0$ mm.

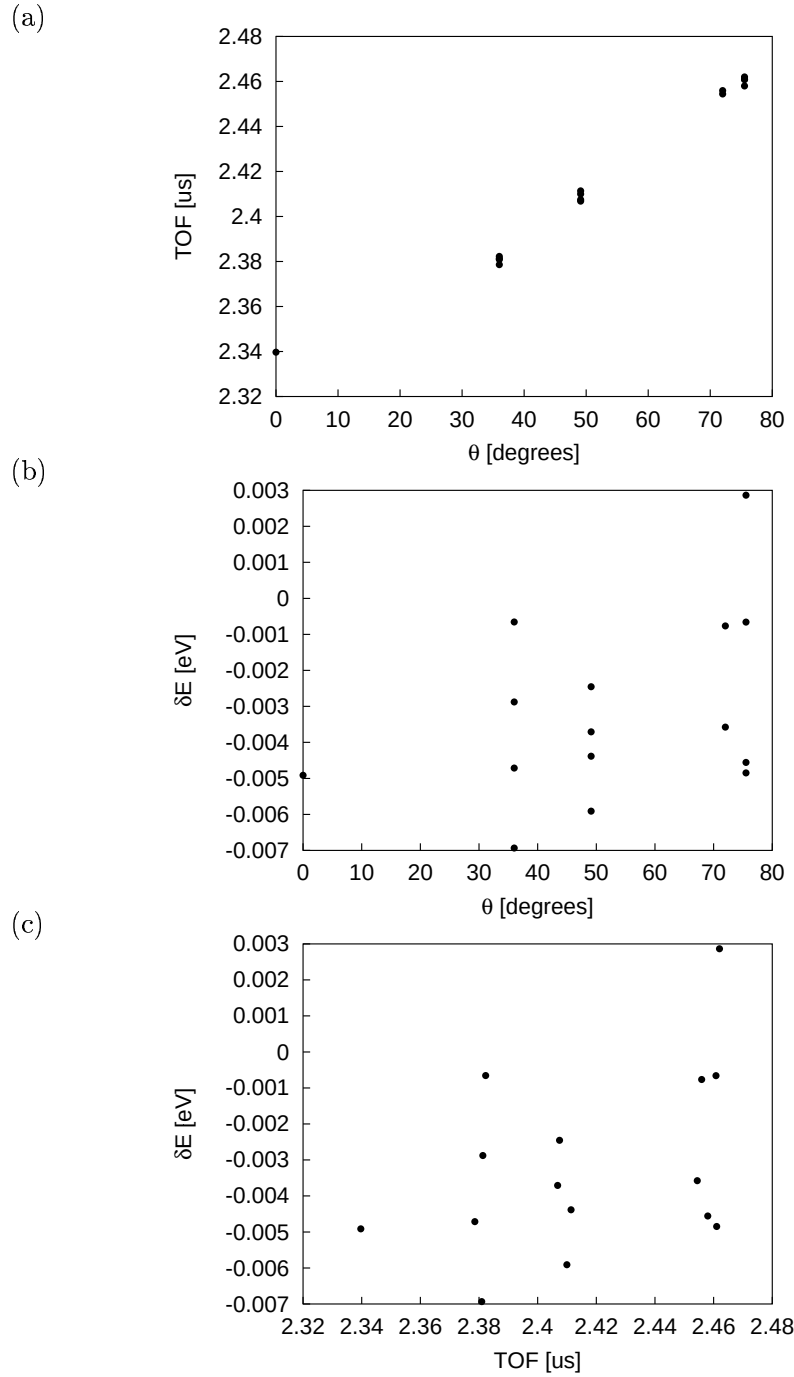


Abbildung 5.20: Der Vakuumtank auf HV. Die Abbildung entspricht Abb. (5.18) für eine Überschußenergie von 10 eV.

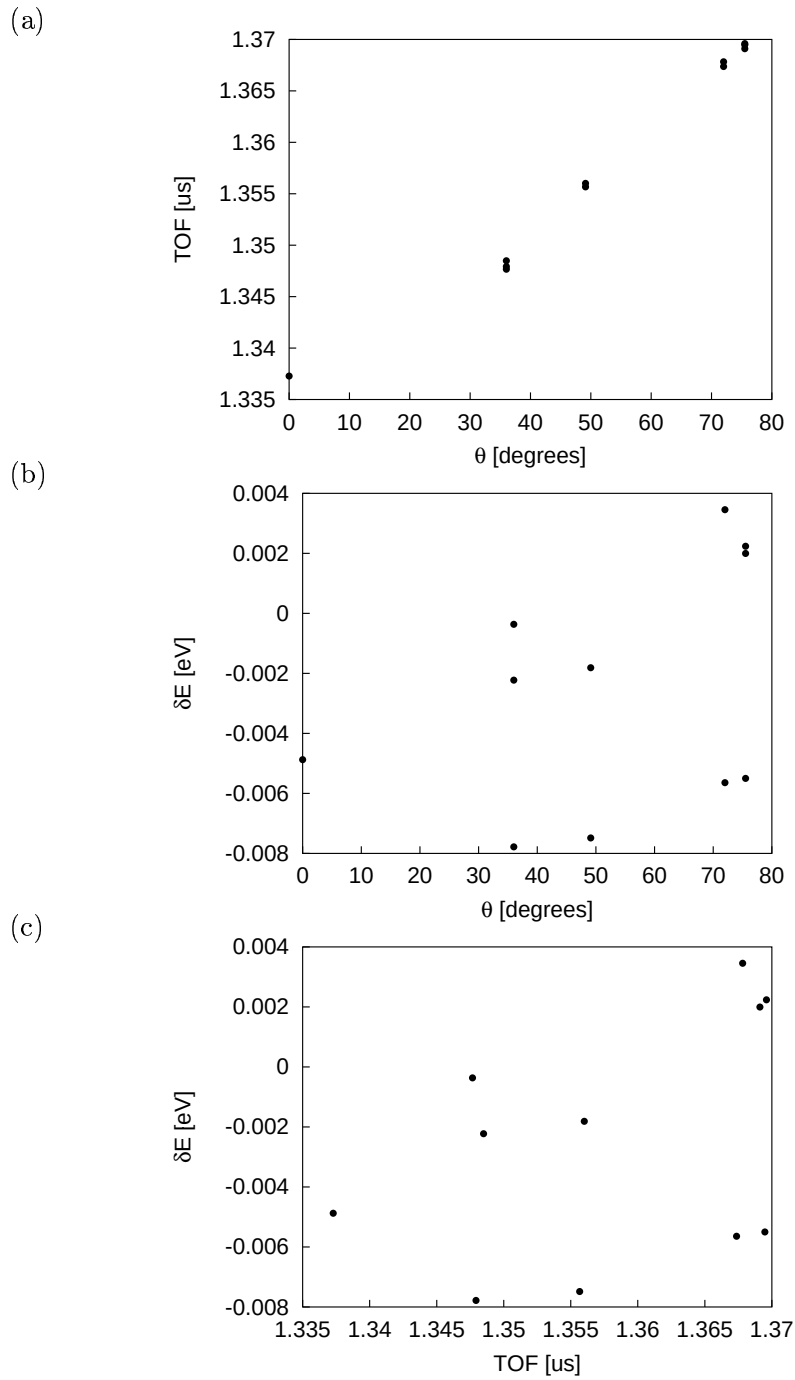
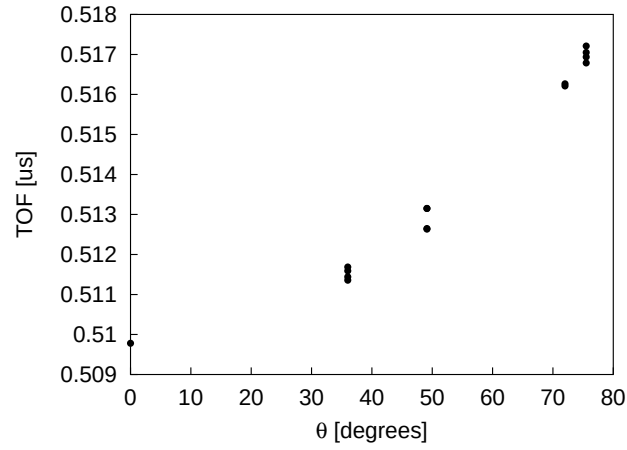
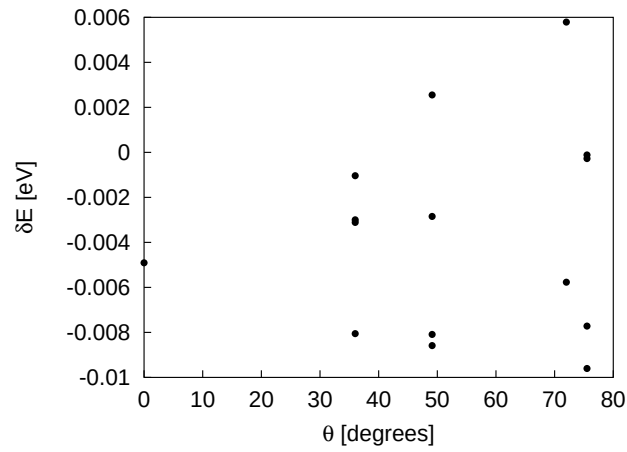


Abbildung 5.21: Der Vakuumtank auf HV. Die Abbildung entspricht Abb. (5.18) für eine Überschußenergie von 100 eV.

(a)



(b)



(c)

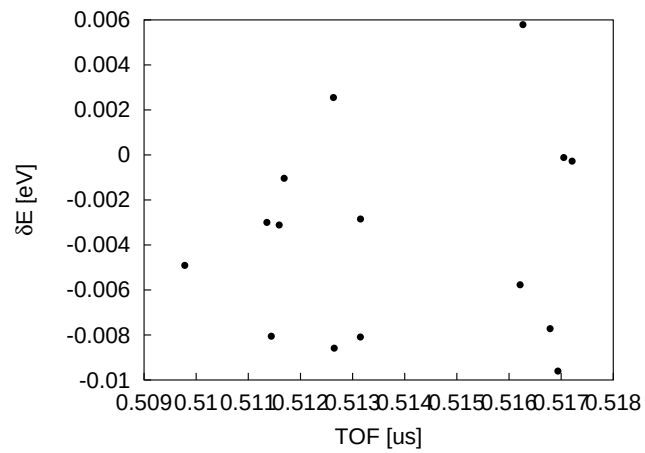


Abbildung 5.22: Der Vakuumtank auf HV. Die Abbildung entspricht Abb. (5.18) für eine Überschußenergie von 1000 eV.

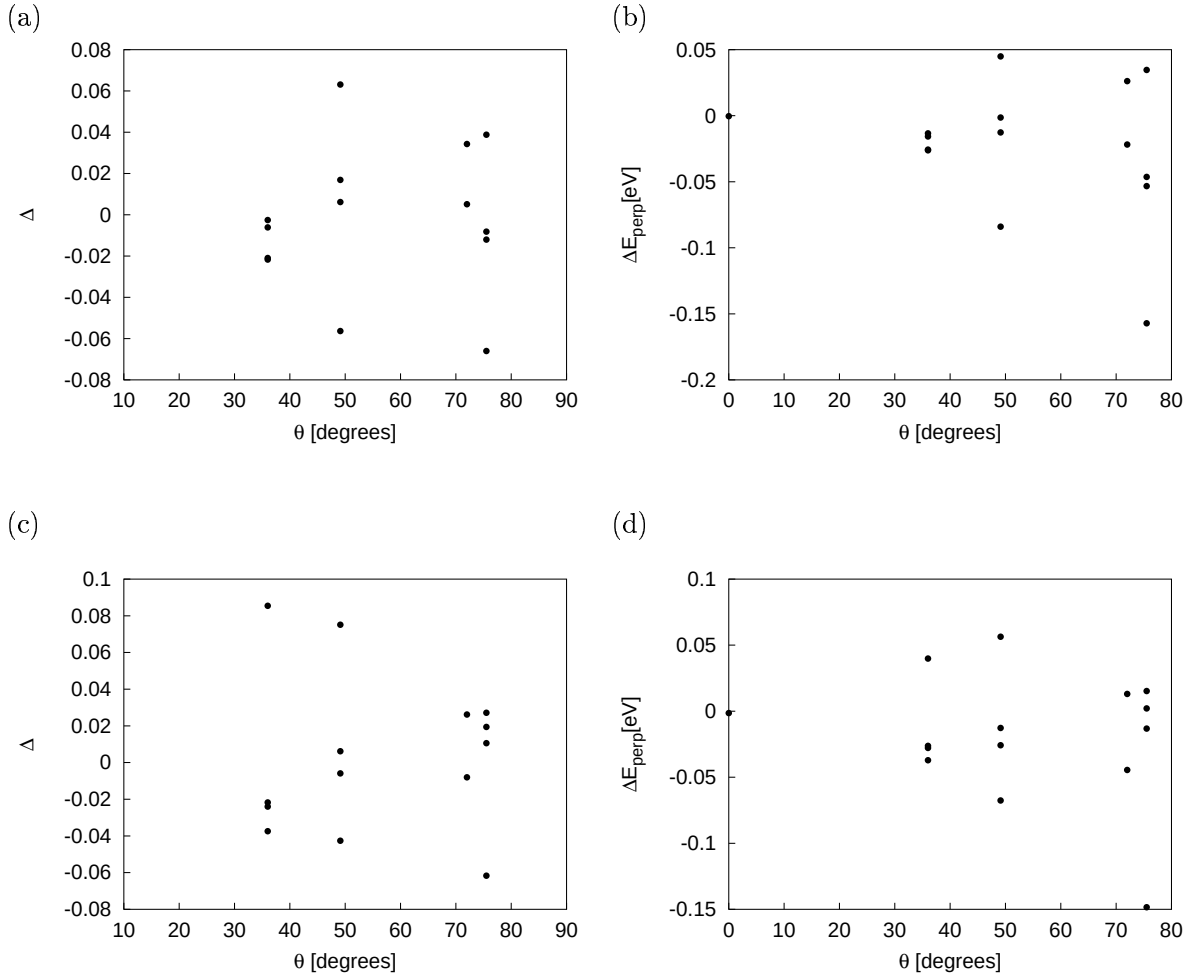


Abbildung 5.23: Der Vakuumtank auf HV. Abbildung (a) zeigt den Δ -Parameter für Elektronen mit einer Energie von $E = U_0 + 1\text{eV}$ und einem Startradius von -32 mm in Abhängigkeit vom Startwinkel θ . In (b) ist die Verletzung der Transversalenergie dieser Elektronen dargestellt. Die Ergebnisse der Simulationen mit Überschußenergien von 10eV sind in (c) und (d) gezeigt.

