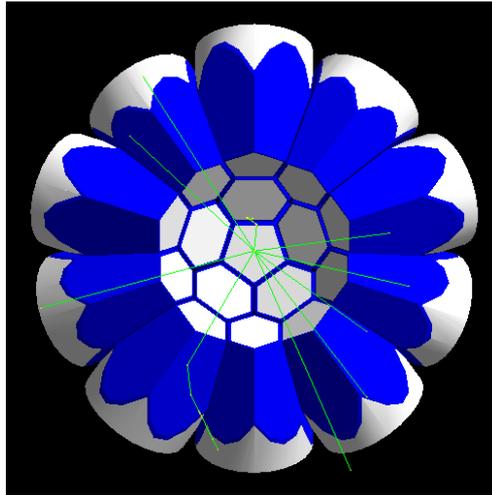


- Bachelor Arbeit -  
**Simulation eines  $4\pi$  BaF<sub>2</sub>-Kalorimeters  
mit GEANT4**



von

Emilio Mevius

J.W.Goethe Universität Frankfurt am Main  
Institut für Angewandte Physik  
Experimentelle Astrophysik

Betreuerin:  
Dr. Kerstin Sonnabend

November 2013

# Inhaltsverzeichnis

Abbildungsverzeichnis . . . . .	I
Tabellenverzeichnis . . . . .	II
<b>1. Einleitung</b>	<b>1</b>
<b>2. Grundlagen</b>	<b>3</b>
2.1. Typische Wechselwirkungen von Elektronen und Photonen mit Materie . . . . .	3
2.1.1. Elektronen . . . . .	3
2.1.2. Photonen . . . . .	3
2.2. Realer Aufbau des Kalorimeters . . . . .	4
2.3. Funktionsweise des Detektors . . . . .	8
2.3.1. Szintillation . . . . .	8
2.3.2. Photomultiplier . . . . .	8
<b>3. Software Werkzeuge</b>	<b>10</b>
3.1. Geant4 . . . . .	10
3.2. HepRep . . . . .	10
3.3. OpenGL . . . . .	11
3.4. ROOT . . . . .	11
<b>4. Implementierung des Kalorimeters</b>	<b>12</b>
4.1. Materialien . . . . .	12
4.2. Geometrie der einzelnen Detektoren . . . . .	12
4.2.1. Zylinder . . . . .	13
4.2.2. Verkürzter Zylinder . . . . .	14
4.2.3. Kristall A . . . . .	14
4.2.4. Kristall A mit Schnitt in den Seiten 1 und 5 im Uhrzeigersinn . . . . .	15
4.2.5. Kristall B . . . . .	16
4.2.6. Kristall B mit Schnitten . . . . .	17
4.3. Zusammensetzung des kompletten Kalorimeters . . . . .	18
4.3.1. Strategie . . . . .	18
4.3.2. Implementierung in Geant4 . . . . .	19
4.3.2.1. Platzierung der Volumina im Muster . . . . .	21
4.3.2.2. Platzierung der Abbildungen eines Musters im Hauptvolumen . . . . .	22
4.4. Sensitives Detektorvolumen . . . . .	22
4.5. Quelle . . . . .	24
4.5.1. Punktquelle . . . . .	24
4.5.2. Aktive Scheibe . . . . .	24
4.6. Physische Prozesse . . . . .	24
4.7. Antwort der Simulation . . . . .	25

<b>5. Ergebnisse</b>	<b>27</b>
5.1. Vergleich mit den gemessenen Daten aus der B.Sc.-Arbeit von Clemens Wolf . . .	33
5.2. Vergleich mit Simulationen in Geant3 . . . . .	33
<b>6. Weitere Möglichkeiten der Simulation</b>	<b>36</b>
<b>Literaturverzeichnis</b>	<b>39</b>
<b>Anhang</b>	<b>39</b>
<b>A. Darstellung der Größen</b>	<b>40</b>
<b>B. Rechnung der Größen</b>	<b>42</b>

# Abbildungsverzeichnis

1.1. Aufbau von FRANZ (Frankfurt Neutron Source) . . . . .	2
2.1. Aufbau eines Detektors . . . . .	5
2.2. Segemente des Detektoraufbaus . . . . .	6
2.3. Eingangsstelle mit Schlitz daneben . . . . .	7
2.4. Ansicht des Kalorimeters von der Ausgangsstelle betrachtet . . . . .	7
2.5. Schema eines Photomultipliers . . . . .	9
4.1. Ansicht des Kristalls A . . . . .	14
4.2. Geschnittenes Pentagon in den Seiten 1 und 5 im Uhrzeigersinn . . . . .	15
4.3. Verfahren für die Folien vom geschnittenen Pentagon . . . . .	16
4.4. Ansicht des Kristalls B . . . . .	17
4.5. Geschnittene Hexagone . . . . .	18
4.6. Muster beim Aufbau des Kalorimeters . . . . .	19
4.7. Entwurf für die Zusammensetzung des Kalorimeters . . . . .	20
4.8. OpenGL-Bild mit einer Ansicht des gesamten Kalorimeters . . . . .	23
5.1. Energiespektren für Punktquellen . . . . .	29
5.2. Energiespektren für die realen Quellen . . . . .	31
5.3. Typische Flugbahnen beim Zerfall von $^{60}\text{Co}$ . . . . .	32
5.4. Typische Flugbahnen beim Zerfall von $^{22}\text{Na}$ . . . . .	32
5.5. Vergleich mit Spektren von Clemens Wolf . . . . .	34
5.6. Multiplizität-Histogramm von Clemens Wolf . . . . .	35
5.7. Vergleich der Geant3-Spektren mit realen Messungen . . . . .	35
6.1. Dimensionierung des inneren Radius und der Detektorbreite . . . . .	37
A.1. Größen des Kristalls A . . . . .	40
A.2. Größen des Kristalls B . . . . .	40
A.3. Abdeckungbezogene Größen . . . . .	41
A.4. Schlitzbezogene Größen . . . . .	41

# Tabellenverzeichnis

2.1. Dimensionen der Segmente des Detektors . . . . .	6
4.1. Angewendete Materialien in der Simulation . . . . .	13
5.1. Ergebnisse der Simulation einer Punktquelle mit den Energien von $^{60}\text{Co}$ . . . . .	27
5.2. Ergebnisse der Simulation einer Punktquelle mit den Energien von $^{22}\text{Na}$ . . . . .	28
5.3. Ergebnisse der Simulation realer Quellen im gesamten Kalorimeter . . . . .	28
5.4. Ergebnisse der Simulation realer Quellen pro Detektor . . . . .	30
6.1. Entwicklung der Eigenschaften des Kalorimeters in der Geant3-Simulation . . . . .	38
6.2. Entwicklung der Eigenschaften des Kalorimeters in dieser Simulation . . . . .	38

# 1. Einleitung

Der Einsatzort des in dieser Arbeit behandelten  $\text{BaF}_2$ -Kalorimeters ist die Johann Wolfgang Goethe-Universität in Frankfurt am Main. Hier wird FRANZ (Frankfurt Neutron Source) aufgebaut, an der neutronen- und protonen- induzierte Reaktionen im Rahmen der Astrophysik untersucht werden soll.

In der Abbildung 1.1 ist der Aufbau von FRANZ dargestellt. Ein Hochfrequenz-Beschleuniger erzeugt zunächst einen Protonenstrahl, der auf eine  ${}^7\text{Li}$ -Schicht trifft, um Neutronen über die Reaktion  ${}^7\text{Li}(p,n){}^7\text{Be}$  zu erzeugen. Ein intensiver kollimierter Neutronenstrahl wird danach erzeugt, mit dem Experimente im Energiebereich von 10 bis zu 400 keV geplant sind [1].

Der Neutronenstrahl soll auf eine Probe treffen, die von einem  $4\pi$   $\text{BaF}_2$ -Kalorimeter umgeben ist. Das Kalorimeter besteht aus einzelnen  $\text{BaF}_2$ -Einkristallen, die als Szintillationsdetektoren wirken und zum Nachweis von  $\gamma$ -Emissionen nach einem Neutroneneinfang genutzt werden, um die  $(n,\gamma)$  Wirkungsquerschnitte der Reaktionen zu messen. Der Neutronenfluss wird voraussichtlich  $4 \cdot 10^7 \frac{n}{s \cdot \text{cm}^2}$  am Probenort betragen [2].

Der Aufbau wird für die nukleare Astrophysik genutzt, nämlich zur Untersuchung des Ursprungs schwerer Elemente im S-Prozess. Zudem ist auch geplant proton-induzierte Experimente durchzuführen, in denen verschiedene Proben direkt mit dem Protonenstrahl bestrahlt werden.

Das Ziel dieser Arbeit ist, das  $4\pi$   $\text{BaF}_2$ -Kalorimeter zu simulieren und die Ergebnisse mit vorherigen Simulationen und später mit den in Versuchen gesammelten Daten zu vergleichen. Dafür werden die deponierte Energie im gesamten Kalorimeter und im jedem einzelnen Detektor, sowie die Multiplizität, Effizienz und andere Eigenschaften verglichen. Früher wurde gezeigt [3], dass die vom CERN entwickelte Software GEANT für diese Versuche geeignet ist. Eine Simulation wurde dann in der Version GEANT3 gemacht. Mittlerweile hat sich in der internationalen Zusammenarbeit eine weiter entwickelte Programmiersprache wie C++ gegenüber FORTRAN durchgesetzt. Die Aufgabe dieser Arbeit ist die Implementierung einer Simulation vom FRANZ-Detektor mit der Software GEANT4.

Aus der Übereinstimmung der Simulationen mit den experimentellen Daten kann man zudem Informationen gewinnen, ob die Simulation den realen Gegebenheiten entspricht, sowie für die Bestimmung von passenden Quellen, Maßen und Dimensionierungen.