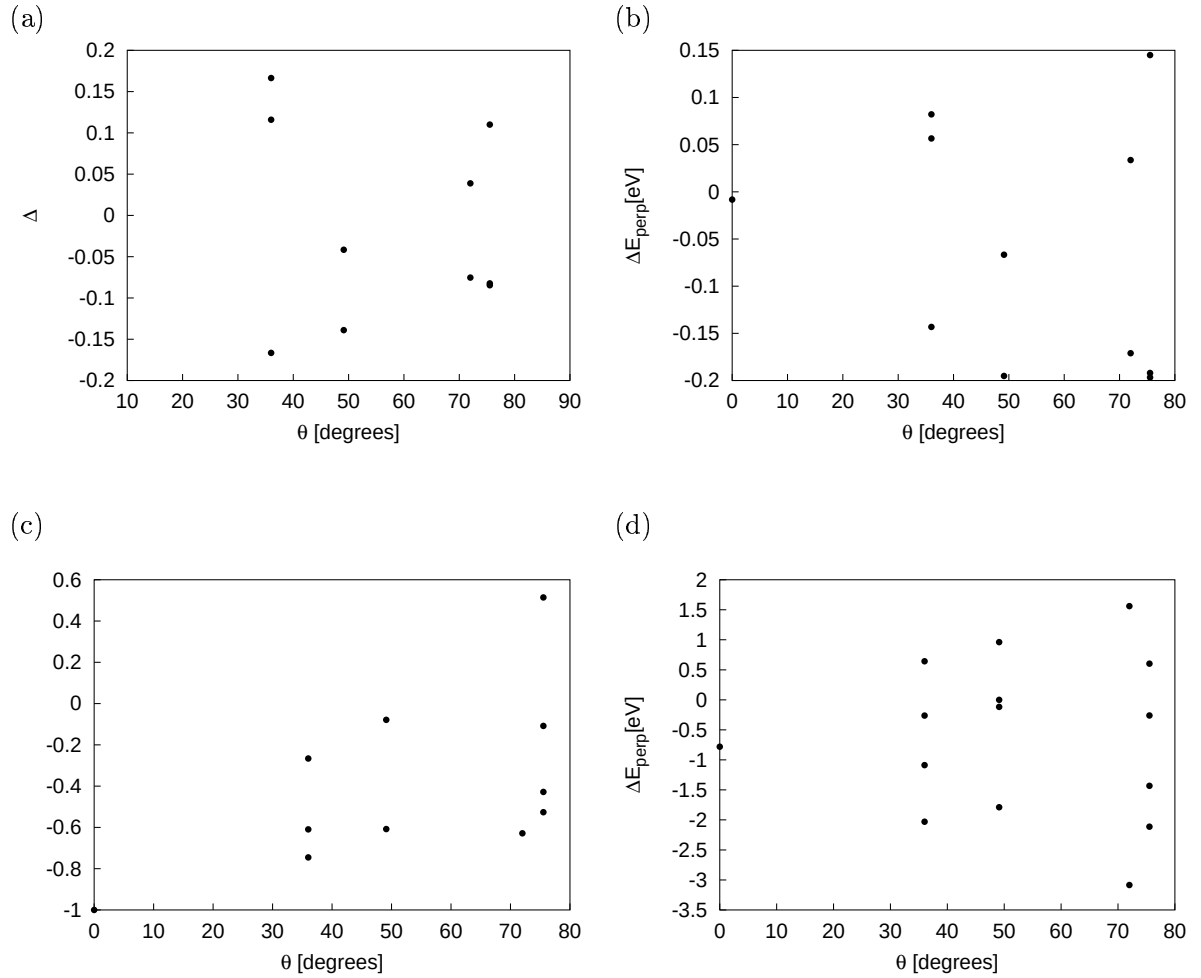


Abbildung 5.24: Der Vakuumtank auf HV. Abb. (a) und (b) zeigen den Δ -Parameter und die Verletzung der Transversalenergie für Elektronen mit der Überschußenergie von 100eV und dem Startradius von -32mm. (c) und (d) zeigen diese Parameter für Elektronen der Energie $E = U_0 + 1\text{keV}$.

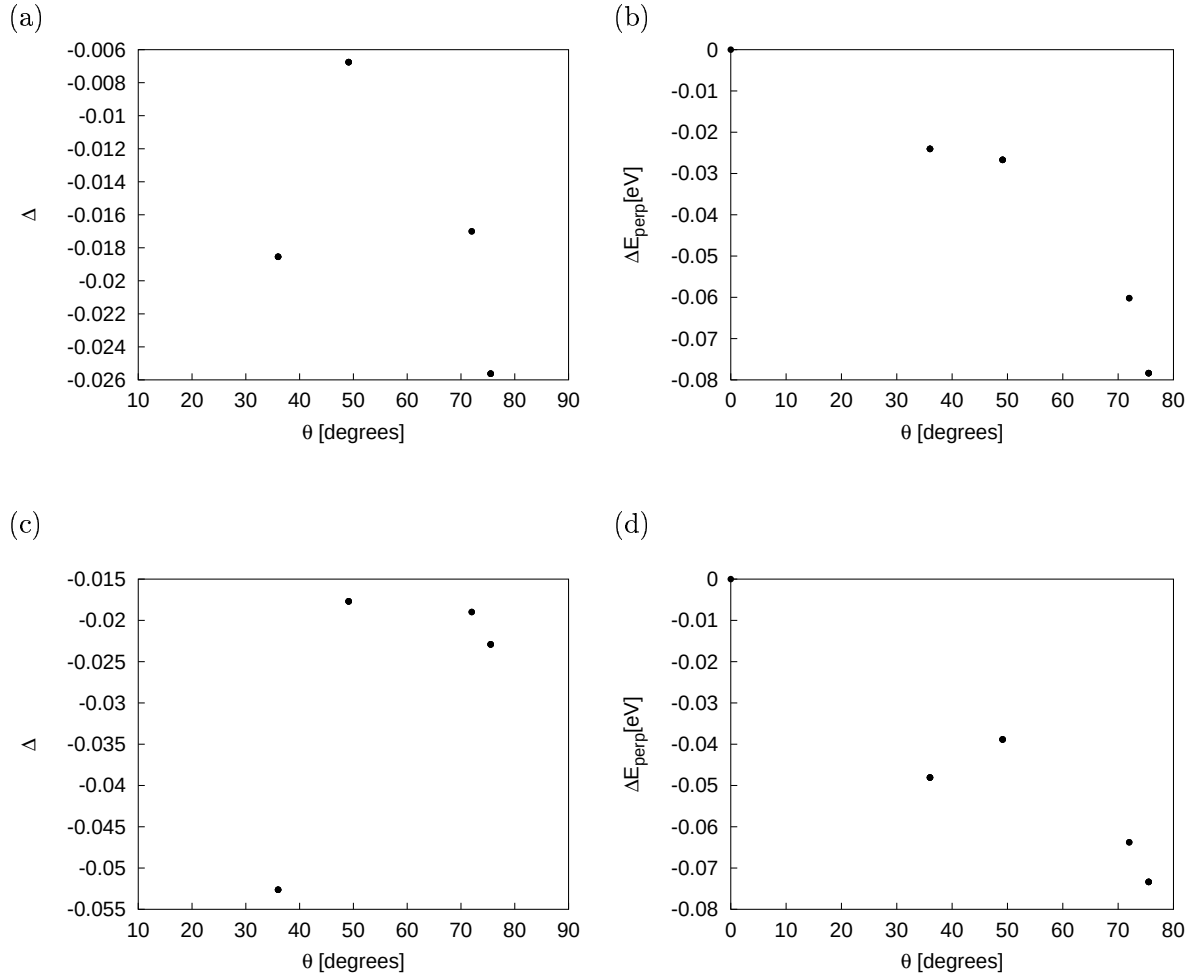


Abbildung 5.25: Der Vakuumtank auf HV. Abbildung (a) zeigt den Δ -Parameter für Elektronen mit einer Energie von $E = U_0 + 1\text{eV}$ und einem Startradius von 0 mm in Abhängigkeit vom Startwinkel θ . In (b) ist die Verletzung der Transversalenergie dieser Elektronen dargestellt. Die Ergebnisse der Simulationen mit Überschußenergien von 10eV sind in (c) und (d) gezeigt.

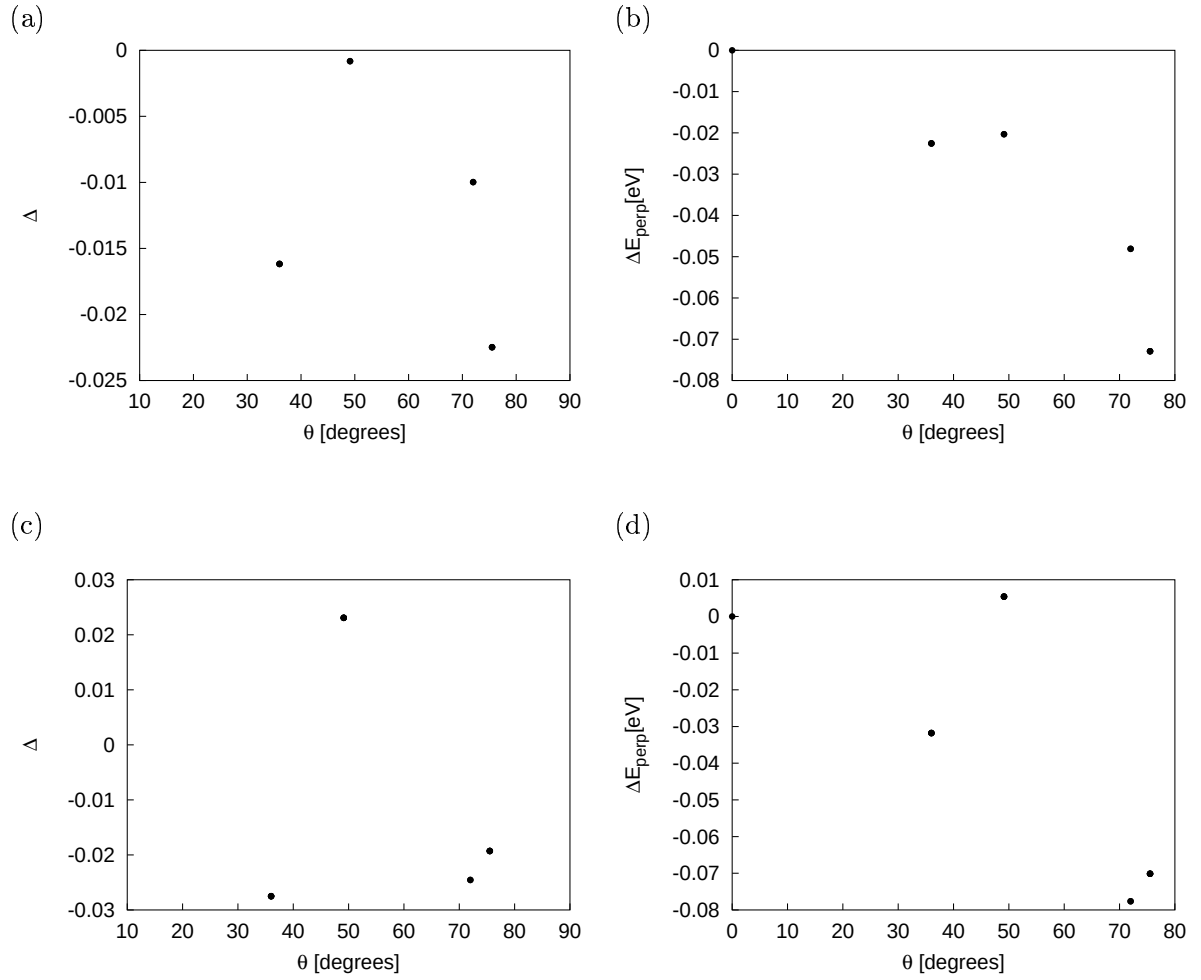


Abbildung 5.26: Der Vakuumtank auf HV. Abb. (a) und (b) zeigen den Δ -Parameter und die Verletzung der Transversalenergie für Elektronen mit der Überschussenergie von 100eV und dem Startradius von 0 mm. (c) und (d) zeigen diese Parameter für Elektronen der Energie $E = U_0 + 1\text{keV}$.

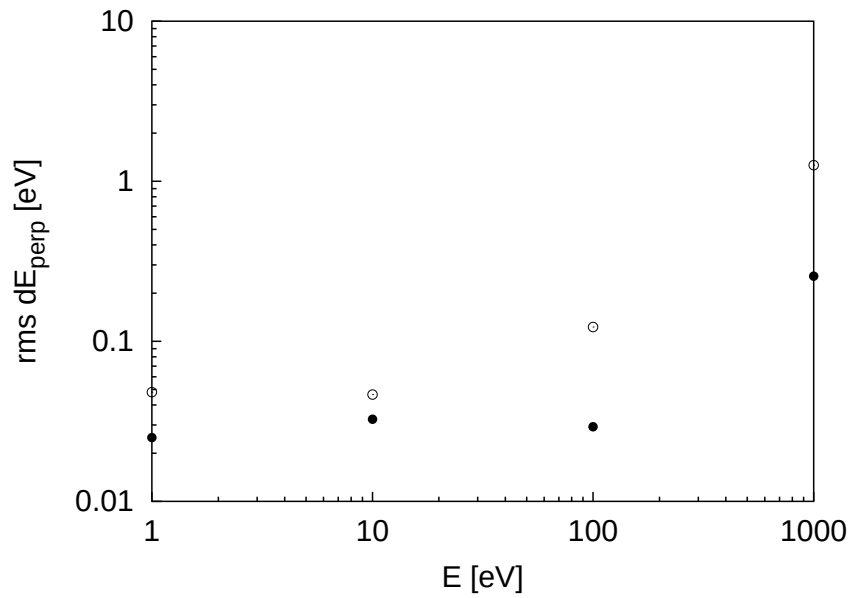
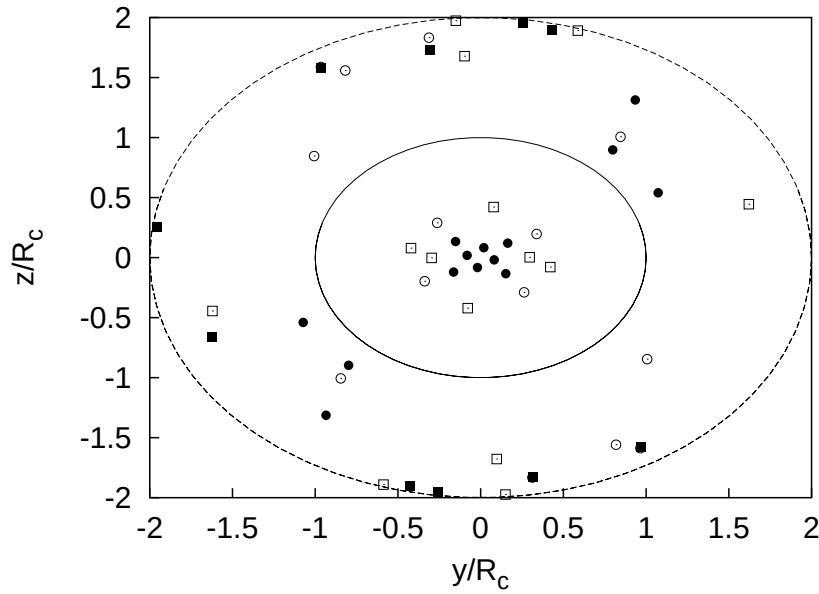


Abbildung 5.27: Der Vakuumtank auf HV. Es ist die Varianz der $\delta E_{\perp}$ gegen die Überschußenergie aufgetragen. Die vollen Kreise beschreiben auf der Achse fliegende Elektronen ($r = 32$ mm), während die Kreise Elektronen auf der äußeren Flugbahn zeigen.

(a)



(b)

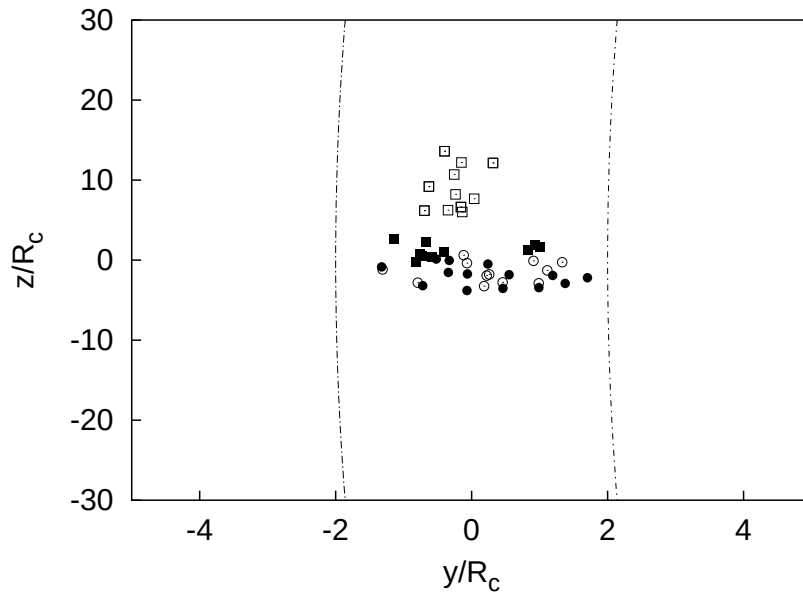


Abbildung 5.28: Der Vakuumtank auf HV. Es sind die Abweichungen der Teilchenposition von der erwarteten Position in der Analysierebene in Vielfachen des Zyklotronradius dargestellt. Die Plots zeigen die Ergebnisse für auf der Achse fliegende Elektronen ($r=0$ mm) (a) und achsenferne Elektronen ($r=32$ mm) (b). Die Überschußenergien betragen 1 eV (volle Kreise), 10 eV (Kreise), 100 eV (volle Quadrate) und 1 keV (Quadrate).

5.3 Zusammenfassung und Wertung

In diesem Kapitel wurden zwei verschiedene Spektrometerdesigns untersucht. Das erste Design besteht aus einem Vakuumtank und einer Elektrode auf einem Potential von -18600 V. In dieser Konfiguration wird ein Potentialdurchgriff von 1.13 V erreicht. Dem gegenüber steht ein Potentialdurchgriff von 0.41V, der im zweiten Entwurf erreicht wird. In diesem Design wird das Potential, ebenfalls -18600 V, direkt am Vakuumtank angelegt. Den Untersuchungen lagen für beide Systeme identische Magnetfelder zugrunde, deren Maximum in den Spektrometersolenoiden bei 3 T und deren Minimum in der Analysierebene bei 3 G liegt. Sie wurden von zwei supraleitenden Spektrometersolenoiden und drei Luftspulen zur Feldkorrektur erzeugt.

Für beide Systeme wurde gezeigt, daß sie Elektronen transmittieren, d.h. daß die Parallelenergie der Teilchen entlang der gesamten Flugbahn größer als das Retardierungspotential ist. Aufgrund dieses Ergebnisses wurden die numerische Genauigkeit, die Verletzung der Adiabasie und die Abbildungseigenschaften der beiden Spektrometerentwürfe untersucht.

Es ist anzumerken, daß diese beiden Systeme nicht exakt miteinander vergleichbar sind, da aufgrund der unterschiedlichen Elektrodenkonfigurationen am Startort der Elektronen unterschiedliche elektrische Potentiale vorliegen. Dies führt dazu, daß die Überschußenergien nicht identisch sind, und somit die unterschiedliche Flugzeiten für die Systeme entstehen.

Ein Vergleich der Ergebnisse ist trotzdem möglich, da die numerische Ungenauigkeit in beiden Fällen in der selben Größenordnung liegt. Die Verletzung der Gesamtenergie, die als Parameter zur Bewertung der Numerik heran gezogen wurde, liegt bei etwa 10 meV.

Auch die Verletzung der Adiabasie und die Positionsabweichungen der Elektronen in der Analysierebene liegen in der selben Größenordnung und zeigen das selbe Verhalten. Die maximale Verletzung der Transversalenergie liegt etwa im Bereich von 150 meV, was etwa 10^{-5} der untersuchten Energie entspricht. Eine adiabatische Führung ist also in beiden Spektrometerentwürfen gewährleistet.

Aufgrund der ähnlichen Simulationsergebnisse wird das zweite besprochene Design, in dem das Potential direkt am Vakuumtank angelegt wird, bevorzugt. Die Gründe dafür liegen in der höheren Homogenität des elektrischen Feldes in der Analysierebene, die sich durch den geringeren Potentialdurchgriff äußert, und in der geringeren technischen Komplexität des Systems, bei gleichen Abbildungs- und adiabatischen Eigenschaften. Desweiteren erlaubt dieses Design eine um 10%größere nutzbare Analysierebene, da keine zusätzlichen Elektroden innerhalb des Vakuumtanks angebracht werden müssen. Durch den Verzicht auf Elektroden werden auch die Oberflächen im Ultrahochvakuum um etwa $\frac{2}{3}$ reduziert, wodurch der Restgasanteil, der durch das

Ausgasen des Metalls entsteht verringert wird.

Zusammenfassend wäre ein Spektrometer, dessen Vakuumtank auf HV liegt vorzuziehen. Es muß aber noch geklärt werden, inwiefern sich die damit verbundenen technischen Schwierigkeiten lösen lassen.

Eine Möglichkeit wäre auch an der Innenwand des Tanks ein Drahtgitter anzubringen, an dem U_0 angelegt wird, während der Tank selbst auf dem Abschirmpotential $U_0 + \varepsilon$ liegt. Dann muß nicht der ganze Vakuumtank auf eine konstante Hochspannung stabilisiert werden. Außerdem könnten niederenergetische sekundäre Elektronen aus den Wänden energetisch von Eintritt in das sensitive Volumen abgehalten werden.

Kapitel 6

Das Vorspektrometer

In diesem Kapitel werden die Aufgaben eines möglichen Vorspektrometers dargelegt. Es werden Untersuchungen über die Flugzeit in einem System aus Vor- und Hauptspektrometer im Hinblick auf den Flugzeitmodus des MAC-E-Filters durchgeführt und gezeigt, daß durch eine Pulsung des Vorspektrometers die Gasquelle im TOF-Modus betrieben werden kann.

Bei dem Vorspektrometer handelt es sich, wie auch bei dem Hauptspektrometer, um einen MAC-E-Filter, dessen Prinzip in Kapitel 2 beschrieben wurde. Es ist vorgesehen, daß die Größe des Vorspektrometers etwa den Dimensionen der zur Zeit in Mainz und Troitsk betriebenen Spektrometern entspricht. Die in diesem Kapitel vorgenommenen Untersuchungen gehen basieren auf einem Entwurf, der von einem Spektrometer von 3 m Länge und 1.2 m Durchmesser ausgeht. Es wurde ein Design mit einer Elektrode simuliert. Das magnetische Führungsfeld wird von 2 Spektrometersolenoiden erzeugt, die ein maximales Magnetfeld von 5 T erzeugen. In der Mitte des Vorspektrometers herrscht ein minimales Magnetfeld von 500 G. Die angelegte Retardierungsspannung soll bei etwa -18 kV liegen, so daß durch das Vorspektrometer ein erster Energiefilter entsteht, der verhindert, daß niederenergetische Elektronen aus dem β -Zerfall in das Hauptspektrometer gelangen und dort durch Reaktionen mit dem Restgas zur Untergrundzählrate beitragen.

Da in Troitsk mit dem dort betriebenen Vorspektrometer Probleme aufgetreten sind, wurde in einer ersten Untersuchung festgestellt, ob das System aus Vor- und Hauptspektrometer (Abb. (6.1)) Elektronen transmittiert, falls kein Potential an der Elektrode des Vorspektrometers angelegt wird. Diese Untersuchung war erfolgreich, und es konnte gezeigt werden, daß eine Transmission der β -Elektronen gewährleistet ist. In der genannten Abbildung ist die dargestellte Flugbahn ($r=32\text{mm}$) asymmetrisch, da bei der Simulation der Detektorsolenoid nicht berücksichtigt wurde und so eine asymmetrische Magnetfeldkonfiguration zustande kam. Da es sich in diesem Kapitel aber nur um prinzipielle Untersuchungen handelt, wird kein Gewicht auf diese Tatsache gelegt.

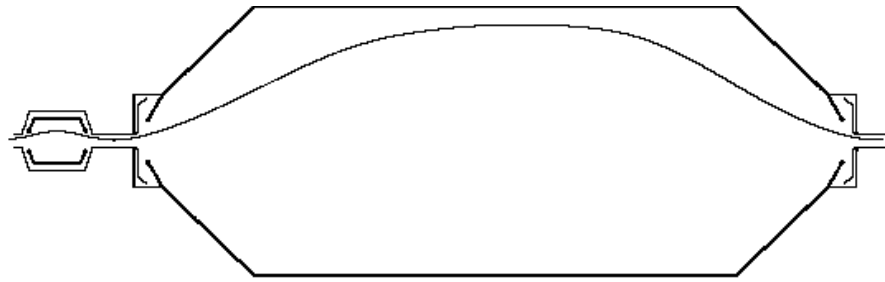


Abbildung 6.1: Vor- und Hauptspektrometer. Die Abbildung zeigt den linearen Aufbau von Vor- und Hauptspektrometer. Die dargestellte Elektronenbahn hat einen Startradius von 32 mm, was der äußeren Flugbahn entspricht.

6.1 Time-of-Flight Modus und Pulsung

Um einen Betrieb des Spektrometers im Flugzeitmodus zu ermöglichen, muß eine Startzeit der Elektronen definiert werden. Dies wird im Mainzer Aufbau durch die Pulsung der Elektronenquelle realisiert. Durch das Anlegen eines Spannungspulses bekommen die Elektronen erst die notwendige Energie, um das Retardierungspotential des Spektrometers zu überwinden, so daß auf diesem Weg ein Startzeitpunkt definiert wird. Danach wird die Quellspannung abgesenkt, damit keine weiteren Elektronen durch das Spektrometer zu Detektor gelangen können. Der Zusammenhang zwischen der Quellspannung und der Zählrate ist in Abb. (6.2) dargestellt.

Bei KATRIN sind zwei Elektronenquellen geplant: Zum einen die Festkörperquelle,

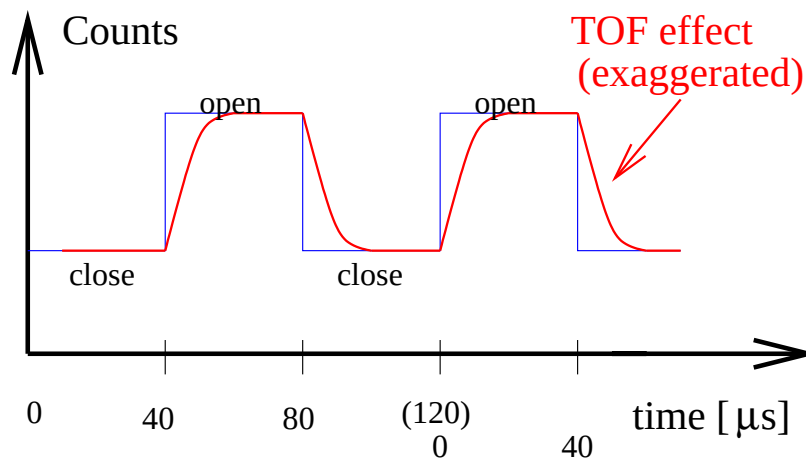


Abbildung 6.2: Pulsung der Festkörperquelle. Es ist der prinzipielle Zusammenhang zwischen der Quellspannung und der Zählrate dargestellt. Die Rechteckkurve beschreibt die Quellspannung, welche das Spektrometer für Elektronen öffnet oder schließt (open/close). Die zweite Kurve stellt den Verlauf der Zählrate dar. Sie ist durch die Flugzeit der Elektronen zu erklären: Nachdem das Spektrometer geöffnet wurde benötigen die schnellsten Teilchen noch eine gewisse Zeit, bis sie den Detektor erreichen. Ebenso treffen noch Elektronen am Detektor ein, wenn das Spektrometer bereits wieder geschlossen ist. Die Abb. ist [Sch01] entnommen.

bei der es, wie auch im Mainzer Aufbau, möglich ist eine Quellspannung anzulegen, zum anderen die Gasquelle, für die eine Pulsung durch die oben geschilderte Weise nur durch hohen technischen Aufwand zu realisieren ist. Dies ist auf die Konstruktion der Gasquelle zurückzuführen, die in Kapitel 2.3 beschrieben ist. Benutzt man also einen Spannungspuls an der Quelle als Startsignal für den Flugzeitmodus, ist nur die Festkörperquelle im MAC-E-TOF Modus zu betreiben.

Geht man jedoch von einer Pulsung der Quelle zu einer Pulsung des Vorspektrometers über, so ist es möglich den MAC-E-TOF Modus für beide Quellen zu nutzen. Dieses Pulsen des Vorspektrometers kann durch eine Erhöhung der Retardierungsspannung an der Elektrode erreicht werden. In dem Fall wird am Beginn des Vorspektrometers ein Startzeitpunkt für die Elektronen definiert.

In Abb.(6.3) ist eine Simulation von Elektronen im Vor- und Hauptspektrometer gezeigt. Es ist die Flugzeit gegenüber der Überschußenergie aufgetragen. Die Elektronen wurden alle auf der Achse ($r=0$ mm) gestartet, wobei sich die unterschiedlichen Kurven durch verschiedene Startwinkel ($\theta = 0^\circ, 30^\circ, 51^\circ$) ergeben. Dabei beschreiben die Kurven mit längeren Flugzeiten die Elektronen mit höheren Startwinkeln. Es ist nun möglich, wie in Kapitel 2.2 beschrieben, eine minimale Flugzeit auszuwählen und so einen Schnitt für die Energie festzulegen.

Ein weiterer Vorteil der Pulsung am Vorspektrometer liegt darin, daß hier ein Filter entsteht, das von den Elektronen entweder passiert wird, oder nicht. Dadurch fallen Effekte, die durch das Schalten auftreten, wie z.B. Einschwingvorgänge, nicht so ins Gewicht wie bei einem direkten Pulsen der Quellen.

Durch das Vorspektrometer entsteht also, neben seiner ursprünglichen Aufgabe, der Reduzierung der Untergrundzählrate, eine geeignete Möglichkeit den MAC-E-TOF Modus für KATRIN zu realisieren.

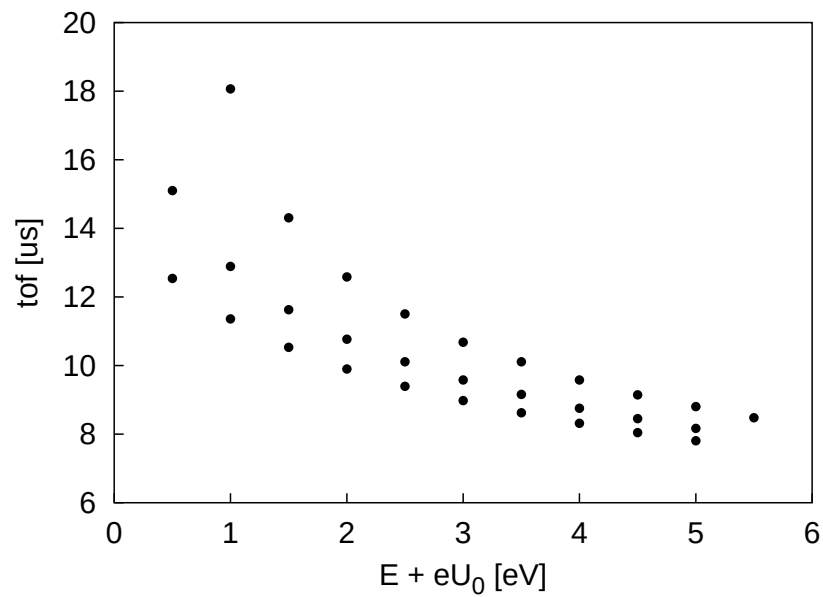


Abbildung 6.3: Simulation der Flugzeit in Vor- und Hauptspektrometer. Es ist die Flugzeit der Elektronen gegen ihre Überschußenergie aufgetragen. Alle Elektronen flogen, mit unterschiedlichen Startwinkeln (0° , 30° , 51°), auf der Achse.