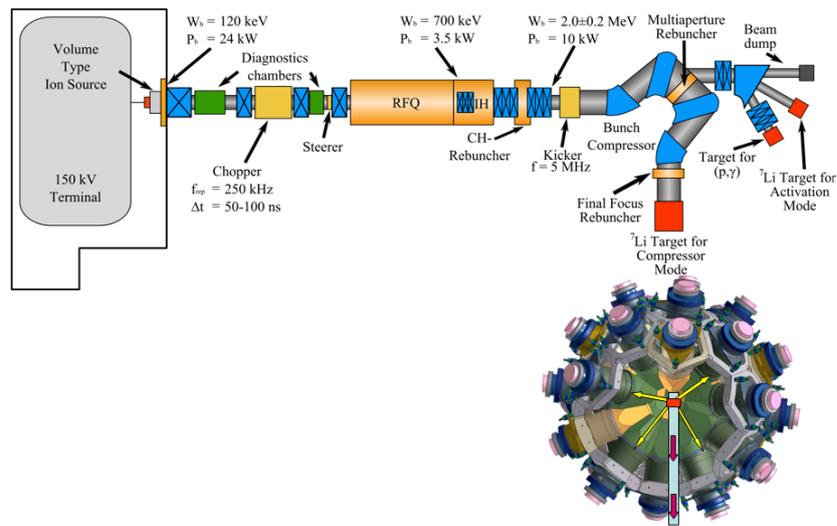


Abbildung 1.1.: Eine Ionenquelle liefert einen kontinuierlichen Protonenstrahl mit 120 keV und 200 mA, links in der Abbildung. Rot markiert sind die zu bestrahlenden Proben. Rechts die Stelle der Probe, die direkt mit dem Protonenstrahl bestrahlt werden soll und unten, in Richtung zum kugelförmigen Detektor, die  ${}^7\text{Li}$ -Probe, die mit einem Protonenstrahl mit einer einstellbaren Endenergie zwischen 1.8 und 2.2 MeV bestrahlt wird. Der gesamte Beschleuniger ist mit Betonblöcken bedeckt, die zusätzlich mit einer Schicht von boriiertem Polyethylen ausgekleidet sind, um gegen die Strahlung zu schützen [1].

## 2. Grundlagen

### 2.1. Typische Wechselwirkungen von Elektronen und Photonen mit Materie

#### 2.1.1. Elektronen

Es gibt zwei Möglichkeiten. Ein Elektron kann aufgrund der elektromagnetischen Wechselwirkung mit einem Atomkern von seiner Bahn gebeugt und gebremst werden. Dabei senden sie die verlorene Energie als sogenannte Bremsstrahlung aus. Zum anderen kann ein Elektron gegen die Elektronen eines Atoms stoßen und dadurch Energie an diese Elektronen abgeben. Die Atome können damit entweder ionisiert oder angeregt werden. Falls sie angeregt werden, werden danach bei der Abregung Photonen emittiert [2] [4].

#### 2.1.2. Photonen

Es gibt drei mögliche Wechselwirkungen: Der Photoelektrische Effekt, die Compton-Streuung und die Paarbildung.

**Photoelektrischer Effekt:** Absorbiert ein Elektron die Energie eines Photons und diese beträgt mindestens die Bindungsenergie des Elektrons, wird das Elektron aus seiner Bindung herausgelöst. Drei Fälle sind dabei zu unterscheiden [4]:

- Äußerer Photoeffekt: Falls die Elektronen aus der Oberfläche von Halbleitern oder Metallen durch Bestrahlung herausgelöst werden.
- Innerer Photoeffekt: Exklusiv in Halbleitern. Man unterscheidet zwischen Photoleitung und Photovoltaischem Effekt. Im ersten Fall wird ein Elektron aus seiner Bahn herausgelöst, wobei ein Elektron-Loch-Paar gebildet wird, dass die Leitfähigkeit des Halbleiters steigt. Im zweiten Fall findet eine Umwandlung von Licht in elektrische Energie statt.
- Photoionisation: Die Atome verlieren durch die Bestrahlung Elektronen und werden ionisiert.

**Compton-Effekt:** In diesem Fall wird ein hochenergetisches Photon seine Energie an einem Elektron teilweise so abgeben, dass es durch einen elastischen Stoß gestreut wird, während das Elektron aus dem Atom herausgeschlagen wird. Dieses Elektron wechselwirkt anschließend weiter mit der Umgebung. Das Photon erfährt durch die Streuung eine Vergrößerung der Wellenlänge,

die vom Streuwinkel  $\phi$  nach der folgenden Beziehung abhängig ist:

$$\Delta\lambda = \frac{h}{m_e \cdot c} \cdot (1 - \cos(\phi)) \quad (2.1)$$

In der Spektroskopie kann man ein sogenanntes Compton-Kontinuum erkennen. Wird ein Photon durch den Compton-Effekt aus dem Detektor herausgeschlagen, gibt es nur einen Bruchteil seiner Energie ab. Der Anteil der Energie, der im Detektor deponiert wird, ist vom Streuwinkel des Photons nach der Beziehung 2.1 abhängig. So werden in den Energiespektren typischerweise eine Reihe von etwa gleich wahrscheinlichen Energien detektiert, die den verschiedenen Streuwinkeln der Photonen entsprechen. Die höchste Energie, die durch den Compton-Effekt deponiert werden kann, wird als Compton-Kante bezeichnet [4].

**Paarbildung:** Ein Photon kann in ein Elektron-Positron-Paar übergehen, falls es genügend Energie besitzt. Diese Schwelle beträgt 1022 keV, was der zweifachen Ruhemasse eines Elektrons entspricht. Aufgrund der Impulserhaltung kann die Paarbildung nur durch Wechselwirkung eines Photons im elektrischen Feld eines Atomkerns oder eines Hüllenelektrons stattfinden [2] [4].

## 2.2. Realer Aufbau des Kalorimeters

Das Ziel des Aufbaus ist es, eine Geometrie zu finden, bei der der gesamte Winkelraum abgedeckt wird und bei der jeder einzelne Kristall denselben Raumwinkel abdeckt. Das in Ref. [3] präsentierte Modell besteht aus 42 Kristallen in Form eines Pyramidenstumpfes. Davon haben 30 ein Hexagon als Grundfläche und 12 ein Pentagon.

Diese Anordnung bringt die zwei Voraussetzungen in Einklang, eine möglichst geringe Modulanzahl zu haben, um die Verluste durch die Lücken zwischen den Detektoren möglichst klein zu halten, und die verfügbare Größe der BaF<sub>2</sub>-Kristalle zu erreichen [5].

In der Abbildung 2.1 ist ein einzelner Detektor schematisch dargestellt.

Um annähernd eine Kugel mit diesen 42 Detektoren zu schaffen, sind die Maße der Grundflächen der Pyramiden mit dem entsprechenden Radius der Kugel folgendermaßen, wie in der Abbildung 2.2 und in der Tabelle 2.1 gezeigt, verknüpft:

Die Kugel hat einen inneren Radius von 10 cm und der Detektor hat eine Dicke von 15 cm, wobei sich diese Maße, wie im Kapitel 6 gezeigt, für diese Simulation auch anpassen lassen. Die Kugel hat demnach einen äußeren Radius von 25 cm. Im Prinzip sind diese Maße ausgewählt, damit ein Gleichgewicht zwischen einem möglichst geringen Szintillatorvolumen, zur Minimierung der Hintergrundstrahlung, und der möglichst großen Oberfläche, um die Anzahl der verwendeten Kristalle und damit die Anzahl der Zwischenräume so klein wie möglich zu halten, entsteht [5].

Um die zylindrischen Photomultiplier (PMT) an die Kristalle anzubringen, haben alle Teile einen zylindrischen Schnitt von 14 cm Durchmesser an der äußeren Grenze.

Zwischen den Kristallen gibt es einen Trennraum von 2 mm. Dieser besteht aus einer Schicht von 0.5 mm Teflon und 0.1 mm Aluminium pro Kristall plus 0.8 mm Luft. Das Teflon sorgt für eine Verminderung der Reflektionsverluste.

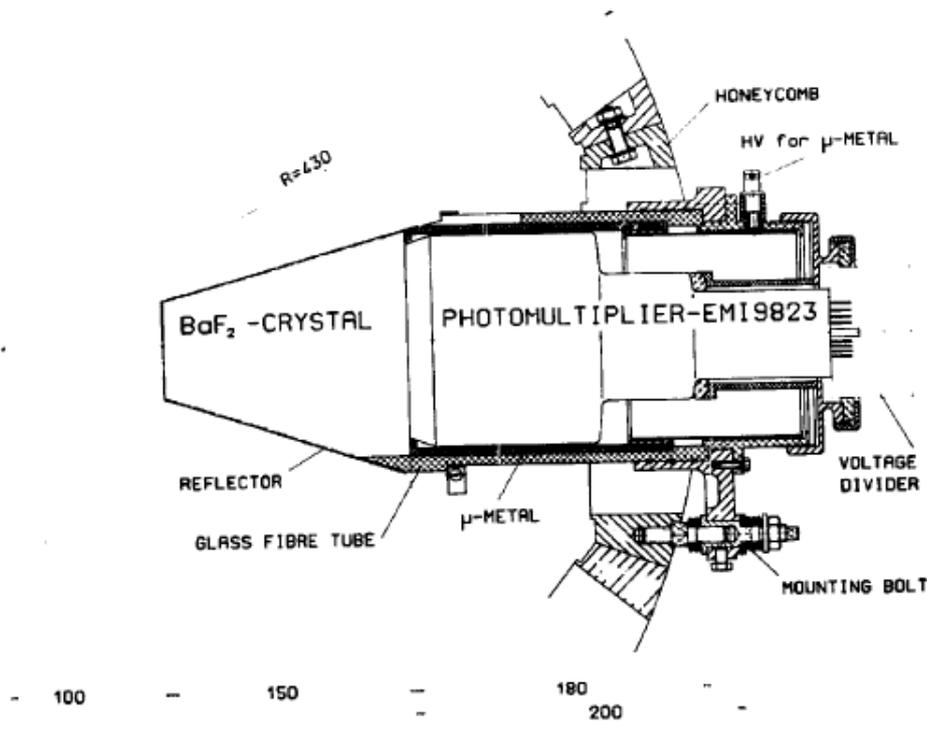


Abbildung 2.1.: Aufbau eines Detektors: Jeder Detektor besteht aus einem BaF<sub>2</sub>-Kristall. An der äußeren Seite des Kristalls ist ein Photomultiplier (PMT) angebracht. Zwischen dem Kristall und dem PMT wird eine dünne Silikonölschicht aufgetragen, um die optische Kopplung zu verbessern. Mit Ausnahme der an dem PMT angelegten Seite wird der Kristall komplett mit schwarzem Klebeband umwickelt, damit das Szintillationslicht nicht im PMT eines anderen Detektors nachgewiesen wird. Der PMT selbst ist von drei 0.1 mm dicken Metalllagen umgeben, um die Auswirkungen von äußeren elektrischen Feldern zu minimieren. Die Detektoren sind von einem Fiberglasrohr umhüllt und in einer kugelförmigen Wabenstruktur befestigt [5].