Kapitel 7

Zusammenfassung und Ausblick

Das Ziel dieser Aufgabe war es Designstudien zu verschiedenen Komponenten von KATRIN, dem geplanten KARlsruhe TRItium Neutrino Experiment durchzuführen. KATRIN zielt darauf ab, die Obergrenze der Anilektronneutrinomasse in den sub-eV Bereich zu verschieben. Zur Zeit wird diese Größe in Troitsk und Mainz experimentell aus Untersuchungen des Tritium- β -Spektrums bestimmt, was zu einer Obergrenze von $2.5 \text{ eV}/c^2$ [Lob99], bzw. $2.2 \text{ eV}/c^2$ [Bon01] führt. Die Sensitivitätsgrenze dieser Experimente ist somit erreicht. Zur Verbesserung der Sensitivität bei gleicher Untergrundzählrate ist eine bessere Auflösung des Spektrometers und eine stärkere, d.h. größere Elektronenquelle notwendig. Weiterhin wird eine bessere Systematik angestrebt. Diese Punkte sind nur durch einen wesentlich größeren Spektrometeraufbau zu realisieren.

Da sich das Prinzip des MAC-E-Filters für diese Art von Experimenten bewährt hat dient es auch als Grundlage für KATRIN. Die Komplexität des Experiments, aber vor allem die Dimensionen, werden jedoch gegenüber den Aufbauten in Troitsk und Mainz stark wachsen. So sind zum Beispiel zwei Quellsysteme geplant, eine Tritium-Festkörperquelle, wie in Mainz, und eine gasförmige Tritiumquelle, wie in Troitsk. Zwischen diesen Quellen und dem Spektrometer, welches eine Länge von 20 m und einen Durchmesser von 7 m haben wird, wird sich ein Transportsystem für die β -Elektronen befinden.

In Verlauf dieser Arbeit wurden Simulationen von zwei verschiedenen Entwürfen des Transportsystems durchgeführt und zwei Designs für das Spektrometer angefertigt und ebenfalls simuliert.

Dazu wurde zunächst ein Computerprogramm entwickelt, das es erlaubt die Magnetfelder und die magnetischen Potentiale gegeneinander gekippter Solenoide zu berechnen. Um den Rechenaufwand zu minimieren wurde zu diesem Zweck ein neuer Algorithmus von F. Glueck [Glu00] implementiert, der die Solenoide aus langen, dünnen Spulen aufbaut. Neben den Berechnungen der Felder und Potentiale erlaubt das Programm auch die Berechnung der Magnetfeldlinien, was zur Veranschaulichung

des magnetischen Flußschlauches und zur Bestimmung des Trapping-Phasenraumes verschiedener Spulenkonfigurationen benötigt wird.

Mit Hilfe der berechneten magnetischen Potentiale war es möglich Bahnverfolgungen für Elektronen in den gegebenen Spulen- und Spektrometerkonfigurationen durchzuführen. Zu diesem Zweck wurde SIMION 7, ein kommerzielles Programm benutzt. Die durchgeführten Simulationen galten der Untersuchung der numerischen Genauigkeit, der Adiabasierverletzung und den Abbildungseigenschaften des Transportsystems und des Spektrometers.

Durch die Untersuchungen des Transportsystems wurden zwei Entwürfe, einer der Firma ACCEL und einer der Troitsker Arbeitsgruppe, miteinander verglichen. Beide Systeme konnten mit einer numerischen Genauigkeit von etwa 10^{-6} untersucht werden. Es stellte sich heraus, daß das Troitsker Design über bessere adiabatische Eigenschaften verfügt, jedoch aufgrund des höheren Volumens auch ein größeres Trappingvolumen hat. Der in diesem Design vorgesehene größere Pumpstutzen erlaubt allerdings auch eine höhere Saugleistung, wodurch ein geringerer Restgasdruck erreicht werden kann, was wiederum zu weniger Zerfällen innerhalb des Trappingvolumens führt. Es ist aber auch möglich die Geometrie des Designs an einen kleineren Pumpstutzen, der dem ACCEL Design entspricht, anzupassen und erneute Untersuchungen durchzuführen.

Es wurden auch zwei Spektrometerdesigns, die beide im Rahmen dieser Arbeit angefertigt wurden, miteinander verglichen. Es handelt sich zum einen um ein Design mit nur einer Elektrode, an der das Retardierungspotential angelegt wird, zum anderen um einen Entwurf, bei dem das Retardierungspotential direkt am Vakuumtank angelegt wird. Beide Entwürfe zeigen ähnliche Eigenschaften bezüglich des adiabatischen Verhaltens und der Abbildungseigenschaften. Die Verletzung der Transversalenergie, eine Maß für die Verletzung der Adiabasie, beträgt im maximalen Fall 150 meV, was einem 10^{-5} Effekt entspricht. Eine adiabatische Führung der Elektronen ist also gewährleistet. Aufgrund der ähnlichen Simulationsergebnisse wird das zweite Design, bei dem das Potential direkt am Vakuumtank anliegt, bevorzugt, da die technische Komplexität aufgrund der fehlenden Elektrode sehr viel geringer ist.

Weitere Simulationen wurden durchgeführt, um zu überprüfen, ob der MAC-E-TOF Modus für das neue Experiment genutzt werden kann. Es stellte sich heraus, daß dies tatsächlich durch das Pulsen der Retardierungsspannung des Vorspektrometers möglich ist.

Die nächsten Schritte, die zu einem abschließenden Design führen können liegen in der Modifikation der bereits hier untersuchten Komponenten. So ist es notwendig die Entwürfe des Transportsystems so zu modifizieren, daß sie den Designanforderungen entsprechen, und die vorgenommenen Untersuchungen zu wiederholen. Desweiteren sollten Untersuchungen mit einer größeren Anzahl von Transportelementen durchgeführt werden, um festzustellen, ob sich Effekte wie die Adiabasierverletzung verstärken

oder aufheben.

In dieser Arbeit musste aus Zeitgründen auf die Untersuchung von Teilchenfallen verzichtet werden, diese Untersuchungen sollten aber Gegenstand zukünftiger Untersuchungen sein.

Erste Schritte in dieser Richtung werden im Rahmen der Diplomarbeit von T. Thümmeler [Thu01] unternommen, indem ein Programm zur Bahnverfolgung in adiabatischer Näherung entwickelt wird. Das Ziel ist es mögliche Teilchenfallen im Spektrometersystem zu identifizieren, sie zu vermeiden oder Möglichkeiten zur Störung der Speicherbedingungen aufzuzeigen. Eine dieser Möglichkeiten liegt darin die Spektrometersolenoiden nicht linear, sondern gekippt zueinander anzuordnen. Durch die so entstehenden Driften werden die gespeicherten Teilchen aus dem im Detektor abgebildeten Flußschlauch geführt. Eine weitere Möglichkeit geladene Teilchen aus Fallen zu entfernen ist das Anlegen eines elektrischen Dipolfeldes, durch das eine $\vec{E} \times \vec{B}$ -Drift verursacht wird, die die Teilchen aus der Falle führt.

Anhang A

Quelltexte

A.1 BField3D

```
// Steering routine for calculation of 3-dimensional
// magnetic fields and potentials
// using modules: sngl_coil_pot, sngl_coil_bfield, mag_c, b_util
// Written: Feb.-April 2001 by B.Flatt, Ch.Weinheimer
//

#include <stdio.h>
#include <math.h>
#include <stdarg.h>
#include "b_struct.h"
#include "b_util.h"

char* filename;
char version_str[50];

int main(int argc, char *argv[]){

    struct coils spul;
    struct bfield_geom geo;
    int istep_max, choice;

    sprintf(version_str,"%s","1.0 date: 04.04.2001");
    if (argc<2){
        printf("bfield_3d filename\n");
        printf(" filename is name of *.par parameter file without .par-tail\n");
        printf(" example: if you have prespec.par then call bfield_3D prespec\n");
        printf(" output files: prespec.blines (B-lines), prespec.bfield (B-field)\n");
        printf("version: %s\n",version_str);
        printf("\n");
    } else
    {
        filename = argv[1];
        printf("general filename: %s\n",filename);
        read_par(&spul, &geo, filename);
        print_par(&spul, &geo);

        do
```



```

        {
printf("Ende:                                0\n");
printf("print parameters:                    1\n");
printf("calculate magnetic field:             2\n");
printf("calculate magnetic potential 3D (for SIMION): 3\n");
printf("calculate magnetic potential 2D (for SIMION): 4\n");
printf("calculate magnetic field lines:           5\n");
printf("write kumac file to draw coils:          6\n");

scanf("%d", &choice);
switch(choice){
case 0 :
    break;
case 1 :
    print_par(&spul,&geo);
    break;
case 2 :
    write_bfield(&spul, &geo, filename);
    write_bfield_from_potential(&spul, &geo, filename);
    break;
case 3 :
    // write_potential(&spul, &geo, filename);
    write_potential_pa(&spul, &geo, filename);
    break;
case 4 :
    // write_potential(&spul, &geo, filename);
    write_potential_2d_pa(&spul, &geo, filename);
    break;
case 5 :
    istep_max = calc_blines(&spul, &geo);
    write_blines(&spul, &geo, filename, istep_max);
    write_coil_kumac(&spul, &geo, filename);
    write_blines_kumac(&spul, &geo, filename);
    break;
case 6 :
    write_coil_kumac(&spul, &geo, filename);
    write_konvergence_kumac(&spul, &geo, filename);
    break;
default :
    printf("--- Fehler ---\n");
}

    } while (choice != 0);
}
return 0;
}

```

A.2 mag_3d

```

// Function for Calculation of 3-dimensional magnetic fields and potentials
// based on sngl_coil_bfield_2d and sngl_coil_pot_2d
// using modules: math_vector,sngl_coil_bfield_2d,sngl_pot_bfield_2d

```

```
// Written: Feb.-April 2001 by B.Flatt, Ch.Weinheimer
//

#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include "math_vector.h"
#include "b_struct.h"
#include "sngl_coil_bfield_2d.h"
#include "sngl_coil_pot_2d.h"

#define I_PRINT 0

//*****BField3D*****
// still single coil but in 3D
//
void bfield3d(double *vec, double *b, int coilnr, struct coils *spulen, struct bfield_geom *geom)
{
    double R_0[3];           //vector to the center of solenoid
    double n_vec[3], l_vec[3], t_vec[3];
                                //unitary vectors in the 3 dimensions n :z-direction
                                // l:perpendicular in plane, t : perpendicular to plane
    double R_1[3];           //Vector from Solenoid to "Point-of-Interest"
    double n,l,t;            //Scalar products of n,l,t with R_1
    double theta;            //Rotation angle
    double bz, br;           //Results of bfield_11-call
    double z, rad;           //parameters for bfield-function
    double br_plane;         //component of radial b-field in plane

    if (I_PRINT>0) printf("vector: %f %f %f\n",vec[0],vec[1],vec[2]);

    R_0[0] = (*spulen).position[coilnr];
    R_0[1] = (*spulen).offset[coilnr];
    R_0[2] = 0;
    if (I_PRINT>0) printf("R_0: %f %f %f\n",R_0[0],R_0[1],R_0[2]);

    R_1[0] = vec[0]-R_0[0];
    R_1[1] = vec[1]-R_0[1];
    R_1[2] = vec[2]-R_0[2];
    if (I_PRINT>0) printf("R_1: %f %f %f\n",R_1[0],R_1[1],R_1[2]);

    theta = (*spulen).rotation_angle[coilnr];

    n_vec[0] = cos(theta);
    n_vec[1] = sin(theta);
    n_vec[2] = 0;
    n=scalar_prod(n_vec,R_1);

    l_vec[0] = -sin(theta);
    l_vec[1] = cos(theta);
    l_vec[2] = 0;
    l=scalar_prod(l_vec,R_1);

    t_vec[0] = 0;
    t_vec[1] = 0;
    t_vec[2] = 1;
}
```

```

t=scalar_prod(t_vec,R_1);

z  = n+(*spulen).position[coilnr];
rad = sqrt(1+l*t*t);

if (I_PRINT>0) printf("an bfield z %f rad %f\n", z,rad);
b_coil(z, rad, coilnr, spulen, &bz, &br);
if (I_PRINT>0) printf("von bfield bz %f br %f\n", bz, br);

/* neue z-Komponente (senkrecht zur Ebene) aus br bestimmen */
/* restliche br-Komponente -> br_plane liegt in Ebene      */
if (rad == 0)
{
    b[0] = bz;
    b[1] = 0;
    b[2] = 0;
}
else
{
    b[2] = R_1[2]/rad*br;
    br_plane = signum(l)*signum(br)*sqrt(br*br-b[2]*b[2]);
    /* aus br_plane und bz durch Drehung neue x- und neue y-Komponente bestimmen */
    b[0] = cos(theta)*bz - sin(theta)*br_plane;
    b[1] = sin(theta)*bz + cos(theta)*br_plane;
}
if (I_PRINT>0) printf("b out: %f %f %f\n",b[0],b[1],b[2]);
}

//*****bfield3d_all*****

void bfield3d_all(double *vec, double *b_all, struct coils *spulen,struct bfield_geom *geom)
{
    int i_coil;
    double b[3];
    b_all[0]=b_all[1]=b_all[2]=0;
    for (i_coil = 0;i_coil <(*spulen).number_of_coils; i_coil ++)
    {
        b[0]=b[1]=b[2]=0;
        bfield3d(vec,b,i_coil,spulen,geom);
        if (I_PRINT>0) printf("vec %f %f %f, i %d, b %f %f %f\n",vec[0],vec[1],vec[2],i_coil,b[0],b[1],b[2]);
        b_all[0] += b[0];
        b_all[1] += b[1];
        b_all[2] += b[2];
    }
}

//***** potential3d *****
// still single coil but in 3D
//
double potential3d(double *vec,int coilnr, struct coils *spulen,struct bfield_geom *geom)
{
    double R_0[3]; //vector to the center of solenoid
    double n_vec[3],l_vec[3],t_vec[3];
    //unitary vectors in the 3 dimensions n :z-direction
    // l:perpendicular in plane, t : perpendicular to plane

```

```

double R_1[3];           //Vector from Solenoid to "Point-of-Interest"
double n,l,t;           //Scalar products of n,l,t with R_1
double theta;           //Rotation angle
double z, rad, pot_3d;   //parameters for bfield-function

if (I_PRINT>0) printf("vector: %f %f %f\n",vec[0],vec[1],vec[2]);

R_0[0] = (*spulen).position[coilnr];
R_0[1] = (*spulen).offset[coilnr];
R_0[2] = 0;
if (I_PRINT>0) printf("R_0: %f %f %f\n",R_0[0],R_0[1],R_0[2]);

R_1[0] = vec[0]-R_0[0];
R_1[1] = vec[1]-R_0[1];
R_1[2] = vec[2]-R_0[2];
if (I_PRINT>0) printf("R_1: %f %f %f\n",R_1[0],R_1[1],R_1[2]);

theta = (*spulen).rotation_angle[coilnr];

n_vec[0] = cos(theta);
n_vec[1] = sin(theta);
n_vec[2] = 0;
n = scalar_prod(n_vec,R_1);
if (I_PRINT>0) printf("n_vec: %f %f %f\n",n_vec[0],n_vec[1],n_vec[1]);
if (I_PRINT>0) printf("n: %f \n",n);

l_vec[0] = -sin(theta);
l_vec[1] = cos(theta);
l_vec[2] = 0;
l=scalar_prod(l_vec,R_1);
if (I_PRINT>0) printf("l_vec: %f %f %f\n",l_vec[0],l_vec[1],l_vec[2]);
if (I_PRINT>0) printf("l: %f \n",l);

t_vec[0] = 0;
t_vec[1] = 0;
t_vec[2] = 1;
t=scalar_prod(t_vec,R_1);
if (I_PRINT>0) printf("t_vec: %f %f %f\n",t_vec[0],t_vec[1],t_vec[2]);
if (I_PRINT>0) printf("t: %f \n",t);

z = n+(*spulen).position[coilnr];
rad = sqrt(l*l+t*t);

if (I_PRINT>0) printf("an potential z %f rad %f\n", z,rad);
pot_3d = thick_coil_pot(rad, z, coilnr, spulen);

return pot_3d;
}

//***** potential3d_all *****

double potential3d_all(double *vec, struct coils* spulen, struct bfield_geom *geom)
{
    int coilnr;
    double potential, potential_sum;

```

```

    potential_sum = 0;
    for ( coilnr = 0 ; coilnr < (*spulen).number_of_coils ;coilnr ++)
    {
        potential = potential3d(vec,coilnr,spulen,geom);
        if (I_PRINT >0) printf("i_coil %d, potential %f\n",coilnr,potential);
        potential_sum += potential;
    }
    if (I_PRINT >1) printf("pot_all : %f\n", potential_sum);
    return potential_sum;
}

```

A.3 b_util

```

// Utilities for 3 dimensional magnetic fields and potentials
// using module mag_3d
// Written: Feb.-April 2001 by B.Flatt, Ch.Weinheimer
// Changes: 04/05/01 changed write_pa to simion7 format and symmetry to planar
//

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <fcntl.h>
#include <unistd.h>
#include "b_struct.h"
#include "math_vector.h"
#include "mag_3d.h"

#define H_DIFF 0.0001 // step of numerical differentiation: pot -> bfield
#define I_PRINT 0 // debug output control parameter
#define MAX_STEP 1000
#define Z_CONV_MINMAX 100.0
#define Z_CONV_STEP 2.0

double bl_x[MAX_BL][MAX_STEP],bl_y[MAX_BL][MAX_STEP],bl_z[MAX_BL][MAX_STEP];
double bl_b[MAX_BL][MAX_STEP];

#define I_BLINES_POT 0 // calc Bfield by differentiation of mag. potenial

//***** definitions for binary PA file for SIMION *****
#define PLANAR 1
#define CYLINDRICAL 0
#define MIRROR_X 1
#define MIRROR_Y 2
#define MIRROR_Z 4
#define MAGNETIC_PA 8

struct header_3d{
    int mode;
    int symmetry;
    double max_voltage;
    int nx;

```

```

    int ny;
    int nz;
    int mirror;
};

/***** read_parfile *****/

void read_par(struct coils* spulen, struct bfield_geom* geom, char *filename){
    char* par_filename[30];
    FILE* parfile;

    int j, idummy;
    double xposition, xradius, xthickness, xlength, xwindings, xcurrent, xangle, xoffset;
    double xmin,xmax,xstep,ymin,ymax,ystep,zmin,zmax,zstep,xthin_coils;
    double bline_x;

    sprintf(par_filename,"%s.par",filename);
    printf("\nNow reading the magnetic parameter file %s...\n",par_filename);
    parfile = fopen(par_filename, "r");
    if (parfile == (FILE *)0) {
        fprintf(stderr, "Can't open the parameter file %s!\n",par_filename);
        exit(1);
    }
    fscanf(parfile, "%d", &idummy);
    (*spulen).number_of_coils = idummy;
    printf("par file contains %d coils\n",(*spulen).number_of_coils);

    for(j=0; j < (*spulen).number_of_coils; j++) {
        fscanf(parfile, "%lf %lf %lf %lf %lf %lf %lf %lf", &xposition, &xradius, &xthickness,
            &xlength, &xwindings,&xthin_coils, &xcurrent, &xangle, &xoffset);
        (*spulen).position[j] = xposition;
        (*spulen).radius[j] = xradius;
        (*spulen).thickness[j]= xthickness;
        (*spulen).length[j] = xlength;
        (*spulen).windings[j] = xwindings;
        (*spulen).number_of_thin_coils[j] = xthin_coils;
        (*spulen).current[j] = xcurrent;
        (*spulen).rotation_angle[j] = xangle/180.*M_PI;
        (*spulen).offset[j] = xoffset;
        (*spulen).delta_r[j] = xthickness/xthin_coils;
        (*spulen).curr_div_thincoils[j]= xcurrent/xthin_coils;
    }

    fscanf(parfile, "%lf %lf %lf", &xmin,&xmax,&xstep);
    (*geom).x_min = xmin;
    (*geom).x_max = xmax;
    (*geom).stepx = xstep;
    fscanf(parfile, "%lf %lf %lf", &ymin,&ymax,&ystep);
    (*geom).y_min = ymin;
    (*geom).y_max = ymax;
    (*geom).stepy = ystep;
    fscanf(parfile, "%lf %lf %lf", &zmin,&zmax, &zstep);
    (*geom).z_min = zmin;
    (*geom).z_max = zmax;
    (*geom).stepz = zstep;
    fscanf(parfile, "%lf", &bline_x);
    (*geom).xbl = bline_x;

```

```

fclose(parfile);
printf("Done.\n");

(*geom).izmin = 0;
(*geom).izmax = ((*geom).z_max - (*geom).z_min)/(*geom).stepz;
if ((*geom).izmax >= MAX_Z) (*geom).izmax = MAX_Z - 1;

(*geom).ixmin = 0;
(*geom).ixmax = ((*geom).x_max - (*geom).x_min)/(*geom).stepx;
if ((*geom).ixmax >= MAX_X) (*geom).ixmax = MAX_X - 1;

(*geom).iymin = 0;
(*geom).iymax = ((*geom).y_max - (*geom).y_min)/(*geom).stepy;
if ((*geom).iymax >= MAX_Y) (*geom).iymax = MAX_Y - 1;
}

//***** print_par *****

void print_par(struct coils * spulen, struct bfield_geom * geom){
    int j;
    printf("\n");
    printf("\n Anzahl der (*Spulen) : %d\n\n", (*spulen).number_of_coils); /* Parameter anzeigen */
    printf("  Nr.  POSITION[cm]  RADIUS[cm]  THICKNESS[cm]  LENGTH[cm]  WINDINGS  THIN COILS  CURRENT\n");
    for (j=0; j<(*spulen).number_of_coils; j++){
        printf("%4d%15.2f%12.2f%11.2f%12.2f%12f%10.2f  %10.2f  %10.1f%12.2f\n",j+1,
            (*spulen).position[j], (*spulen).radius[j],
            (*spulen).thickness[j],(*spulen).length[j],
            (*spulen).windings[j], (*spulen).number_of_thin_coils[j], (*spulen).current[j],
            (*spulen).rotation_angle[j]*180./M_PI, (*spulen).offset[j]);
    }
    printf("\n Iterationsbereich:\n");
    printf("      XMIN[cm]      XMAX[cm]      STEPX[cm]\n");
    printf("%12.2f%12.2f%12.2f\n", (*geom).x_min, (*geom).x_max, (*geom).stepx);
    printf("      YMIN[cm]      YMAX[cm]      STEPY[cm]\n");
    printf("%12.2f%12.2f%12.2f\n", (*geom).y_min, (*geom).y_max, (*geom).stepy);
    printf("      ZMIN[cm]      ZMAX[cm]      STEPZ[cm]\n");
    printf("%12.2f%12.2f%12.2f\n", (*geom).z_min, (*geom).z_max, (*geom).stepz);
    printf("\n R-Bereich fuer Feldlinien\n");
    printf("      bei X0[cm]\n");
    printf("%12.2f\n", (*geom).xbl);
    printf("\n");
}

//***** write_coil_kumac *****

void write_coil_kumac(struct coils* spulen, struct bfield_geom* geom,
                    char *filename){
    char* kumac_filename[30];
    FILE* kumacfile;
    int j;
    double ctheta,stheta,x0,x1,x2,x3,y0,y1,y2,y3;

    sprintf(kumac_filename,"%s.kumac",filename);
    printf("\nNow reading the magnetic parameter file %s...\n",kumac_filename);
    kumacfile = fopen(kumac_filename, "w");
    if (kumacfile == (FILE *)0) {

```

```

    fprintf(stderr, "Can't open the kumac file %s!\n", kumac_filename);
    exit(1);
}
for (j=0; j<(*spulen).number_of_coils; j++){
    ctheta = cos((*spulen).rotation_angle[j]);
    stheta = sin((*spulen).rotation_angle[j]);
    x0 = (*spulen).position[j] - stheta*(*spulen).radius[j]
        - ctheta*(*spulen).length[j]/2.0;
    y0 = (*spulen).offset[j] + ctheta*(*spulen).radius[j]
        - stheta*(*spulen).length[j]/2.0;
    x1 = (*spulen).position[j] - stheta*(*spulen).radius[j]
        + ctheta*(*spulen).length[j]/2.0;
    y1 = (*spulen).offset[j] + ctheta*(*spulen).radius[j]
        + stheta*(*spulen).length[j]/2.0;
    x2 = (*spulen).position[j] - stheta*((*spulen).radius[j]+(*spulen).thickness[j])
        - ctheta*(*spulen).length[j]/2.0;
    y2 = (*spulen).offset[j] + ctheta*((*spulen).radius[j]+(*spulen).thickness[j])
        - stheta*(*spulen).length[j]/2.0;
    x3 = (*spulen).position[j] - stheta*((*spulen).radius[j]+(*spulen).thickness[j])
        + ctheta*(*spulen).length[j]/2.0;
    y3 = (*spulen).offset[j] + ctheta*((*spulen).radius[j]+(*spulen).thickness[j])
        + stheta*(*spulen).length[j]/2.0;

    fprintf(kumacfile, "line %f %f %f %f\n", x0, y0, x1, y1);
    fprintf(kumacfile, "line %f %f %f %f\n", x1, y1, x2, y2);
    fprintf(kumacfile, "line %f %f %f %f\n", x2, y2, x3, y3);
    fprintf(kumacfile, "line %f %f %f %f\n", x3, y3, x0, y0);
    fprintf(kumacfile, "line %f %f %f %f\n", x0, y0, x2, y2);
    fprintf(kumacfile, "line %f %f %f %f\n", x1, y1, x3, y3);
    x0 = (*spulen).position[j] + stheta*(*spulen).radius[j]
        - ctheta*(*spulen).length[j]/2.0;
    y0 = (*spulen).offset[j] - ctheta*(*spulen).radius[j]
        - stheta*(*spulen).length[j]/2.0;
    x1 = (*spulen).position[j] + stheta*(*spulen).radius[j]
        + ctheta*(*spulen).length[j]/2.0;
    y1 = (*spulen).offset[j] - ctheta*(*spulen).radius[j]
        + stheta*(*spulen).length[j]/2.0;
    x2 = (*spulen).position[j] + stheta*((*spulen).radius[j]+(*spulen).thickness[j])
        - ctheta*(*spulen).length[j]/2.0;
    y2 = (*spulen).offset[j] - ctheta*((*spulen).radius[j]+(*spulen).thickness[j])
        - stheta*(*spulen).length[j]/2.0;
    x3 = (*spulen).position[j] + stheta*((*spulen).radius[j]+(*spulen).thickness[j])
        + ctheta*(*spulen).length[j]/2.0;
    y3 = (*spulen).offset[j] - ctheta*((*spulen).radius[j]+(*spulen).thickness[j])
        + stheta*(*spulen).length[j]/2.0;

    fprintf(kumacfile, "line %f %f %f %f\n", x0, y0, x1, y1);
    fprintf(kumacfile, "line %f %f %f %f\n", x1, y1, x2, y2);
    fprintf(kumacfile, "line %f %f %f %f\n", x2, y2, x3, y3);
    fprintf(kumacfile, "line %f %f %f %f\n", x3, y3, x0, y0);
    fprintf(kumacfile, "line %f %f %f %f\n", x0, y0, x2, y2);
    fprintf(kumacfile, "line %f %f %f %f\n", x1, y1, x3, y3);
}
fclose(kumacfile);
return;
}

//***** write_konvergence_kumac *****

```



```

void write_konvergence_kumac(struct coils* spulen, struct bfield_geom* geom,
                           char *filename){
    char* kumac_filename[30];
    FILE* kumacfile;
    int j,i_step;
    double ctheta,stheta,radius,radius_sqr;
    double zMin,zMax,z_zMin,z_zMax,z_zMin_sqr,z_zMax_sqr,z;
    double R_c,x1_3d,x1_3d_old,y1_3d,y1_3d_old, x2_3d,x2_3d_old,y2_3d,y2_3d_old;

    sprintf(kumac_filename,"%s.konv.kumac",filename);
    printf("\nNow reading the magnetic parameter file %s...\n",kumac_filename);
    kumacfile = fopen(kumac_filename, "w");
    if (kumacfile == (FILE *)0) {
        fprintf(stderr, "Can't open the kumac file %s!\n",kumac_filename);
        exit(1);
    }
    for (j=0; j<(*spulen).number_of_coils; j++){
        ctheta = cos((*spulen).rotation_angle[j]);
        stheta = sin((*spulen).rotation_angle[j]);
        radius = (*spulen).radius[j];
        radius_sqr = radius*radius;
        zMin = - (*spulen).length[j]/2.0;
        zMax = + (*spulen).length[j]/2.0;

        i_step = 0;
        z=-Z_CONV_MINMAX;
        x1_3d=y1_3d=x2_3d=y2_3d=0;
        while (z<=Z_CONV_MINMAX){
            z_zMin = z-zMin;
            z_zMax = z-zMax;
            z_zMin_sqr = z_zMin * z_zMin;
            z_zMax_sqr = z_zMax * z_zMax;
            x1_3d_old = x1_3d;
            y1_3d_old = y1_3d;
            x2_3d_old = x2_3d;
            y2_3d_old = y2_3d;
            if (z < zMin){
                R_c = sqrt(radius_sqr + z_zMin_sqr);
            } else {
                if (z > zMax) {
                    R_c = sqrt(radius_sqr + z_zMax_sqr);
                } else {
                    R_c = radius;
                }
            }
            x1_3d = (*spulen).position[j] - stheta*R_c
                - ctheta*z;
            y1_3d = (*spulen).offset[j] + ctheta*R_c
                - stheta*z;
            x2_3d = (*spulen).position[j] + stheta*R_c
                - ctheta*z;
            y2_3d = (*spulen).offset[j] - ctheta*R_c
                - stheta*z;

            if (i_step>0){
                fprintf(kumacfile,"line %f %f %f %f\n",x1_3d_old,y1_3d_old,x1_3d,y1_3d);
                fprintf(kumacfile,"line %f %f %f %f\n",x2_3d_old,y2_3d_old,x2_3d,y2_3d);
            }
        }
    }
}

```

```

    }
    i_step++;
    z+=Z_CONV_STEP;
}
}
fclose(kumacfile);
return;
}

//*****Write BField*****
void write_bfield(struct coils* spulen, struct bfield_geom* geom, char* filename)
{
    int ix,iy,iz;
    double vec[3],b_all[3];
    char* bfield_filename[30];
    FILE *outfile;

    sprintf(bfield_filename,"%s.bfield",filename);
    outfile = fopen(bfield_filename, "w");
    printf("CALC BFIELD, output file %s\n",bfield_filename);
    if (outfile == (FILE *)0)
    {
        fprintf(stderr, "Can't write output file %s\n",bfield_filename);
        exit(1);
    }
    fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).x_min, (*geom).x_max, (*geom).stepx);
    fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).y_min, (*geom).y_max, (*geom).stepy);
    fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).z_min, (*geom).z_max, (*geom).stepz);
    fprintf(outfile, "*****\n");
    for(ix=0; ix <= (*geom).ixmax; ix++)
    {
        vec[0] = (*geom).x_min + ix*(*geom).stepx;
        printf("x: %12.2f\n", vec[0]);

        for(iy=0; iy <= (*geom).iymax; iy++)
        {
            vec[1] = (*geom).y_min + iy*(*geom).stepy;
            if (I_PRINT>1) printf("t: %12.2f\n", vec[1]);

            for(iz=0; iz <= (*geom).izmax; iz++)
            {
                vec[2] = (*geom).z_min + iz*(*geom).stepz;
                bfield3d_all(vec,b_all,spulen,geom);
                fprintf(outfile, "%12.5f %12.5f %12.5f %12.9f %12.9f %12.9f\n",
                    vec[0], vec[1],vec[2], b_all[0], b_all[1],b_all[2]);
            }
        }
    }
    fclose(outfile);
}

//***** write_bfield_from_potential *****
void write_bfield_from_potential(struct coils* spulen, struct bfield_geom* geom, char* filename)
{