Im realen Aufbau des Detektors fehlen zwei Teile: der Eingangskristall, um die Wechselwirkung des Szintillatormaterials mit dem Strahl zu vermeiden.

Bestimmte Kristalle sind so geschnitten, dass in der Kugel ein Schlitz bleibt, der die Kugel in zwei große Teile trennt. In der Abbildung 2.3 lässt sich die Form dieses Schlitzes sehen, während in der Abbildung 2.4 die in zwei Teile getrennte Kugel zu betrachten ist. Für den Aufbau der Kugel werden beide Kugelteile separat aufgebaut und danach werden sie zusammengefügt bis der Schlitz einen Zwischenraum von 1 cm bildet. In der Realität sind die beiden Teile einander nicht vollständig bis zum Punkt der perfekten Kugel angenähert, wenn der Zwischenraum 1 cm beträgt, sondern sie sind noch durch einen eher vernachlässigbaren Abstand voneinander getrennt. Auf Grund dessen wird dieser Abstand in der Simulation nicht berücksichtigt und man geht von einer perfekten Kugel aus.

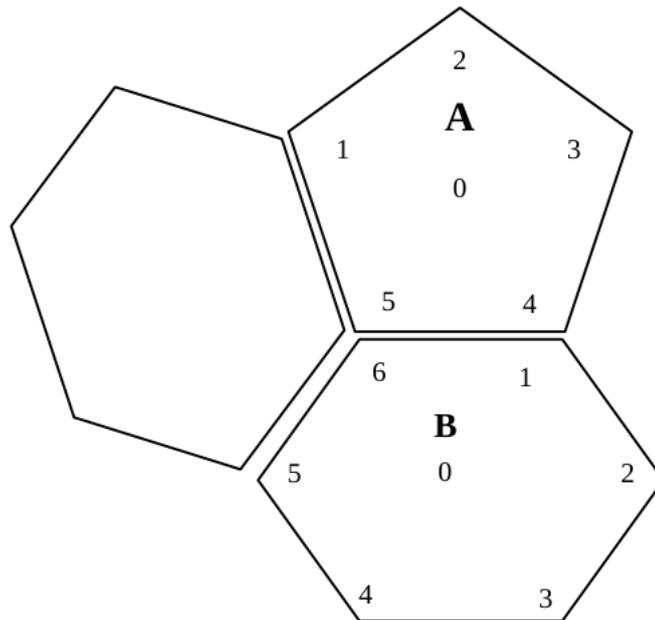


Abbildung 2.2.: Die Grundfläche der zwei Kristallarten, A und B, mit den gekennzeichneten Segmenten, deren Maße in der Tabelle 2.1 zu finden sind [3].

Segment	Kristall A	Kristall B
1-2	0.4329	0.3178
2-3	0.4329	0.3178
3-4	0.4329	0.4298
4-5	0.4329	0.3178
5-6	0.4329	0.3178
6-1	-	0.4298
1-0	0.3682	0.3453
2-0	0.3682	0.3820
3-0	0.3682	0.3453
4-0	0.3682	0.3453
5-0	0.3682	0.3820
6-0	-	0.3453

Tabelle 2.1.: Dimensionen der in der Abbildung 2.2 gezeigten Segmente mit dem Abstand der Oberflächen zur Mitte der Kugel als Einheit. Die Proportion muss erfüllt werden, damit die 42 Flächen den gesamten Raumwinkel perfekt abdecken [3].

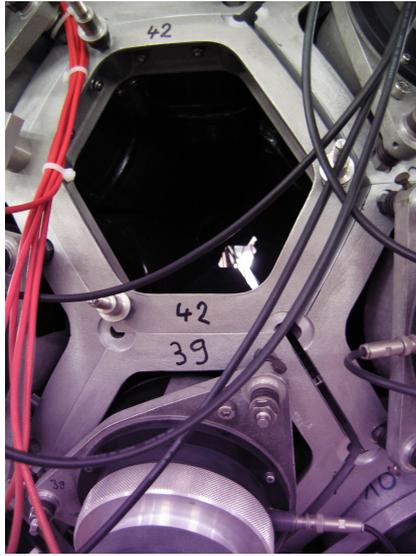


Abbildung 2.3.: Ansicht des Kalorimeters bei der Eingangsstelle. Man erkennt in der Wabenstruktur die Form des Schlitzes rechts des Eingangskristalls.

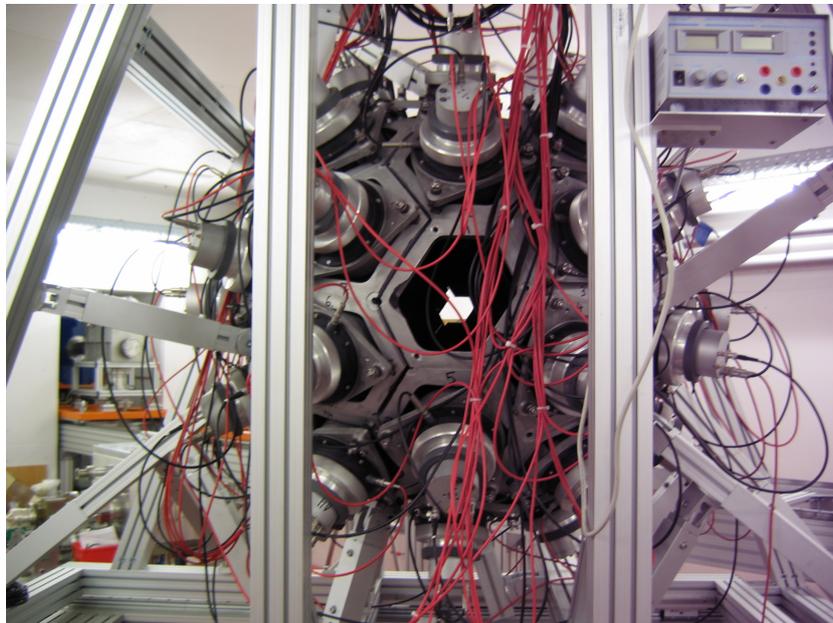


Abbildung 2.4.: Ansicht des Kalorimeters von der Ausgangsstelle betrachtet. Man erkennt im Bild die zwei fehlende Kristalle B, den Eingangs- und den Ausgangskristall. Links vom Ausgangskristall erkennt man den Schlitz, der die Kugel in zwei große Teile trennt.

## 2.3. Funktionsweise des Detektors

### 2.3.1. Szintillation

Als Szintillator gilt ein Material, dessen Moleküle bei der Wechselwirkung mit energiereichen Photonen oder geladenen Teilchen durch Stoßprozesse angeregt werden und die Anregungsenergie wieder als Photonen abgeben. Der Vorgang wird als Szintillation bezeichnet [4].

Bei der Szintillation steht das einfallende Photon mit dem Szintillator vorwiegend über den Compton-Effekt in Wechselwirkung, indem ein Teil der Energie an ein Elektron des Szintillators abgegeben wird. Wenn diese übertragene Energie größer ist als die Bindungsenergie, wird das Elektron herausgelöst. Dieses Elektron wiederum wechselwirkt, wie im Abschnitt 2.1.1 beschrieben mit dem Szintillator, wodurch aus dem ursprünglichen energiereichen Photon viele niederenergetische Photonen freigesetzt werden. Das Vorgehen wiederholt sich bis die übertragene Energie die Bindungsenergie des Szintillators nicht mehr überwinden kann. In dieser letzten Reaktion wird die restliche Energie an ein Elektron abgegeben, welches diese über Stöße wiederum in Photonen umwandelt. Die letztendlich entstehenden niederenergetischen Photonen, deren Energieverteilung von der Wahl des Szintillatormaterials abhängig ist, bezeichnet man als Szintillationslicht. Sie haben so viel Energie verloren, dass der für den Nachweis im Photomultiplier (PMT) notwendige photoelektrische Effekt wahrscheinlicher wird [2].

Bariumfluorid ( $\text{BaF}_2$ ) wirkt im Kalorimeter als Szintillator und seine Aufgabe ist die Umwandlung hochenergetischer Photonen in viele niederenergetische, damit sie vom PMT wahrscheinlicher nachgewiesen werden können [2].

$\text{BaF}_2$  bietet die folgenden Vorteile als Szintillationsmaterial [2]:

- Es weist einen geringen Neutroneneinfangquerschnitt im Energiebereich von einigen keV auf. Diese Eigenschaft ist für die mit FRANZ geplanten Experimente entscheidend, da an der Probe zum großen Teil eine Neutronenstreuung stattfinden wird. Ein hoher Neutroneneinfangquerschnitt des Szintillators hätte einen hohen Messhintergrund zur Folge.
- Es bietet eine akzeptable Energie- sowie eine sehr gute Zeitauflösung. Andere Materialien wie Germanium (Ge) bieten eine bessere Energieauflösung, aber keine gute Zeitauflösung, was für den Nachweis einzelner Ereignisse entscheidend ist.