```

```

int ix,iy,iz;
double vec[3],vec_ph[3], vec_mh[3], b_all[3];
char* bfield_filename[40];
FILE *outfile;

sprintf(bfield_filename,"%s.bfield_pot",filename);
outfile = fopen(bfield_filename, "w");
printf("CALC BFIELD, output file %s\n",bfield_filename);
if (outfile == (FILE *)0)
{
    fprintf(stderr, "Can't write output file %s\n",bfield_filename);
    exit(1);
}
fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).x_min, (*geom).x_max, (*geom).stepx);
fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).y_min, (*geom).y_max, (*geom).stepy);
fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).z_min, (*geom).z_max, (*geom).stepz);
fprintf(outfile, "*****\n");
for(ix=0; ix <= (*geom).ixmax; ix++)
{
    vec[0] = (*geom).x_min + ix*(*geom).stepx;
    printf("x: %12.2f\n", vec[0]);

    for(iy=0; iy <= (*geom).iymax; iy++)
    {
        vec[1] = (*geom).y_min + iy*(*geom).stepy;
        if (I_PRINT>1) printf("t: %12.2f\n", vec[1]);

        for(iz=0; iz <= (*geom).izmax; iz++)
        {
            vec[2] = (*geom).z_min + iz*(*geom).stepz;
            vec_ph[0] = vec[0]+H_DIFF;
            vec_mh[0] = vec[0]-H_DIFF;
            vec_ph[1] = vec[1];
            vec_mh[1] = vec[1];
            vec_ph[2] = vec[2];
            vec_mh[2] = vec[2];
            printf("vec_ph %f %f %f pot %f , vec_mh %f %f %f pot %f\n",
            vec_ph[0],vec_ph[1],vec_ph[2],potential3d_all(vec_ph,spulen,geom),
            vec_mh[0],vec_mh[1],vec_mh[2],potential3d_all(vec_mh,spulen,geom));
            b_all[0] = ( potential3d_all(vec_ph,spulen,geom)
            -potential3d_all(vec_mh,spulen,geom) )/(2.0*H_DIFF);
            vec_ph[0] = vec[0];
            vec_mh[0] = vec[0];
            vec_ph[1] = vec[1]+H_DIFF;
            vec_mh[1] = vec[1]-H_DIFF;
            vec_ph[2] = vec[2];
            vec_mh[2] = vec[2];
            b_all[1] = ( potential3d_all(vec_ph,spulen,geom)
            -potential3d_all(vec_mh,spulen,geom) )/(2.0*H_DIFF);
            vec_ph[0] = vec[0];
            vec_mh[0] = vec[0];
            vec_ph[1] = vec[1];
            vec_mh[1] = vec[1];
            vec_ph[2] = vec[2]+H_DIFF;
            vec_mh[2] = vec[2]-H_DIFF;
            b_all[2] = ( potential3d_all(vec_ph,spulen,geom)
            -potential3d_all(vec_mh,spulen,geom) )/(2.0*H_DIFF);

```

```

        fprintf(outfile, "%12.5f %12.5f %12.5f %12.9f %12.9f %12.9f\n",
                    vec[0], vec[1], vec[2], b_all[0], b_all[1], b_all[2]);
    }
}
fclose(outfile);
}

//***** calc_blines *****

int calc_blines(struct coils* spulen, struct bfield_geom* geom){
    const double step_factor = 0.1;
    int ibl, iy, iz, i, istep, istep_up, istep_down, back_turn;
    double vec[3], b_all[3], stepl, vec0_next, vec0_old, b_all0_old, btot;

    stepl = (*geom).stepx * step_factor;
    ibl = 0;
    for(iy=(*geom).iymin; iy <= (*geom).iymax; iy++)
    {
        for(iz=(*geom).izmin; iz <= (*geom).izmax; iz++)
        {
            printf("berechne B-Feldlinie %d\n", ibl);
            istep_up = (int)((*geom).x_max-(*geom).xbl)/(*geom).stepx;
            istep_down = (int)((*geom).xbl-(*geom).x_min)/(*geom).stepx;
            for (i=0; i<istep_up+istep_down+1; i++)
            {
                bl_x[ibl][i]=0;
                bl_y[ibl][i]=0;
                bl_z[ibl][i]=0;
                bl_b[ibl][i]=0;
            }
            vec[0] = (*geom).xbl;
            vec[1] = (*geom).y_min + iy*(*geom).stepy;
            vec[2] = (*geom).z_min + iz*(*geom).stepz;
            bfield3d_all(vec, b_all, spulen, geom);
            btot = absvalue(b_all);
            istep = istep_down;
            bl_x[ibl][istep] = vec[0];
            bl_y[ibl][istep] = vec[1];
            bl_z[ibl][istep] = vec[2];
            bl_b[ibl][istep] = btot;
            printf("bl %d %d %f %f %f %f\n", istep, ibl,
                bl_x[ibl][istep], bl_y[ibl][istep],
                bl_z[ibl][istep], bl_b[ibl][istep]);
            /* Doing upwards */
            back_turn=0;
            b_all[0] = 0;
            for (istep=1; istep<istep_up+1; istep++)
            {
                vec0_next = (*geom).xbl+(*geom).stepx*istep;
                if (I_PRINT>0) printf("vec0 %f ,vec0_next %f b_all[0] %f \n",
                    vec[0], vec0_next, b_all[0]);
                while ( (vec[0]<vec0_next) && (back_turn<1) )
                {
                    b_all0_old = b_all[0];
                    if (I_PRINT>0){

```

```

        printf("vor bfield3d vec %f %f %f, b_all %f %f %f \n",
               vec[0],vec[1],vec[2],b_all[0],b_all[1],b_all[2]);
    }
    //      print_par(spulen,geom);
    bfield3d_all(vec,b_all,spulen,geom);

    if (I_PRINT>0){
        printf("after bfield3d vec %f %f %f, b_all %f %f %f \n",
               vec[0],vec[1],vec[2],b_all[0],b_all[1],b_all[2]);
    }
    btot = absvalue(b_all);
    vec0_old = vec[0];
    vec[0] += stepl*b_all[0]/btot;
    vec[1] += stepl*b_all[1]/btot;
    vec[2] += stepl*b_all[2]/btot;
    if (b_all0_old*b_all[0]<0) back_turn=1;
}

printf("back_turn %d b_all0_old %f b_all[0] %f\n",back_turn,
       b_all0_old, b_all[0]);
if (back_turn<1)
{
    vec[1] -= ((vec[0]-vec0_next)/(vec[0]-vec0_old))
              *stepl*b_all[1]/btot;
    vec[2] -= ((vec[0]-vec0_next)/(vec[0]-vec0_old))
              *stepl*b_all[2]/btot;
    vec[0] = vec0_next;
    bl_x[ibl][istep_down+istep] = vec[0];
    bl_y[ibl][istep_down+istep] = vec[1];
    bl_z[ibl][istep_down+istep] = vec[2];
    bl_b[ibl][istep_down+istep] = btot;
    printf("bl2 %d %d %f %f %f\n",istep, ibl,
           bl_x[ibl][istep_down+istep],bl_y[ibl][istep_down+istep],bl_z[ibl][istep_down+istep]);
    } else {
        printf("back_turn %d b_all0_old %f b_all[0] %f\n",back_turn,
               b_all0_old, b_all[0]);
    }
}

/* Going downwards */
vec[0] = (*geom).xbl;
vec[1] = (*geom).y_min + iy*(*geom).stepy;
vec[2] = (*geom).z_min + iz*(*geom).stepz;
back_turn=0;
b_all[0] = 0;
for (istep=1;istep<istep_down+1;istep++)
{
    vec0_next = (*geom).xbl-(*geom).stepx*istep;
    while ( (vec[0]>vec0_next) && (back_turn<1) )
    {
        b_all0_old = b_all[0];
        bfield3d_all(vec,b_all,spulen,geom);
        btot = absvalue(b_all);
        vec0_old = vec[0];
        vec[0] -= stepl*b_all[0]/btot;
        vec[1] -= stepl*b_all[1]/btot;
        vec[2] -= stepl*b_all[2]/btot;
        if (b_all0_old*b_all[0]<0) back_turn=1;
    }
}

```

```

        if (back_turn<1)
    {
        vec[1] += ((vec[0]-vec0_next)/(vec[0]-vec0_old))
                  *step1*b_all[1]/btot;
        vec[2] += ((vec[0]-vec0_next)/(vec[0]-vec0_old))
                  *step1*b_all[2]/btot;

        vec[0] = vec0_next;
        bl_x[ibl][istep_down-istep] = vec[0];
        bl_y[ibl][istep_down-istep] = vec[1];
        bl_z[ibl][istep_down-istep] = vec[2];
        bl_b[ibl][istep_down-istep] = btot;
    }
    }
    ibl++;
}
}
printf("istep_max %d\n",istep_down+istep_up+1);
return istep_down+istep_up+1;
}

//***** write_blines *****

void write_blines(struct coils* spulen, struct bfield_geom* geom, char* filename,
                  int istep_max)
{
    int ibl, istep, iy, iz;
    char* blines_filename[30];
    FILE *outfile;
    double vec[3],vec_ph[3], vec_mh[3], b_all[3], diff_b;

    sprintf(blines_filename,"%s.blines",filename);
    outfile = fopen(blines_filename, "w");
    printf("CALC BLINIEN, output file %s\n",blines_filename);
    if (outfile == (FILE *)0){
        fprintf(stderr, "Can't write output file %s\n",blines_filename);
        exit(1);
    }

    for (istep=0;istep<istep_max;istep++)
    {
        ibl=0;
        fprintf(outfile, "%12f ", bl_x[ibl][istep]);
        for(iy=(*geom).iymin; iy <= (*geom).iymax; iy++)
        {
            for(iz=(*geom).izmin; iz <= (*geom).izmax; iz++)
            {
                if (I_BLINES_POT>0) {
                    vec[0] = bl_x[ibl][istep];
                    vec[1] = bl_y[ibl][istep];
                    vec[2] = bl_z[ibl][istep];
                    vec_ph[0] = vec[0]+H_DIFF;
                    vec_mh[0] = vec[0]-H_DIFF;
                    vec_ph[1] = vec[1];
                    vec_mh[1] = vec[1];
                    vec_ph[2] = vec[2];
                    vec_mh[2] = vec[2];
                    b_all[0] = ( potential3d_all(vec_ph,spulen,geom)

```

```

                                -potential3d_all(vec_mh,spulen,geom) )/(2.0*H_DIFF);
        vec_ph[0] = vec[0];
        vec_mh[0] = vec[0];
vec_ph[1] = vec[1]+H_DIFF;
vec_mh[1] = vec[1]-H_DIFF;
vec_ph[2] = vec[2];
vec_mh[2] = vec[2];
b_all[1] = ( potential3d_all(vec_ph,spulen,geom)
                                -potential3d_all(vec_mh,spulen,geom) )/(2.0*H_DIFF);
vec_ph[0] = vec[0];
vec_mh[0] = vec[0];
vec_ph[1] = vec[1];
vec_mh[1] = vec[1];
vec_ph[2] = vec[2]+H_DIFF;
vec_mh[2] = vec[2]-H_DIFF;
b_all[2] = ( potential3d_all(vec_ph,spulen,geom)
                                -potential3d_all(vec_mh,spulen,geom) )/(2.0*H_DIFF);
diff_b = absvalue(b_all)- bl_b[ibl][istep];
fprintf(outfile, "%12f %12f %12f",bl_y[ibl][istep],
                                bl_z[ibl][istep],
                                bl_b[ibl][istep],diff_b);
        } else {
            fprintf(outfile, "%12f %12f %12f",bl_y[ibl][istep],
                                bl_z[ibl][istep],
                                bl_b[ibl][istep]);
        }
        ibl++;
    }
}
    fprintf(outfile, "\n");
}
fclose(outfile);
}

```

```

//***** write_blines_kumac *****

```

```

void write_blines_kumac(struct coils* spulen, struct bfield_geom* geom,
                        char *filename){
    char* kumac_filename[30];
    char* blines_filename[30];
    FILE* kumacfile;
    char* y_str[2];
    char* z_str[2];
    char* b_str[2];
    int ibl,iy,iz,nbl;

    sprintf(kumac_filename,"%s.kumac",filename);
    printf("\nNow reading the magnetic parameter file %s...\n",kumac_filename);
    kumacfile = fopen(kumac_filename, "a");
    if (kumacfile == (FILE *)0) {
        fprintf(stderr, "Can't open the kumac file %s!\n",kumac_filename);
        exit(1);
    }
    ibl=0;
    for(iy=(*geom).iymin; iy <= (*geom).iymax; iy++){
        for(iz=(*geom).izmin; iz <= (*geom).izmax; iz++){

```

```

        ibl++;
    }
}
nbl = ibl;
if (nbl>10) nbl=10;
fprintf(kumacfile,"v/read x");
for (ibl=0;ibl<nbl;ibl++){
    sprintf(y_str,"y%1d",ibl);
    sprintf(z_str,"z%1d",ibl);
    sprintf(b_str,"b%1d",ibl);

    fprintf(kumacfile,"%s,%s,%s",y_str,z_str,b_str);
}
sprintf(blines_filename,"%s.blines",filename);
fprintf(kumacfile," %s\n",blines_filename);
fclose(kumacfile);
return;
}

//***** write_potential *****
// ASCII file
//
void write_potential(struct coils* spulen, struct bfield_geom* geom, char* filename)
{
    int ix,iy,iz;
    double vec[3],pot;
    char* potential_filename[30];
    FILE *outfile;

    sprintf(potential_filename,"%s.pot",filename);
    outfile = fopen(potential_filename, "w");
    printf("CALC BFIELD, output file %s\n",potential_filename);
    if (outfile == (FILE *)0)
    {
        fprintf(stderr, "Can't write output file %s\n",potential_filename);
        exit(1);
    }
    fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).x_min, (*geom).x_max, (*geom).stepx);
    fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).y_min, (*geom).y_max, (*geom).stepy);
    fprintf(outfile, "%12.2f %12.2f %12.2f\n", (*geom).z_min, (*geom).z_max, (*geom).stepz);
    fprintf(outfile, "*****\n");
    for(ix=0; ix <= (*geom).ixmax; ix++)
    {
        vec[0] = (*geom).x_min + ix*(*geom).stepx;
        printf("x: %12.2f\n", vec[0]);

        for(iy=0; iy <= (*geom).iymax; iy++)
        {
            vec[1] = (*geom).y_min + iy*(*geom).stepy;
            if (I_PRINT>1) printf("t: %12.2f\n", vec[1]);

            for(iz=0; iz <= (*geom).izmax; iz++)
            {
                vec[2] = (*geom).z_min + iz*(*geom).stepz;
                pot = potential3d_all(vec,spulen,geom);
            }
        }
    }
}

```



```

        fprintf(outfile, "%12.2f %12.2f% 12.2f %12.6f \n",  vec[0], vec[1],vec[2], pot);
    }
}
    }
    fclose(outfile);
    return;
}

//***** write_potential_pa *****
// binary file for SIMION
//

void write_potential_pa(struct coils* spulen, struct bfield_geom* geom, char* filename)
{
    int ix,iy,iz,i;
    double vec[3],pot, dummy;
    char* potential_filename[30];
    int f_out;
    struct header_3d h;
    FILE *first_open_outfile;

    sprintf(potential_filename,"%s.pa",filename);
    printf("CALC BFIELD, output file %s\n",potential_filename);
    first_open_outfile = fopen(potential_filename, "w");
    fclose(first_open_outfile);
    f_out = open(potential_filename,O_WRONLY||O_CREAT,0666);    /* write binary mode */
    h.mode = -1; // for SIMION 7.0 format
    h.symmetry = PLANAR;
    h.max_voltage = 100000.0;
    h.mirror = 1608; // 1608 for magnetic field (no mirror), add mirror conditions
    h.nx = (*geom).ixmax+1;
    h.ny = (*geom).iymax+1;
    h.nz = (*geom).izmax+1;
    write(f_out,&h,sizeof(h));

    i = 0;
    for(iz=0; iz <= (*geom).izmax; iz++)
    {
        vec[2] = (*geom).z_min + iz*(*geom).stepz;

        for(iy=0; iy <= (*geom).iymax; iy++)
        {
            vec[1] = (*geom).y_min + iy*(*geom).stepy;
            if (I_PRINT>1) printf("t: %12.2f\n", vec[1]);

            for(ix=0; ix <= (*geom).ixmax; ix++)
            {
                vec[0] = (*geom).x_min + ix*(*geom).stepx;
                if (I_PRINT>0) printf("x: %12.2f\n", vec[0]);
                pot = 100.0*potential3d_all(vec,spulen,geom);
                write(f_out,&pot,sizeof(pot));
                i++;
                if (modf( (double)(i)/100.0, &dummy) == 0.0)
                {
                    printf("i %d, x %f, y %f, z %f, pot %f\n",i,vec[0],vec[1],vec[2],pot);
                }
            }
        }
    }
}

```

```

    }
}

    }
    printf("Number of grid points:      %d\n",i);
    close(f_out);
    return;
}

//***** write_potential_pa *****
// binary file for SIMION
//

void write_potential_2d_pa(struct coils* spulen, struct bfield_geom* geom, char* filename)
{
    int ix,iy,iz,i;
    double vec[3],pot, dummy;
    char* potential_filename[30];
    int f_out;
    struct header_3d h;
    FILE *first_open_outfile;

    sprintf(potential_filename,"%s.pa",filename);
    printf("CALC BFIELD, output file %s\n",potential_filename);
    first_open_outfile = fopen(potential_filename, "w");
    fclose(first_open_outfile);
    f_out = open(potential_filename,O_WRONLY||O_CREAT,0666);    /* write binary mode */
    h.mode = -1; // for SIMION 7.0 format
    h.symmetry = CYLINDRICAL;
    h.max_voltage = 100000.0;
    h.mirror = 1610; // 1608 for magnetic field (no mirro), add mirror conditions
    h.nx = (*geom).ixmax+1;
    h.ny = (*geom).iymax+1;
    h.nz = 1;
    write(f_out,&h,sizeof(h));

    i = 0;
    vec[2] = 0.0;
    for(iy=0; iy <= (*geom).iymax; iy++)
    {
        vec[1] = (*geom).y_min + iy*(*geom).stepy;
        if (I_PRINT>1) printf("t: %12.2f\n", vec[1]);

        for(ix=0; ix <= (*geom).ixmax; ix++)
        {
            vec[0] = (*geom).x_min + ix*(*geom).stepx;
            printf("x: %12.2f\n", vec[0]);

            pot = 100.0*potential3d_all(vec,spulen,geom);
            write(f_out,&pot,sizeof(pot));
            i++;
            if (modf( (double)(i)/100.0, &dummy) == 0.0)
            {
                printf("i %d, x %f, y %f, z %f, pot %f\n",i,vec[0],vec[1],vec[2],pot);
            }
        }
    }
}

```

```

    }
    printf("Number of grid points:      %d\n",i);
    close(f_out);
    return;
}

```

A.4 sngl_coil_pot_2d

```

// Function for calculation of 2-dimensional magnetic potential of single coil
// according expansion by Ference Glueck
// Written: Feb.-April 2001 by B.Flatt
//

#include <stdio.h>
#include <math.h>
#include "b_struct.h"

#define CONVERGENCE_LIMIT 1E-10
#define N_MAX 100
#define mu_0 (M_PI/25000.0)          // 4pi/100000
#define IPRINT 0

double coil_pot(double r,double z,double radius, int coilnr, struct coils* spulen)
{
    double R_c, R_c_sqr;                //Radius of convergence
    double zMin, zMax;                  //lower and upper end of coil
    double z_zMin, z_zMax;              //z-zMin, z-zMax
    double z_zMin_sqr, z_zMax_sqr, min_sqr, max_sqr;
    double diff_potential, old_pot, potential, factor, on_axis_pot;
    double radius_sqr,C_min, C_max, r_div_R_c_fac;
    double fn_min, fn_max, gn_min, gn_max, sn_min, sn_max, Sn;
    double a_n, b_n, c_n, GO_min, GO_max;
    int n;

    zMin = (*spulen).position[coilnr] - (*spulen).length[coilnr]/2.0;
    zMax = (*spulen).position[coilnr] + (*spulen).length[coilnr]/2.0;

    z_zMin = z-zMin;
    z_zMax = z-zMax;
    z_zMin_sqr = z_zMin * z_zMin;
    z_zMax_sqr = z_zMax * z_zMax;

    radius_sqr = radius*radius;

    factor = 0.5*mu_0 * (*spulen).curr_div_thincoils[coilnr]
              /(*spulen).length[coilnr];
    if (IPRINT >2) printf("factor %f\n",factor);

    //on axis
    GO_min = sqrt(radius_sqr + z_zMin_sqr);
    GO_max = sqrt(radius_sqr + z_zMax_sqr);
    on_axis_pot = factor * ((GO_min - GO_max)-(*spulen).length[coilnr]);
    if (IPRINT >0) printf("on_axis_pot : %f\n", on_axis_pot );
}

```

```

//off axis
potential = 0.0;
if (r>0) {
    if (z < zMin) {
        R_c = sqrt(radius_sqr + z_zMin_sqr);
    } else {
        if (z > zMax) {
            R_c = sqrt(radius_sqr + z_zMax_sqr);
        } else {
            R_c = radius;
        }
    }
    // check whether convergence is fulfilled, otherwise potential=0
    if (r < R_c) {
        min_sqr = radius_sqr + z_zMin_sqr;
        max_sqr = radius_sqr + z_zMax_sqr;
        R_c_sqr = R_c*R_c;
        C_min = R_c_sqr/min_sqr;
        C_max = R_c_sqr/max_sqr;

        // n=0
        n=0;
        fn_min = 1/sqrt(min_sqr);
        fn_max = 1/sqrt(max_sqr);
        gn_min = - z_zMin*fn_min/min_sqr;
        gn_max = - z_zMax*fn_max/max_sqr;
        sn_min = 1.0/(n+1.0)*((2*n+1)*fn_min + z_zMin * gn_min);
        sn_max = 1.0/(n+1.0)*((2*n+1)*fn_max + z_zMax * gn_max);
        Sn = factor * (sn_min - sn_max);
        r_div_R_c_fac = r*r/R_c_sqr;
        potential = Sn * 1/(n+1) * r_div_R_c_fac;
        diff_potential = 1;

        //n>0
        n=1;
        while ( (diff_potential > CONVERGENCE_LIMIT) && (n<N_MAX) )
        {
            old_pot = potential;
            a_n = (4.0*n-1.0)/(4.0*n*n);
            b_n = (2.0*n-1.0)*(2.0*n-1.0)/(4.0*n*n);
            c_n = - (4.0*n+1.0)/R_c_sqr;
            fn_min = C_min * (a_n * z_zMin * gn_min + b_n * fn_min);
            fn_max = C_max * (a_n * z_zMax * gn_max + b_n * fn_max);
            gn_min = C_min * (c_n * z_zMin * fn_min + gn_min);
            gn_max = C_max * (c_n * z_zMax * fn_max + gn_max);
            sn_min = 1.0/(n+1.0)*((2*n+1)*fn_min + z_zMin * gn_min);
            sn_max = 1.0/(n+1.0)*((2*n+1)*fn_max + z_zMax * gn_max);
            Sn = factor * (sn_min - sn_max);
            r_div_R_c_fac *= r*r/R_c_sqr;
            potential += Sn * 1/(n+1) * r_div_R_c_fac;
            diff_potential = fabs(potential - old_pot);
            if (IPRINT >2) printf("loop %d potential: %f , old_potential: %f diff %f\n", n, potential, old_pot, diff_potential);
            n++;
        }
        if (IPRINT >0)
        {
            printf("loop %d potential: %f , old_potential: %f diff %f\n",

```

```

        n, potential, old_pot, diff_potential);
    }
}
potential *= -0.25*R_c_sqr;
return on_axis_pot + potential;
}

double thick_coil_pot(double r, double z, int coilnr, struct coils* spulen)
{
    int i;
    double potential, potential_sum, radius;

    potential_sum = 0;
    for (i=0; i <= (*spulen).number_of_thin_coils[coilnr]; i++)
    {
        radius = (*spulen).radius[coilnr] + (*spulen).delta_r[coilnr] * i;
        potential = coil_pot(r, z, radius, coilnr, spulen);
        if (IPRINT > 1) printf("i_thin_coil %d, potential %f\n", i, potential);
        if ( (i==0) || (i==(*spulen).number_of_thin_coils[coilnr]) )
    {
        potential *= 0.5;
    }
        potential_sum += potential;
    }
    if (IPRINT > 1) printf("thick_coil_pot: %f \n", potential_sum);
    return potential_sum;
}

```

A.5 snl_coil_bfield_2d

```

// Function for Calculation of 2-dimensional magnetic field of single coil
// based on old: bfield_11.c, spul_mag.f
// near axis: expansion by Ferenc Glueck
// Written: Feb.-April 2001 by B.Flatt, Ch.Weinheimer
//

#include <stdio.h>
#include <math.h>
#include "b_struct.h"
#define K_MAX 0.8 // switch between elliptical integrals from Abramowitz and
// expansion by Ferenc Glueck
#define CONVERGENCE_LIMIT 1E-10
#define N_MAX 100

/* z and r are to be given here in cm */

/***** bfield *****/
double mksa(double b_gauss) {
    return (b_gauss * 1e-5);
}

```

```

double int_elliptic(int num, double x) {
    int j;
    double dl, m1, dummi;

    /* Abramowitz (17.3.34) */
    const double a1[5] = {1.38629436112,
0.09666344259,
0.03590092383,
0.03742563713,
0.01451196212};
    const double b1[5] = {0.5,
0.12498593597,
0.06880248576,
0.03328355346,
0.00441787012};

    /* Abramowitz (17.3.36) */
    const double a2[5] = {1.0,
0.44325141463,
0.06260601220,
0.04757383546,
0.01736506451};
    const double b2[5] = {0.0,
0.24998368310,
0.09200180037,
0.04069697526,
0.00526449639};
    dummi = 0.0;
    m1 = 1.0 - x;
    dl = log(m1);
    for(j = 0; j < 5; j++){
        switch(num){
            case 1 :dummi += (a1[j] - dl * b1[j]) * pow(m1, j);
                break;
            case 3 :dummi += (b1[j]*(j*dl+1) - a1[j]*j) * pow(m1, j-1);
                break;
            case 2 :dummi += (a2[j] - dl * b2[j]) * pow(m1, j);
                break;
            case 4 :dummi += (b2[j]*(j*dl+1) - a2[j]*j) * pow(m1, j-1);
                break;
        }
    }
    return(dummi);
}

void brthe(double r, double theta, double radi, double *br, double *btheta) {

    double m, squart, brack1, brack2;

    squart = sqrt(radi*radi + r*r + 2.0*radi*r*sin(theta));

    /* Der Parameter f"ur die elliptischen Funktionen: */
    m = (4.0 * radi * r * sin(theta))/(squart*squart);
    brack1 = (2.0 - m) * int_elliptic(1, m) - 2.0 * int_elliptic(2, m);
    brack2 = (2.0 - m) * int_elliptic(3, m) - int_elliptic(1, m)
        - 2.0 * int_elliptic(4, m);
}

```

```

/* Die Radialkomponente der Feldstärke: */
*br = (brack1 + (4.0 - 2.0*m)*brack2)*radi/(r*tan(theta)*squart);

/* Die Theta-Komponente der Feldstärke: */
*btheta = ((r + radi*sin(theta))*brack1
           + (radi*radi - r*r)*m*brack2/r)/(-r*squart*sin(theta));
}

void b_loop(double current, double radius, double z, double rho,
           double* bz_ptr, double* brho_ptr){

/* Errechnet das Magnetfeld einer stromdurchflossenen Leiterschleife */
/* returns B-Field in Tesla, STROM in A, length units in cm */

int isign;
double dummi, r1, theta1, br1, bthe1, brho1, bz1;
double k,kz,k1,s,s2,s3,s4,sz,sr,a,a_old,c,g,g1,gprime,k2n,k2nm2;
double diff_g, g_old;
int n,nf;

if (rho == 0.0) {
/* (a) Der Punkt liegt auf der z-Achse */
/* vgl. S.Fl"ugge, Rechenmethoden der Elektrodynamik */
dummi = 1.0 / pow( sqrt(1.0+z*z/(radius*radius) ), 3);
bz1 = 2.0 * M_PI * dummi / radius;
brho1 = 0.0;
*bz_ptr = mksa(bz1) * current;
*brho_ptr = mksa(brho1) * current;
} else {
/* s<>0 since rho <>0 */
s = sqrt( (radius+rho)*(radius+rho)+ z*z);
s2= (radius+rho)*(radius+rho)+ z*z;
s3= s2*s;
s4= s2*s2;
k = 2.0 * sqrt(radius*rho)/s;
// expansion of Farence Glueck near axis
if (k<K_MAX){
sz = z/s;
sr = (radius+rho)/s;
kz = -sz/s*k;
k1 = 2*radius/(s*s) - k*k*sr/s;
g1 = 0.0;
g = 0.0;
a_old = M_PI/8.0;
k2n = 1.0;
k2nm2= 1.0;
diff_g = 1.0;
g_old = 0.0;
n = 0;
while ( (diff_g > CONVERGENCE_LIMIT) && (n<N_MAX) ){
nf = (float)n;
a = (2.0*nf+3.0)*(2.0*nf+3.0)/( (2.0*nf+4.0)*(2.0*nf+4.0) )*a_old;
a_old = a;
c = (1.0 - 2.0*(nf+2.0)*(nf+2.0)/((2.0*nf+3.0)*(2.0*nf+3.0))
    + 1.0/(2.0*nf+3.0) ) * a;