Ein Nachteil dieses Materials ist die Verunreinigung des Bariums mit  $^{226}\text{Radium}$ , welches ein  $\alpha$ -Strahler ist und wodurch ein  $\alpha$ -Hintergrund erzeugt wird. Die Wechselwirkung von  $\alpha$ -Teilchen mit Materie im Vergleich zu der von Photonen ist aber leicht unterschiedlich und lässt sich somit herausfiltern. In der Zerfallsreihe von  $^{226}\text{Radium}$  gibt es allerdings auch  $\beta$ -Strahler, wodurch trotzdem ein Hintergrund erhalten wird [2].

### 2.3.2. Photomultiplier

Ein Photomultiplier (engl. photomultiplier tube, PMT) ist eine spezielle Elektronenröhre mit dem Zweck, einkommende schwache Lichtsignale (bis hin zu einzelnen Photonen) zu detektieren und durch Erzeugung und Verstärkung zu messen. In der Abbildung 2.5 sieht man ein Schema eines

PMT [4].

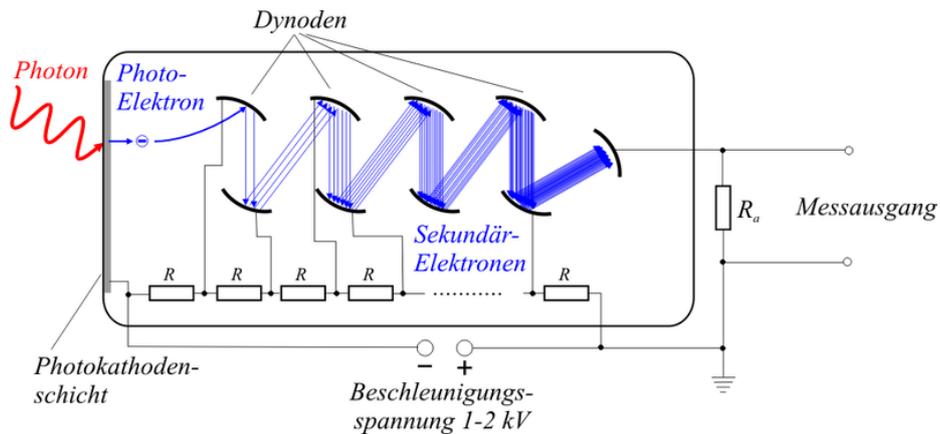


Abbildung 2.5.: Schema eines Photomultipliers [4].

An der Seite, links in der Abbildung, wird eine Photokathodenschicht aufgetragen. Die Photonen treffen auf diese Schicht und lösen durch den äußeren photoelektrischen Effekt Elektronen aus ihrer Oberfläche heraus. Diese freigesetzten Elektronen werden in einem elektrischen Feld beschleunigt und treffen auf sogenannte Dynoden (eine Serie von Einzelelektroden), die auf zunehmend positivem Potential liegen, von links nach rechts in der Abbildung. Aus der Oberfläche der Dynoden wird dann jedes auftreffende Elektron mehrere Sekundärelektronen herausschlagen und die Anzahl der Elektronen von Dynode zu Dynode wird exponentiell zunehmen. Am Ende der Röhre treffen die Elektronen auf eine Anode und erzeugen einen Spannungsabfall über einen Widerstand [4].

Der entsprechende Strom aus dem Ausgangssignal ist proportional zur Anzahl der auftreffenden Photonen und wird an einen ADC (Analog to Digital Converter) weitergegeben. Zur Datenaufnahme wird ein CAEN v1751 Digitizer verwendet, der aus einem FADC (Flash-Analog-to-Digital-Converter) und einem FPGA (Field Programmable Gate Array) besteht [2].

## 3. Software Werkzeuge

### 3.1. Geant4

Die Software wurde von RD44, einer weltweiten Kollaboration von etwa 100 Wissenschaftlern, im CERN entwickelt. Sie dient zur Entwicklung von Simulationen über die Wechselwirkung von Teilchenstrahlen mit Materie. Das findet Anwendung in der Hochenergie-, Nuklear- und Beschleunigerphysik, sowie in der Medizin und Astrophysik. Geant4 ist ein geeignetes Werkzeug für die genannten Experimente, da es eine hohe Anzahl von Materialien, Physikeinheiten, Physikkonstanten, Daten über Teilchen übereinstimmend mit der „Particle Data Group“ (<http://pdg.lbl.gov>), ein breites Energieintervall (von etwa 250 eV bis in die Ordnung von TeV je nach Experiment) und viele wichtige physikalische Prozesse für hochenergetische Experimente mit verschiedenen Implementationen je nach Situation, u.a. die meisten möglichen nuklearen Wechselwirkungen, einschließt [6].

Diese vierte Version des Programms ist in der Programmiersprache C++ geschrieben, was ein großer Schritt im Vergleich mit der letzten Version ist, die in FORTRAN geschrieben wurde. Das Programm ist objektorientiert programmiert, daher modular, bearbeitbar und erweiterbar.

In der Arbeit [3] wurde gezeigt, dass sich diese Software für die Simulation von einem  $4\pi$  BaF<sub>2</sub> Detektor, wie der von FRANZ, eignet. Die Wechselwirkung mit der Umgebung wird gezählt, wenn ein Energieübertrag von mindestens 10 keV erreicht wird. Zu diesem Zeitpunkt verschwinden die aus der Quelle des Detektors geschossenen Teilchen und die Energie wird im Material deponiert. Die Szintillation ist von der experimentell bestimmten Energieauflösung abhängig und wird nicht simuliert. Die erhaltenen Ergebnisse stimmen sehr genau mit durchgeführten Messungen überein [3].

### 3.2. HepRep

Geant4 nutzt viele Visualisationssysteme, die je nach Anforderung der Experimente eingesetzt werden können. Die am meisten benutzte Option in dieser Simulation ist HepRep. Geant4 schreibt nach Wunsch eine Datei mit dem Suffix .heprep. Diese Datei ist im standard hierarchischen Format XML geschrieben, das nicht besonders für Geant4 entwickelt wurde, sondern mit der Absicht, als generisches Interface zwischen Daten und einer Visualisationssoftware zu dienen. Da HepRep nicht besonders für Geant4 entwickelt wurde, enthält es nicht die ganze Hierarchie von Volumina, die Geant4 bietet. Daher zerlegt Geant4 die komplexen Gestalten in einfachere, bevor sie in die HepRep-Datei geschrieben werden [7]. Das ist eine Quelle möglicher Fehler und ein Nachteil dieses Systems.

Einige Visualisationssysteme werden durch Eingaben im Geant4-Interface gesteuert und andere, darunter heprep, durch andere Anwendungen. Das von Geant4 benutzte Werkzeug, um diese Datei zu behandeln, heißt HepRApp. HepRApp ist eine Java-Anwendung und für ihre Nutzung benötigt man zusätzlich eine JAR-Datei. Die benutzte Java-Version für diese Simulation ist 6.0.32. Die Vorteile von diesem System sind die zahlreichen Möglichkeiten, die den Entwickler die Sichtbarkeit kontrollieren lassen, um einen komplexen Aufbau zu erleichtern: Zoom, Rotation, Translation, sowie Kontrolle über einzelne Teile der Simulation, z.B. an- und ausschalten, Eigenschaften ansehen, per Eigenschaften auswählen, etc [8].

### **3.3. OpenGL**

Dieses Visualisationssystem kann man mit direkten Eingaben im Geant4-Interface ohne weitere Anwendungen nutzen, wenn sie während der Installation von Geant4 als extra Option verlangt wird. Dieses System eignet sich für schnelle Echtzeitvisualisationen, was durch Hardwarebeschleunigung geschafft wird. Die Schnelligkeit erlaubt u.a. Animationen zu erzeugen [9].

Im Vergleich mit HepRep kann man ein erzeugtes Bild nicht kontrollieren, sondern die Bilder bleiben wie ein Foto vom angegebenen Sichtwinkel. Man kann aber bessere Bilder von den Detektoren mit diesem Werkzeug erhalten. Für diese Arbeit wird OpenGL dafür benutzt, Bilder mit eingefärbten Flächen und aus verschiedenen Winkeln zu erhalten, um den Aufbau des Kalorimeters zu veranschaulichen.

### **3.4. ROOT**

ROOT ist ein vom CERN koordiniertes Open Source Projekt für wissenschaftliche Datenanalyse, das auch mit der Programmiersprache C++ entwickelt wurde, womit es sich für die Nutzung zusammen mit Geant4 eignet. ROOT bietet sowohl einen Interpreter als auch ein graphisches Interface an [10].

Für die Art von Simulationen dieser Arbeit eignet sich diese Software besonders aufgrund der vorprogrammierten Klassen und Methoden, die eine große Menge von zeitlich zusammenhängenden Daten beliebig zu analysieren erlaubt. Histogramme mit Energiespektren in allen Detektoren sowie im gesamten Kalorimeter werden erzeugt. Zudem werden in einer Datei nach jeder Simulation mit der Klasse TTree alle Ereignisse für spätere Untersuchungen registriert, mit dem Ziel nur spezifische Daten zu filtern.

## 4. Implementierung des Kalorimeters

Nach dem im Abschnitt 2.2 gezeigten realen Aufbau des Kalorimeters wird in diesem Kapitel die Implementation mit der Software Geant4 beschrieben.

Zwei Teile von dieser Beschreibung gehören nicht zur Simulation: Das schwarze Klebeband hat eine zu schwache Wechselwirkung mit der Quelle. Dieser Faktor macht die Simulation aufwendiger ohne lohnswerte Resultate zu bringen und wird daher als vernachlässigbar betrachtet. Andererseits ist die Wechselwirkung mit der Wabenstruktur nicht ganz vernachlässigbar, weil das Produkt dieser Wechselwirkung zurück in den Photomultiplier gelangen kann. Das macht aber die Simulation ebenfalls viel aufwändiger und der Einfluss von diesem Material sowie von den elektronischen Anteilen des Aufbaus außerhalb des Kalorimeters in den ausgelesenen Ergebnisse gehören nicht zu dieser Arbeit.

### 4.1. Materialien

Es gibt in Geant4 verschiedene Weisen ein Material nach seinen Eigenschaften und Bestandteilen zu definieren. Die Elemente selbst werden nach ihrer Ordnungszahl, molaren Masse und andere optionale Eigenschaften definiert. Die Daten werden mit einer Genauigkeit von bis zu zwei Dezimalstellen mit Rundung angegeben.

Die Luft wird vereinfacht als eine Verbindung von 70% Stickstoff (N) und 30% Sauerstoff (O) definiert, was üblich in derartigen Simulationen ist. Bariumfluorid wird als ein Molekül aus einem Atom Barium (Ba) und zwei Atomen Fluor (F) definiert und Teflon als ein Molekül aus zwei Atomen Kohlenstoff (C) und vier Atomen Fluor (F). Auch wenn Aluminium ein Element ist, wird es wie ein Material definiert, da es in der Simulation für die entsprechende Folie direkt anzuwenden ist. In der Tabelle 4.1 werden die angewendeten Daten gezeigt.

### 4.2. Geometrie der einzelnen Detektoren

Die Weise, auf die man mit Geant4 die verschiedenen Volumina darstellt, ist folgende: Zuerst wird ein Festkörper je nach den reinen geometrischen Bedingungen definiert, danach ein sogenannter logischer Körper, der dem Festkörper ein Material zuordnet und am Ende wird ein physischer Körper definiert, der letztendlich dem logischen Körper eine Position im Raum zuordnet. In diesem Abschnitt wird die ausgewählte Strategie für den Entwurf des Festkörpers von jedem einzelnen Detektortyp beschrieben.

Für die Erzeugung der gewünschten Volumina werden oft boolesche Operationen verwendet.

Element	Ordnungszahl	Molare Masse [g/mol]	Dichte [g/cm <sup>3</sup> ]
Stickstoff (N)	7	14.01	
Sauerstoff (O)	8	16	
Barium (Ba)	56	137.33	
Fluor (F)	9	19	
Kohlenstoff (C)	6	12.01	
Material			
Luft			0.0012
BariumFluorid (BaF <sub>2</sub> )			4.89
Teflon (C <sub>2</sub> F <sub>4</sub> )			2.16
Aluminium (Al)	13	26.98	2.7

Tabelle 4.1.: Daten der für die in der Simulation angewendeten Materialien. Aluminium wird wie ein einfaches Material definiert [4].

Geant4 erlaubt Union, Schnittstelle und Subtraktion als boolesche Operationen. Man muss hier besonders aufpassen: Bei der Subtraktion in Geant4 dürfen die Punkte und Flächen zweier Volumina nicht übereinstimmen. Es ist sinnvoll, wenn man bedenkt, dass nach einer Subtraktion von einem Randteil eines Volumens die Flächen dieses Teils nicht mehr zum Volumen gehören würden. Aufgrund dessen werden sie nicht mitsubtrahiert und wenn die Punkte oder Flächen zweier Volumina einer Subtraktion übereinstimmen, sind sie nach der Subtraktion noch vorhanden.

Diese Eigenschaft verursacht allerdings einige Schwierigkeiten, wie man später sehen wird. Zunächst muss man die Subtraktion eines Volumenrandteils so implementieren, dass der zu subtrahierende Teil in allen Dimensionen größer ist als der Teil, von dem subtrahiert wird. Nicht selten werden Volumina definiert, die wiederum für eine neue Subtraktion nützlich sind und daher sollte das erste Volumen zweimal vergrößert werden.

Neben der Beschreibung der Implementation werden zur Veranschaulichung auch Bilder gezeigt, die mit dem Visualisationssystem OpenGL erzeugt wurden. Alle angegebenen Größen können im Anhang B nachgeschlagen werden.

### 4.2.1. Zylinder

Da die Kristalle im äußeren Bereich einen Zylinderschnitt besitzen, muss man getrennt einen Zylinder definieren, mit dem danach die Schnittstelle mit den Kristallen berechnet wird. Geant4 bietet für so ein grundlegendes Volumen die Klasse G4Tubs, bei der die Höhe und der Radius des Zylinders anzugeben sind.

Der Radius des Zylinder beträgt ursprünglich 7 cm. Dieser Radius wird jedoch an den äußeren Radius der Kugel angepasst, um die Dimensionierung des ganzen Kalorimeters (siehe Kapitel 6) zu erlauben. Der Wert des Radius beträgt dann  $7/25$  des äußeren Radius der Kugel.

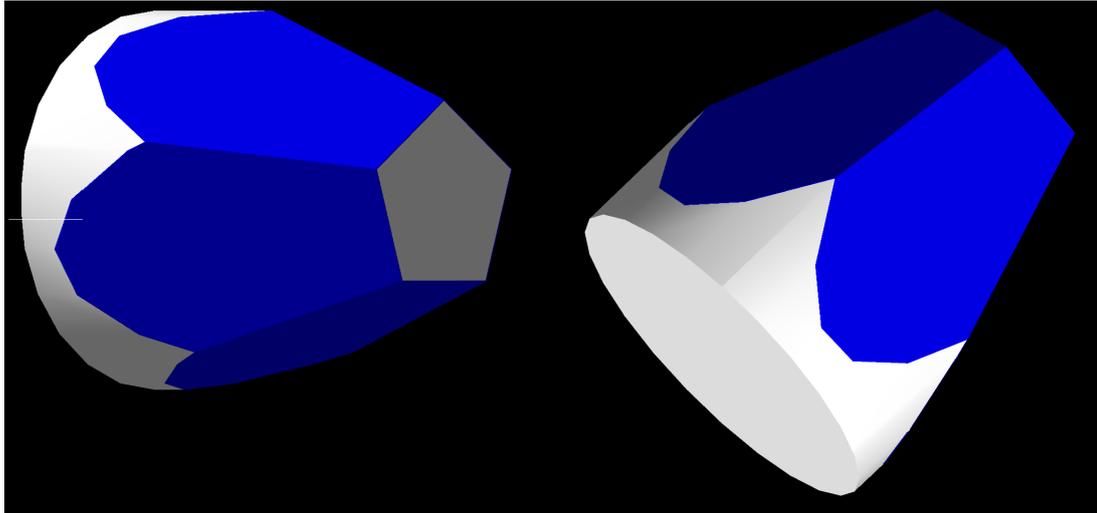


Abbildung 4.1.: Ansicht des Kristalls A. Links die Pentagonoberfläche im inneren Bereich der Kugel und rechts eine Ansicht des zylindrischen Schnittes. In Blau sind die Teflon- und Aluminiumfolien markiert.

#### 4.2.2. Verkürzter Zylinder

Für die abgeschnittenen Kristalle am Rand des Schlitzes benötigt man auch einen kleineren Zylinder. Der Radius des verkürzten Zylinders wird je nach Fall so abgeschätzt: Radius des Zylinders minus  $\text{pentagonCircumscribedCut}/2$  falls der Kristall A (mit einem Pentagon als Grundfläche) geschnitten wird und minus  $\text{trapezoidHeightCut}/2$  falls der Kristall B (mit einem Hexagon als Grundfläche) geschnitten wird. Hierbei sind die Größen  $\text{pentagonCircumscribedCut}$  und  $\text{trapezoidHeightCut}$  in der Abbildung A.4 dargestellt. Der Zylinder wird somit verkürzt und um dieselbe Größe entsprechend verschoben.

Für die Verkürzung des Radius im Fall des Pentagons wird diese Größe genommen, da das Pentagon an den Seiten 1 und 5 im Uhrzeigersinn geschnitten wird, wobei der Zylinder später an der Achse des  $\text{CircumscribedRadius}$  verschoben werden muss.

Das Hexagon kann sowohl am Dach des Trapezoids als auch an der Seite geschnitten werden. Für die Verkürzung des Radius in diesem Fall wird einheitlich die Höhe des Trapezoids genommen. Es kann nicht zwei verschiedene Zylinder für beide Fälle geben, da in manchen Fällen beide Seiten gleichzeitig am Schlitz liegen. Experimentell wurde zudem bestätigt, dass die Zylinder am Rand des Schlitzes etwa dieselbe Größe hatten.

#### 4.2.3. Kristall A

Die erste Kristallart ist ein Pyramidenstumpf mit einem Pentagon als Grundfläche. Die Höhe des Volumens entspricht mit ursprünglich 15 cm der Dicke der Detektorkugel. In der Abbildung 4.1 kann man eine Darstellung des Volumens sehen.

Geant4 bietet die Klasse `G4Polyhedra`, bei der die Höhe des Volumens sowie die Inkreisradien beider Pentagone anzugeben ist, um das Volumen zu erhalten. Danach kommt man mittels einer Schnittstelle zwischen dem Pyramidenstumpf und dem Zylinder auf die endgültige Form des

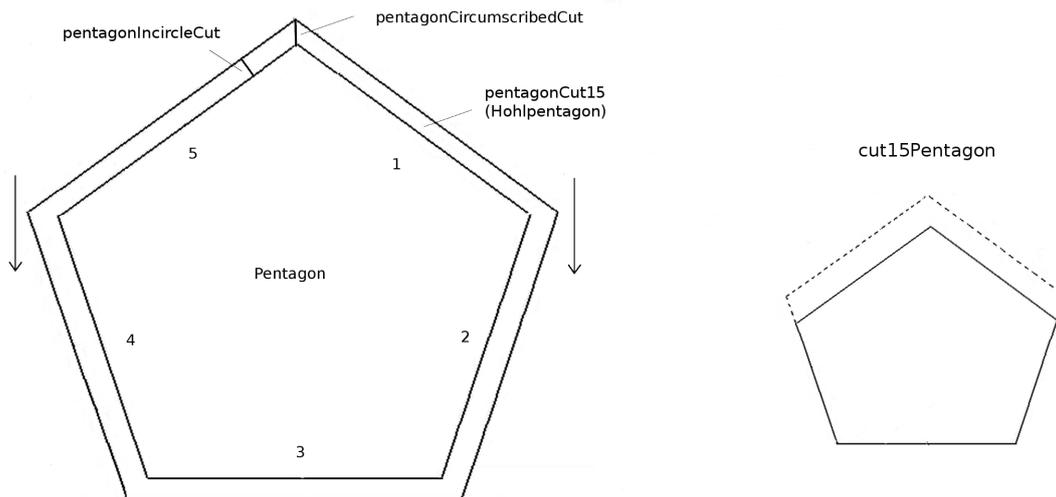


Abbildung 4.2.: Das Hohlpentagon hat einen inneren Inkreisradius so groß wie der vom Pentagon und einen äußeren so groß wie der innere plus die gewünschte Länge des Stückes, das vom Pentagon abgeschnitten werden soll. Dieses Hohlpentagon wird zuerst nach unten verschoben und dann vom Pentagon subtrahiert. cut15Pentagon ist das resultierende in den Seiten 1 und 5 geschnittene Pentagon.