```

```

        g += c*k2n;
k2n = k2n*k*k;
if (n>0) {
    g1 += 2.0*nf*c*k2nm2;
    k2nm2 = k2nm2*k*k;
}
    n++;
    diff_g = fabs(g_old-g);
    g_old = g;
}
gprime = g1*k;
bz1 = 64.0*radius*radius*(g/s3+0.5*rho*(k1/s3*g1 -3.0*g*sr/s4));
brho1 = -64.0*radius*radius* 0.5*rho*(kz/s3*gprime-3.0*g*sz/s4);
} else {
    /* (b) Der Punkt liegt nicht auf der z-Achse */
    /* Wegen Schwierigkeiten der Berechnungen fuer negative z wird hier */
    /* nur mit positiven Werten von z gerechnet und das Vorzeichen am Schluss */
    /* bestimmt (Brho aendert das Vorzeichen und Bz nicht). */
    isign = 1;
    if(z < 0.0) {
        z = -z;
        isign = -1;
    }
    /* Transformation auf Kugelkoordinaten */
    if (z == 0.0) {
        r1 = rho;
        theta1 = 0.5 * M_PI;
    } else {
        r1 = sqrt(rho*rho + z*z);
        theta1 = atan(rho/z);
    }

    /* Berechnung der Feldst"arken */
    brthe(r1, theta1, radius, &br1, &bthe1);

    /* Ruecktransformation auf Zylinderkoordinaten */
    brho1 = br1 * sin(theta1) + bthe1 * cos(theta1);
    bz1 = br1 * cos(theta1) - bthe1 * sin(theta1);

    /* Vorzeichen von Brho und Bz siehe oben */
    brho1 = brho1*isign;
}
*bz_ptr = mksa(bz1) * current;
*brho_ptr = mksa(brho1) * current;
}
}

void b_coil(double z,double r, int coilnr, struct coils* spulen,
double *bz_ptr, double *br_ptr){

int i_rad, i_z, inside;
double z_dist, z_start, z_loop, radius, normalized_current, bz1, br1;

*br_ptr = 0.0;
*bz_ptr = 0.0;
if ( (z >= (*spulen).position[coilnr]-(*spulen).length[coilnr]/2.0)
    && (z <= (*spulen).position[coilnr]+(*spulen).length[coilnr]/2.0)

```



```

        && (r >= (*spulen).radius[coilnr])
        && (r <= (*spulen).radius[coilnr] + (*spulen).thickness[coilnr])    ){
    inside = 1;
} else {
    inside = 0;

    /* step parameter in z-direction */
    z_dist = (*spulen).length[coilnr]/(*spulen).windings[coilnr];
    z_start = (*spulen).position[coilnr] - (*spulen).length[coilnr]/2.0;

    /* normalize current such, that bfield is independent on #windings, #thincoils */
    normalized_current = (*spulen).curr_div_thincoils[coilnr]/(*spulen).windings[coilnr];

    /* integrate over different radii */
    for (i_rad=0; i_rad <= (*spulen).number_of_thin_coils[coilnr]; i_rad++){
radius = (*spulen).radius[coilnr] + (*spulen).delta_r[coilnr] * i_rad;
/* integrate over different windings */
    for (i_z = 0; i_z <= (*spulen).windings[coilnr]; i_z++){
        z_loop = z_start + i_z*z_dist;
        b_loop(normalized_current, radius, z-z_loop, r, &bz1, &br1);
        /* weight boarder of integrations by 0.5 (trapez integration) */
        if ( (i_rad==0) || (i_rad==(*spulen).number_of_thin_coils[coilnr]) ){
            bz1 *= 0.5;
            br1 *= 0.5;
        }
        if ( (i_z==0) || (i_z==(*spulen).windings[coilnr]) ){
            bz1 *= 0.5;
            br1 *= 0.5;
        }
        *bz_ptr += bz1;
        *br_ptr += br1;
    }
}
    }
    }
    return;
}

```

A.6 math_vector

```

// Bibliothekname: math_vector.c
// written by Thomas Thümmeler in January 2001
// Version of 02/07/2001
// extended by B.Flatt on 02/13/01
// new features by T.Thümmeler on 02/21/01
// math_vector.c supplies math function to work with vectors
//
#include <math.h>
#include <stdio.h>

int signum(double x){
    if (x>=0) {

```

```

        return +1;
    } else {
        return -1;
    }
}

double absvalue(double *vector)
{
    double value;

    value = sqrt(vector[0]*vector[0] + vector[1]*vector[1] + vector[2]*vector[2]);

    return value;
}

double scalar_prod(double *vec1, double *vec2)
{
    double value;

    value = vec1[0]*vec2[0]+vec1[1]*vec2[1]+vec1[2]*vec2[2];

    return value;
}

void vector_times_scalar(double *vec, double value, double *result)
{
    int index;

    for (index = 0; index <= 2; index++)
    {
        result[index] = vec[index] * value;
    }
}

void vector_sum(double *vec1, double *vec2, double *sum)
{
    int index;

    for (index = 0; index <= 2; index++)
    {
        sum[index] = vec1[index] + vec2[index];
    }
}

double angle(double *vec1, double *vec2)
{
    double value;

    value = (180./M_PI)*acos((scalar_prod(vec1,vec2))/(absvalue(vec1)*absvalue(vec2)));

    return value;
}

double angle_rad(double *vec1, double *vec2)
{
    double value;

```

```

    value = acos((scalar_prod(vec1,vec2))/(absvalue(vec1)*absvalue(vec2)));

    return value;
}

void cross_prod(double *vec1,double *vec2,double *vec3)
{
    vec3[0] = (vec1[1]*vec2[2] - vec1[2]*vec2[1]);
    vec3[1] = (vec1[2]*vec2[0] - vec1[0]*vec2[2]);
    vec3[2] = (vec1[0]*vec2[1] - vec1[1]*vec2[0]);
}

void vector_curvation(double *p12,double *p23,double *radius)
{
    double k0[3],p13[3],n[3],n1[3],n2[3];
    double a;
    int komp;

    for (komp = 0; komp <= 2; komp++)
    {
        p12[komp] = 0.5 * p12[komp];
        p23[komp] = 0.5 * p23[komp];
        // vectors to the centre
        p13[komp] = p12[komp]+p23[komp];
        // vector from origin of p12 to destination of p23
    }

    cross_prod(p12,p23,n);           // calculate normal vector

    cross_prod(n,p12,n1);           // calculate perpendicular vector of
    // point to point vector 1 in normal plane

    cross_prod(n,p23,n2);           // calculate perpendicular vector of
    // point to point vector 2 in normal plane

    a = 0;
    if ((n1[0]*n2[1] - n1[1]*n2[0]) != 0)
    {
        a = (p13[0]*n2[1] - p13[1]*n2[0]) / (n1[0]*n2[1] - n1[1]*n2[0]);
    }
    else
    {
        if ((n1[0]*n2[2] - n1[2]*n2[0]) != 0)
        {
            a = (p13[0]*n2[2] - p13[2]*n2[0]) / (n1[0]*n2[2] - n1[2]*n2[0]);
        }
        else
        {
            if ((n1[1]*n2[2] - n1[2]*n2[1]) != 0)
            {
                a = (p13[1]*n2[2] - p13[2]*n2[1])
                    / (n1[1]*n2[2] - n1[2]*n2[1]);
            }
        }
    }
    // calculates the multiplication faktor of perpendicular vector 1 to

```

```

    // the crosspoint of both perpendicular vectors

    for (komp=0; komp<=2; komp++)
    {
        k0[komp] = p12[komp] + a*n1[komp];          // calculate origin of radius vector
        radius[komp] = 2.*p12[komp] - k0[komp];      // calculate radius vector
    }

    radius[3] = 1./absvalue(radius);                // store curvation value in fourth component
}

void point_curvation(double *x1,double *x2,double *x3,double *radius)
{
    double p12[3],p23[3];
    int komp;

    for (komp = 0; komp <= 2; komp++)
    {
        p12[komp] = (x2[komp] - x1[komp]);          // calculate point to point vectors
        p23[komp] = (x3[komp] - x2[komp]);
    }

    vector_curvation(p12,p23,radius);
}

void rotate_vec(double *vec1,double *omega, double *vec3)
// rotates vector vec1 (output vec3) around rotation axis omega
// by angle |omega|
// see Bronstein: section 2.6.5.2.3, page 215,216
{
    double theta, stheta, ctheta, one_ctheta, alpha, beta, gamma;

    theta = absvalue(omega);
    ctheta = cos(theta);
    one_ctheta = 1.0-cos(theta);
    stheta = sin(theta);
    alpha = omega[0]/absvalue(omega);
    beta = omega[1]/absvalue(omega);
    gamma = omega[2]/absvalue(omega);
    // printf("alpha %f, beta %f gamma %f theta %f\n", alpha, beta, gamma, theta);
    vec3[0] = vec1[0]*(ctheta + alpha*alpha*one_ctheta) +
        vec1[1]*(gamma*stheta + alpha*beta*one_ctheta) +
        vec1[2]*(-beta*stheta + alpha*gamma*one_ctheta);
    vec3[1] = vec1[0]*(-gamma*stheta + alpha*beta*one_ctheta) +
        vec1[1]*(ctheta + beta*beta*one_ctheta) +
        vec1[2]*(alpha*stheta + beta*gamma*one_ctheta);
    vec3[2] = vec1[0]*(beta*stheta + gamma*alpha*one_ctheta) +
        vec1[1]*(-alpha*stheta + gamma*beta*one_ctheta) +
        vec1[2]*(ctheta + gamma*gamma*one_ctheta);
    return;
}

void rotate_vec_on_z_axis(double *vec1,double *vec2, double *vec3)
// rotates vector vec1 (output vec3) in such a way,
// that vec2 points in z-direction

```

```

// see Bronstein: section 2.6.5.2.3, page 215,216
{
    double vecz[3],rot_axis[3];
    double theta, stheta, ctheta, one_ctheta, alpha, beta, gamma, abs_rot_axis;
    vecz[0] = 0.0;
    vecz[1] = 0.0;
    vecz[2] = 1.0;
    theta = angle_rad(vec2,vecz);
    ctheta = cos(theta);
    one_ctheta = 1.0-cos(theta);
    stheta = sin(theta);
    cross_prod(vecz,vec2,rot_axis);
    abs_rot_axis = absvalue(rot_axis);
    // printf("rot_axis %f %f %f\n",rot_axis[0],rot_axis[1], rot_axis[2]);
    if (abs_rot_axis==0.0) {
        vec3[0] = vec1[0];
        vec3[1] = vec1[1];
        vec3[2] = vec1[2];
    } else {
        alpha = rot_axis[0]/abs_rot_axis;
        beta = rot_axis[1]/abs_rot_axis;
        gamma = rot_axis[2]/abs_rot_axis;
        //printf("alpha %f, beta %f gamma %f theta %f\n", alpha, beta, gamma, theta);
        vec3[0] = vec2[0]*(ctheta + alpha*alpha*one_ctheta) +
            vec2[1]*(gamma*stheta + alpha*beta*one_ctheta) +
            vec2[2]*(-beta*stheta + alpha*gamma*one_ctheta);
        vec3[1] = vec2[0]*(-gamma*stheta + alpha*beta*one_ctheta) +
            vec2[1]*(ctheta + beta*beta*one_ctheta) +
            vec2[2]*(alpha*stheta + beta*gamma*one_ctheta);
        vec3[2] = vec2[0]*(beta*stheta + gamma*alpha*one_ctheta) +
            vec2[1]*(-alpha*stheta + gamma*beta*one_ctheta) +
            vec2[2]*(ctheta + gamma*gamma*one_ctheta);
        printf("rotated vec2: %f %f %f\n",vec3[0],vec3[1],vec3[2]);
        vec3[0] = vec1[0]*(ctheta + alpha*alpha*one_ctheta) +
            vec1[1]*(gamma*stheta + alpha*beta*one_ctheta) +
            vec1[2]*(-beta*stheta + alpha*gamma*one_ctheta);
        vec3[1] = vec1[0]*(-gamma*stheta + alpha*beta*one_ctheta) +
            vec1[1]*(ctheta + beta*beta*one_ctheta) +
            vec1[2]*(alpha*stheta + beta*gamma*one_ctheta);
        vec3[2] = vec1[0]*(beta*stheta + gamma*alpha*one_ctheta) +
            vec1[1]*(-alpha*stheta + gamma*beta*one_ctheta) +
            vec1[2]*(ctheta + gamma*gamma*one_ctheta);
    }
    return;
}

void spherical_coordinate(double *vec,double* r, double* theta, double* phi)
// gives back from vector vec spherical coordinates
{
    double vecz[3];

    vecz[0] = 0.0;
    vecz[1] = 0.0;
    vecz[2] = 1.0;
    printf("vec %f %f %f theta %f\n",vec[0],vec[1],vec[2],angle(vec,vecz));
}

```

```
*r = absvalue(vec);
*theta = angle_rad(vec,vecz);
if (vec[1]==0.0) {
    *phi=signum(vec[0])*M_PI/2.0;
} else {
    if (vec[1]>0) {
        *phi = atan(vec[0]/vec[1]);
    } else {
        *phi = M_PI + atan(vec[0]/vec[1]);
    }
}
return;
}
```


Literaturverzeichnis

- [Ath95] C. Athanassopoulos et al., Phys. Rev. Lett. **75** (1995) 2650
- [Bon99] J. Bonn, L. Bornschein, B. Degen, E.W. Otten, Ch. Weinheimer, NIM **A421** (1999) 256
- [Bon01] J. Bonn, et al. Nuclear Physics B 91 (2001) 273-276
- [Bor00] B. Bornschein, Dissertation, Inst. f. Physik, Universität Mainz, 2000
- [Die00] DARK 2000, Heidelberg, Vortrag von A. Dietz, Juli 2000
- [Don00] Long Baseline News, Juli 2000, <http://hepunix.rl.ac.uk/minos/longbnews/0007.html>
- [Eit00] K. Eitel, New Jour. Phys. **2** (2000) 1
- [Ell87] S.R. Elliot et al., Phys. Rev. Lett. **59** (1987) 2020
- [Fel98] G.J. Feldmann and R.D. Cousins, Phys. Rev. **D57** (1998) 3873
- [Fer34] E. Fermi, Z. Phys. **88** (1934) 161
- [Fic00] L. Fickinger, Diplomarbeit, Inst. f. Physik, Universität Mainz, 2000
- [Fuk98] Y. Fukuda et al., Phys. Rev. Lett. **81** (1998) 1562
- [Fuk99] Y. Fukuda et al., Phys. Lett. **B467** (1999) 185
- [Glu00] F. Glueck, Persönliche Mitteilung, Veröffentlichung in Vorbereitung
- [Jac82] J.D. Jackson, Klassische Elektrodynamik, W. de Gruyter, Berlin 1982
- [Kat01] A next generation Tritium Beta Decay Experiment, <http://www-ik1.fzk.de/tritium>
- [Lob99] V.M. Lobashev et al., Phys. Lett. **B460** (1999) 227
- [Pau30] W. Pauli, Brief an die Physikalische Gesellschaft in Tübingen vom 4.12.1930, nachzulesen in L.M. Brown, Phys. Today **31** (1978) 23
- [Paw95] PAW, Physics Analysis Workstation, Application Software Group Computing and Networks Group, CERN, 1995

- [Pic92a] A. Picard, H. Backe, H. Barth, J. Bonn, B. Degen, Th. Edling, R. Haid, A. Hermann, P. Leiderer, Th. Loeken, A. Molz, R.B. Moore, A. Osipowicz, E.W. Otten, M. Przyrembel, M. Schrader, M. Steininger and Ch. Weinheimer, NIM **B63** (1992) 345
- [Thu01] T. Thümmeler, Diplomarbeit, Inst. f. Physik, Universität Mainz, voraussichtliche Fertigstellung 2001
- [Rei59] F. Reines, C.L. Cowan, Phys. Rev. **113** (1959) 273
- [Sch97] N. Schmitz, 'Neutrino-physik', Teubner Verlag, Stuttgart, 1997
- [Sch01] J.-P. Schall, Diplomarbeit, Inst. f. Physik, Universität Mainz, in Vorbereitung
- [SIM] <http://www.srv.net/~klack/simion.html>
- [Far01] Y. Farzan, O.L.G. Peres, A.Yu. Smirnov hep-ph/0105105
- [Tur99] M.S. Turner, astro-ph/9912211
- [Ulr00] H. Ulrich, Diplomarbeit, Inst. f. Physik, Universität Mainz, 2000
- [Wei99] Ch. Weinheimer, B. Degen, A. Bleile, J. Bonn, L. Bornschein, O. Kazachenko, A. Kovalik, E.W. Otten, Phys. Lett. **B460** (1999) 219