Kristalls aus  $\text{BaF}_2$ .

Teflon-Folie: Zuerst wird ein größerer Pyramidenstumpf geschaffen. Die angewendete Klasse verlangt dafür nur eine zusätzliche Länge für den Inkreisradius, was der Gleichung (B.13) zu entnehmen ist. Von diesem großen Pyramidenstumpf wird der kleinere subtrahiert und die verbliebene Schicht überschneidet sich mit demselben Zylinder.

Aluminium-Folie: Zuerst wird ein noch etwas größerer Pyramidenstumpf geschaffen. Die benötigte zusätzliche Länge für den Inkreisradius erhält man aus der Gleichung (B.14). Von diesem Pyramidenstumpf wird der nächst kleinere Pyramidenstumpf subtrahiert, der für den Aufbau der Teflon-Folie geschaffen wird. Auf diese Weise verbleibt nur die gewünschte Aluminium-Folie. Am Ende überschneidet sich diese Folie auch mit demselben Zylinder.

#### 4.2.4. Kristall A mit Schnitt in den Seiten 1 und 5 im Uhrzeigersinn

Die geschnittenen Kristalle befinden sich am Rand des Schlitzes in der aufgebauten Kugel. In diesem Fall wird aus dem Pyramidenstumpf ein Hohlpyramidenstumpf, wie in der Abbildung 4.2 gezeigt, zuerst nach unten verschoben und danach subtrahiert, um die Seiten 1 und 5 im Uhrzeigersinn wie gewünscht zu schneiden. Die Größen `pentagonIncircleCut` und `pentagonCircumscribedCut` aus der Abbildung 4.2 berechnen sich mit den Gleichungen (B.46) bzw. (B.47).

Teflon-Folie: Man geht zunächst ähnlich wie beim Kristall A vor, aber statt vom großen Pyramidenstumpf der kleinere zu subtrahieren, wird der geschnittene Pyramidenstumpf subtrahiert. Danach wird ein größerer Hohlpyramidenstumpf, wie in der Abbildung 4.3 gezeigt, zuerst nach unten verschoben und danach subtrahiert. Die verbliebene Schicht überschneidet sich letztendlich

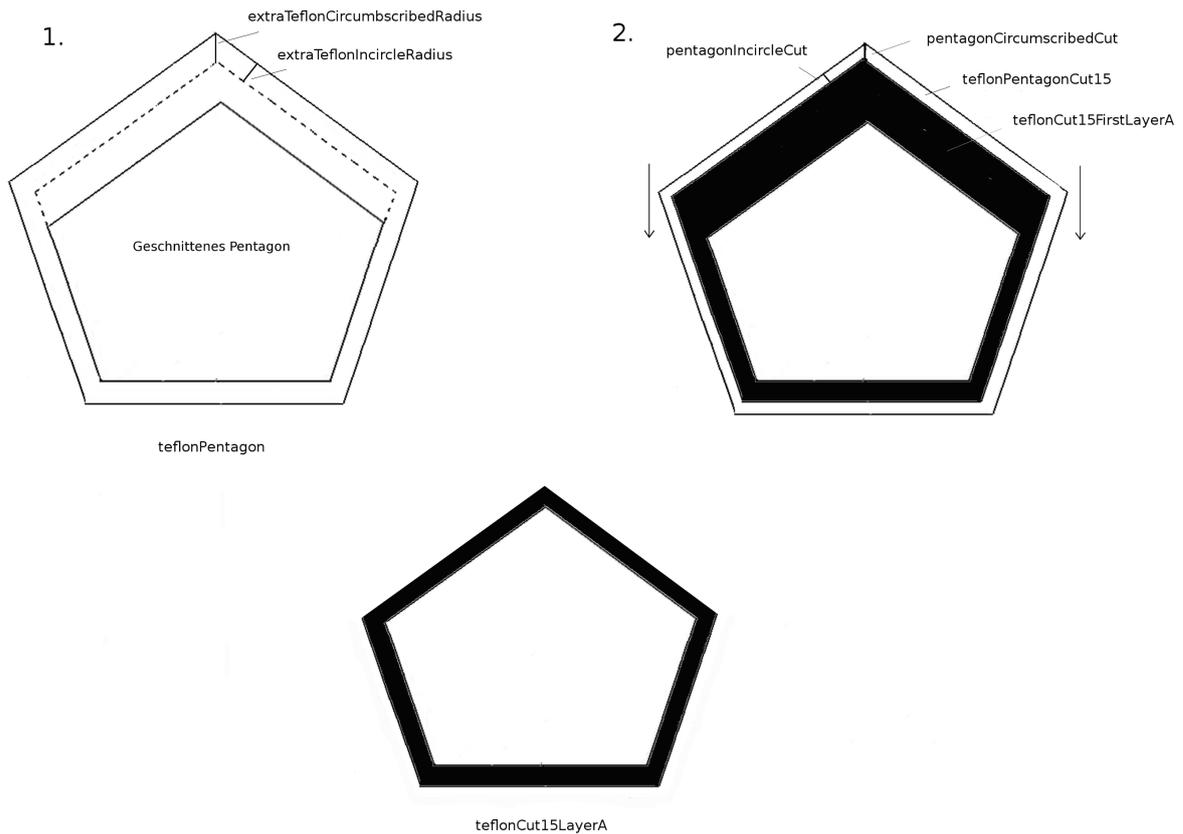


Abbildung 4.3.: Im ersten Schritt: Eine erste Folie wird erzeugt, indem vom größeren Pentagon aus Teflon das abgeschnittene Pentagon subtrahiert wird. Im zweiten Schritt: Ein größeres Hohlpentagon (treflonPentagonCut15 im Bild), wobei die Vergrößerung des Inkreisradius der Länge des Stückes entspricht, das vom Pentagon abgeschnitten wird, wird zuerst nach unten verschoben und vom Produkt der Subtraktion im ersten Schritt subtrahiert.

mit dem gekürzten Zylinder. Die Größen `extraTeflonIncircleRadius` und `extraTeflonCircumscribedRadius` Gleichung aus der Abbildung 4.3 berechnen sich mit den Gleichungen (B.13) bzw. (B.18)

Aluminium-Folie: Man geht wieder ähnlich wie beim Kristall A vor, aber aus dem noch größeren Pyramidenstumpf wird statt des nächst kleineren Pyramidenstumpfs der nächst kleinere geschnittene Pyramidenstumpf subtrahiert und danach wird das Hohlpyramidenstumpf, wie in der Abbildung 4.3 gezeigt, subtrahiert. Die verbliebene Schicht überschneidet sich letztendlich mit dem gekürzten Zylinder.

#### 4.2.5. Kristall B

Die zweite Kristallart ist ein Pyramidenstumpf mit einem Hexagon als Grundfläche. Die Höhe des Volumens entspricht mit ursprünglich 15 cm der Dicke der Detektorkugel. Auf dem Bild 4.4, kann man eine Darstellung des Volumens sehen.

Im Prinzip wurde als einfachster Fall versucht, zwei Trapezoide zu verbinden, aber aufgrund

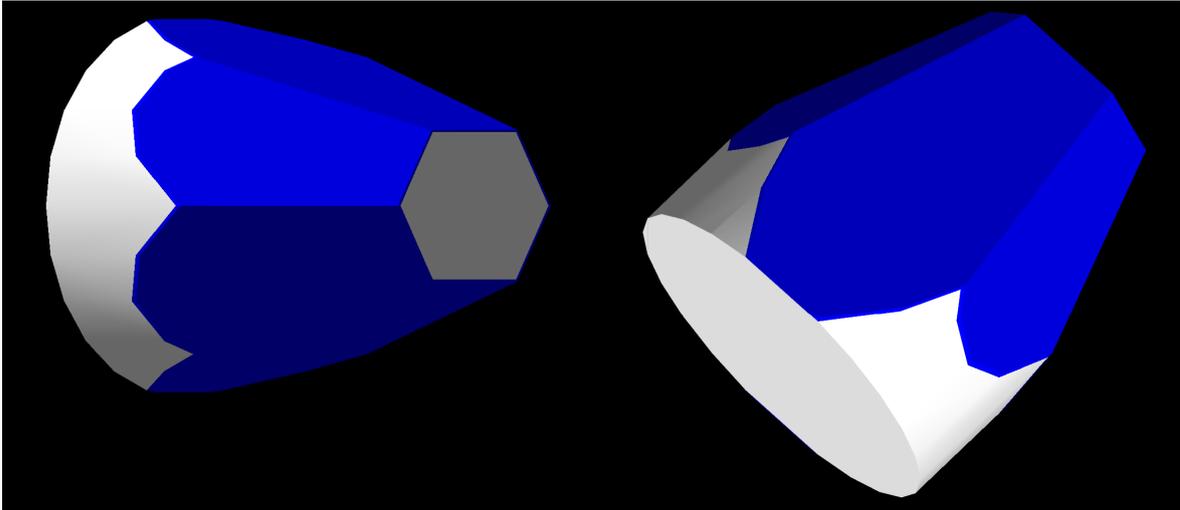


Abbildung 4.4.: Ansicht des Kristalls B. Links die Hexagonoberfläche im inneren Bereich der Kugel und rechts eine Ansicht des zylindrischen Schnittes. In Blau sind die Teflon- und Aluminiumfolien markiert.

der Eigenschaft von Geant4, die Randflächen nicht zu subtrahieren, wurde diese Option später nicht geeignet. In Geant4 wird die Subtraktion einer Union so durchgeführt, dass zuerst das eine Volumen subtrahiert wird und danach das andere. Auf diese Weise bleibt bei der Konstruktion der Teflon- bzw. Aluminium-Folien eine Fläche in der Mitte des Volumens, wenn man versucht, das kleinere Hexagon vom größeren zu subtrahieren. Daher geht die Entscheidung für die Konstruktion des Hexagons über die Klasse `G4ExtrudedSolid`, die am Anfang komplexer aussieht, aber einen einfacheren Weg für alle das Hexagon betreffenden Volumina bietet. Zuletzt kommt man mittels einer Schnittstelle zwischen dem Pyramidenstumpf und dem Zylinder auf die endgültige Form des Kristalls.

Die Teflon- bzw. Aluminium-Folie wird wie bei Kristall A analog zu Abschnitt 4.2.3 durch einen Pyramidenstumpf mit einem Hexagon statt eines Pentagons als Grundfläche gemacht.

#### 4.2.6. Kristall B mit Schnitten

Um die Schnitte von den Hexagonen zu implementieren, definiert man zuerst Trapezoide einzeln mit der Form der fehlenden Seite mit der Klasse `G4GenericTrap`, wie in der Abbildung 4.5 dargestellt. Dann werden sie vom Pyramidenstumpf nacheinander subtrahiert. Es gibt drei Fälle von Schnitten und bei allen drei geht man mit den entsprechenden subtrahierenden Trapezoiden gleich vor:

- Schnitt in den Seiten 2 und 3 des Hexagons im Uhrzeigersinn
- Schnitt in den Seiten 3 und 4 des Hexagons im Uhrzeigersinn
- Schnitt in den Seiten 1 und 2 des Hexagons im Uhrzeigersinn

Die Größen `trapezoidHeightCut`, `trapezoidSideWidthCut` und `trapezoidBaseCut` aus der Abbildung 4.5 berechnen sich mit den Gleichungen (B.48), (B.49) bzw. (B.50).

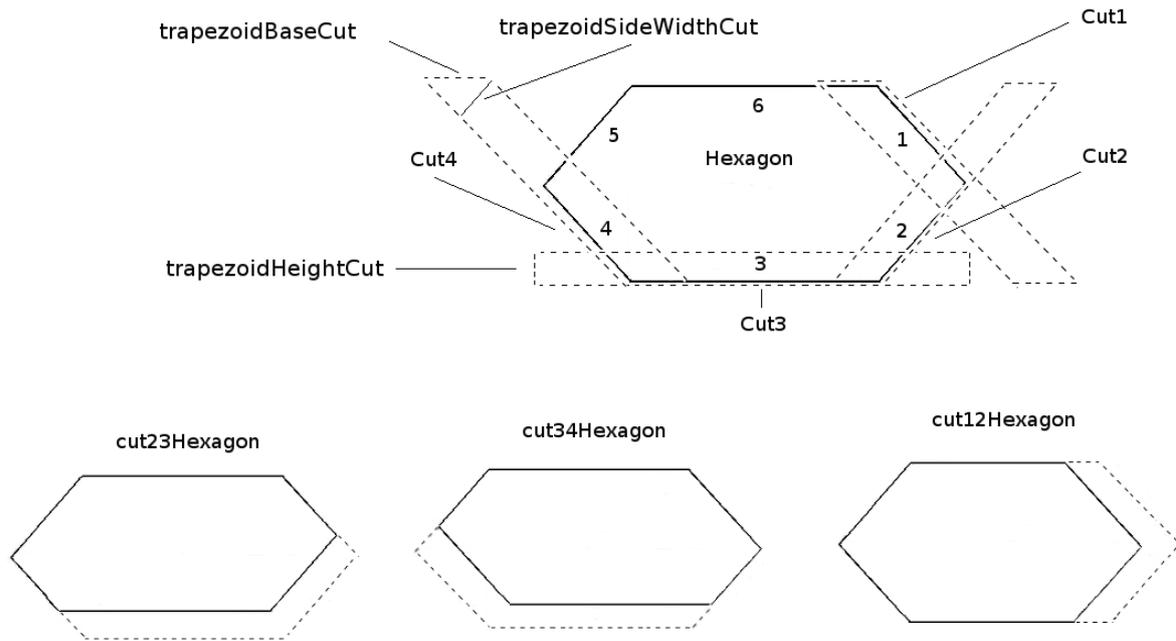


Abbildung 4.5.: Die Trapezoiden mit den gestrichelten Linien verfügen über eine ausreichende Größe, um die entsprechenden Stücke abzuschneiden. Die drei geschnittenen Hexagone erhält man, indem die oben gezeigten Trapezoide nacheinander vom Hexagon subtrahiert werden.

Die Teflon- bzw. Aluminium-Folie wird nach denselben Algorithmen geschaffen wie beim geschnittenen Kristall A im Abschnitt 4.2.4, aber durch einen Pyramidenstumpf mit einem Hexagon statt eines Pentagons als Grundfläche und der oben genannten Methode, um die Volumina zu schneiden.

### 4.3. Zusammensetzung des kompletten Kalorimeters

#### 4.3.1. Strategie

Die Strategie, um eine Kugel mit allen Teilen zu schaffen, besteht darin, ein Muster zu erkennen, zu bauen und wiederholt an den gewünschten Stellen zu platzieren. In der Abbildung 4.6 wird die grundlegende Anordnung der Muster dargestellt, die sich in dieser Geometrie wiederholt. Die Aufzählung der Einzelteile hilft, um den Aufbau nachzuvollziehen.

Die Zwischenräume sind nicht alle gleich: Zwischen Kristall A und den Kristallen vom Typ B gibt es den im Kapitel 2.2 genannten Zwischenraum  $a$ , aber zwischen den Kristallen vom Typ B gibt es einen etwas größeren Zwischenraum, nämlich  $2 \cdot a \cdot \sin(36^\circ)$  und  $\sin(36^\circ)$  ist leicht größer als  $1/2$ . Jedoch stört das den Aufbau nicht, da der Zwischenraum zwischen Kristall A und den Kristallen vom Typ B überall in der Kugel wie in diesem Muster bleibt.

In der Abbildung 4.7 sieht man die zwei Hälften der kompletten Kugel mit den aufgezählten Detektoren. Zum Aufbau fängt man an der Grenze der positiven Z-Achse an, in Richtung des

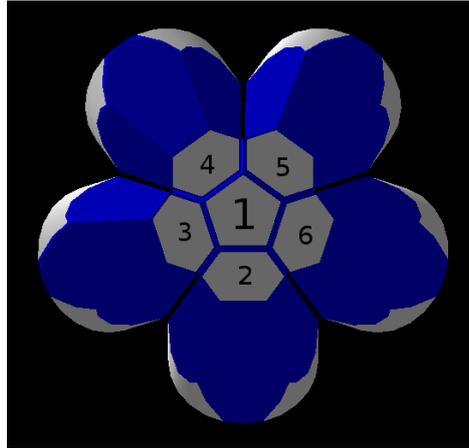


Abbildung 4.6.: In der Abbildung sieht man ein an der senkrechten Achse um  $180^\circ$  gedrehtes Muster beim Aufbau des Kalorimeters. Diese Drehung dient dazu, die Grundflächen zu betrachten. Kristall A, der eine Pentagongrundfläche aufweist, liegt in der Mitte an Stelle 1 und die fünf Kristalle vom Typ B liegen, wie im Bild gezeigt, um den Hauptkristall. Für die Aufzählung der Positionen fängt man in der  $-90^\circ$ -Stelle an und geht gegen den Uhrzeigersinn vor (im Bild umgekehrt dargestellt).

Betrachters. Die folgenden vier Muster werden dabei angewendet:

- Das erste Muster ist ein Ring aus fünf Kristallen vom Typ B. Es wird nur einmal abgebildet.
- Das zweite Muster hat drei Kristalle vom Typ B. Der erste liegt in der  $-18^\circ$ -Stelle, was Position 3 in der Abbildung 4.6 entspricht. Die Positionen 3, 4 und 5 des Musters sind somit besetzt. Die Kristalle in den Positionen 4 und 5 entsprechen den geschnittenen Kristallen der ersten bzw. zweiten Art, die im Abschnitt 4.2.6 präsentiert wurden. Dieses Muster wird fünfmal so um das erste abgebildet, dass in Position 2 des Musters einer der Kristalle vom Typ B des ersten Musters liegt. Der Kristall in Position 4 der zweiten Abbildung und der in Position 5 der fünften Abbildung entfallen und stellen den Ausgangskristall bzw. den Eingangskristalls dar.
- Das dritte Muster hat zwei Kristalle vom Typ B. Der erste liegt in der  $-90^\circ$ -Stelle, was Position 2 entspricht. Es sind also die Positionen 2 und 3 besetzt. Der Kristall in Position 1 entspricht dem geschnittenen Kristall A aus dem Abschnitt 4.2.3, während der Kristall in Position 3 dem geschnittenen Kristall der dritten Art aus dem Abschnitt 4.2.6 entspricht. Dieses Muster wird wieder fünfmal, wie in der Abbildung 4.7 gezeigt, abgebildet.
- Das vierte Muster besteht aus nur einem Kristall des Typs A, die „Decke“ der Kugel, um  $180^\circ$  an der Z-Achse gedreht.

### 4.3.2. Implementierung in Geant4

Die Funktionsweise von Geant4 ist folgende: Es wird ein sogenanntes World-Volumen definiert, das die ganze Geometrie der Simulation enthält. Alle anderen physischen Volumina des gesamten Detektors sollen an den logischen World-Volumen hängen. Auf dieselbe Weise kann man andere

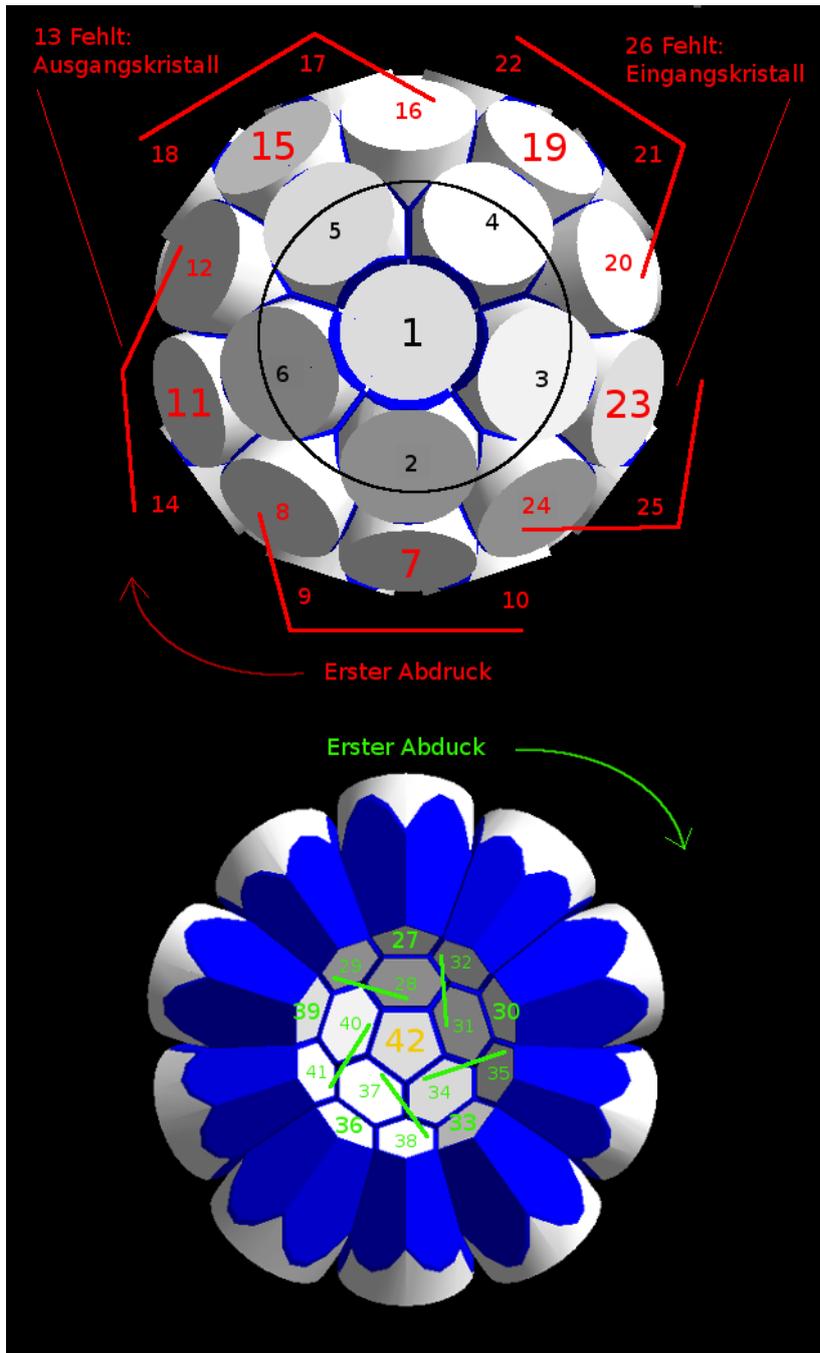


Abbildung 4.7.: Entwurf für die Zusammensetzung des Kalorimeters. Oben sieht man die Hälfte der Kugel nach der Geant4-Ansicht. Unten ist die andere Hälfte in ihrer ursprünglichen Stelle. Schwarz markiert sind die Detektoren des ersten Musters, rot sind alle Abbildungen des zweiten Musters, grün die des dritten und gelb das vierte Muster. Die großen Zahlen stellen Kristall A der Abbildung dar und die Linien verbinden die entsprechenden Reihen der Kristalle vom Typ B um den Hauptkristall.

physische Volumina an den logischen Volumina anderer Körper hängen. Dieses Modell fördert den hierarchischen Aufbau von Geometrien.

Diese übliche hierarchische Methode von Geant4 eignet sich aber nur um reine geometrische Teile zu entwerfen, die keine sensitive Funktion haben. Man kann nicht direkt die Detektoren an ein Muster hängen und danach das Muster ans World-Volumen hängen, da auf diese Weise nur das Muster und ihr entsprechendes Material sensitiv gegenüber der Quelle sein darf. Geant4 verfügt über Optionen, um Wiederholungen zu definieren, die aber nur für lineare und zylindrische Symmetrien nützlich sind. Für ein kugelsymmetrisches Kalorimeter, wie das an FRANZ eingesetzt, bietet Geant4 die Klasse `G4AssemblyVolume` an, die als Muster von Detektoren wirkt. Diese Klassen werden ohne Material definiert, sodass nur die Position zwischen den Tochervolumina von Bedeutung ist.

Beim Aufbau des Musters und der gesamten Kugel ist zudem zu beachten, dass die Zwischenräume zwischen den Kristallen eine Vergrößerung des inneren Radius erfordern, damit sich die einzelnen Teile beim Aufbau nicht überlappen. Um die exakte Vergrößerung herauszufinden, wird zunächst die gesamte innere Fläche berechnet und dann die neue größere Fläche mit den zusätzlichen Zwischenräumen mit den Gleichungen (B.51) bzw. (B.52). Danach errechnet sich die Vergrößerung des inneren Radius nach dem in der Gleichung (B.53) dargestellten Verhältnis.

Hier muss aufgepasst werden, dass die einzelnen Detektoren nach dem ursprünglichen inneren Radius hergestellt werden. Erst nachdem alle Teile geschaffen sind, darf man den inneren Radius für die Zusammensetzung des Kalorimeters vergrößern.

#### 4.3.2.1. Platzierung der Volumina im Muster

Man benötigt zwei Parameter: Rotation und Position. Die Beschreibung der Position in Geant4 erwartet kartesische Koordinaten und in diesem Verfahren eignen sich die sphärischen Koordinaten. So werden die üblichen Transformationen durchgeführt. Die Rotation hingegen wird mit drei Rotationswinkeln definiert. Diese drei Winkel sind unabhängig voneinander und die folgenden Drehungen werden in dieser Reihenfolge durchgeführt: phi, theta und psy.

- phi der Rotation: von der +Y-Achse bis zur +X-Achse.
- theta der Rotation: von der +Z-Achse bis zur +Y-Achse.
- psy der Rotation: wie phi, aber nachdem die Drehung von theta durchgeführt wurde.

Auch wenn in der Dokumentation dieselben Namen angewendet werden, sind diese Winkel nicht gleich den Winkeln aus den sphärischen Koordinaten der Positionierung. Für die Platzierung eines Volumens im Muster gelten aber immer die folgenden Identitäten:

- phi der Rotation: Korrektur.
- theta der Rotation:  $-\theta$  der Position in sphärischen Koordinaten.
- psy der Rotation:  $-(\phi \text{ der Position in sphärischen Koordinaten} + 90^\circ)$

Damit die Kristalle vom Typ B die gewünschte Rotation erreichen, soll zuerst die Drehung von theta durchgeführt werden und danach die von psy, also  $0^\circ$  in phi. Die Drehung in psy wird so gebraucht, da bei Kristall B, der sich in der  $-90^\circ$ -Stellung befindet, keine Rotation vorkommt. Für die Kristalle vom Typ A gibt man dieselbe Angabe an, aber mit dem Winkel  $90^\circ$  als phi, der als Korrektur des von den Kristallen von Typ B benötigten Ausdrucks in psy wirkt.

Der Winkel theta nach den sphärischen Koordinaten ist hierbei konstant und lässt sich mit der Gleichung (B.54) berechnen. Der Winkel phi vergrößert sich um  $72^\circ$  mit jedem Schritt der Kristalle vom Typ B, was den Winkeln zwischen den Seiten eines Pentagons entspricht.

#### 4.3.2.2. Platzierung der Abbildungen eines Musters im Hauptvolumen

Man benötigt hier keine Translation, sondern nur die Rotation des Musters. Daher werden die Rotationswinkel wie von Geant4 erwartet ohne Transformation angegeben. Das hat auch die folgende Auswirkung: Der Aufbau eines Musters wird als Funktion der Translationswinkel in sphärischen Koordinaten erstellt und daher wird das Muster gegen den Uhrzeigersinn gefüllt, wie in der Abbildung 4.6 gezeigt. Beim Aufbau der Kugel werden die Muster aber im Uhrzeigersinn abgebildet, wie in der Abbildung 4.7 gezeigt, was dem normalen Verlauf des von Geant4 erwarteten Rotationswinkeln entspricht.

Der Winkel theta der Rotation, der dem Winkel zwischen Kristall 1 nach der Abbildung 4.7 und dem Hauptkristall eines neuen Musters entspricht, beträgt  $-2 \cdot \text{patternTheta}$  und  $180^\circ - 2 \cdot \text{patternTheta}$  für das Muster 2 bzw. 3, wobei  $\text{patternTheta}$  (Gleichung (B.54)) der Winkel zwischen Kristall A und den Kristallen vom Typ B in einem Muster ist. Dieser Winkel ist konstant für alle Abbildungen eines Musters. Der Winkel phi entspricht der anfänglichen Orientierung des Musters (eine Drehung von  $180^\circ$  für die Muster 2 und 3, wie in der Abbildung 4.7 gezeigt), während der Winkel psy mit  $0^\circ$  anfängt und sich erhöht, wie der Winkel phi beim Aufbau eines Musters, also um  $72^\circ$  mit jedem Schritt.

Es gibt ein kleines Hindernis mit dem tatsächlichen Aufbau nach diesem Modell: Es gibt zwei Löcher in verschiedenen Abbildungen des zweiten Musters, den Eingangs- bzw. den Ausgangskristall. Dafür geht man folgenderweise vor: Jedes Mal, wenn bei der Schleife dieses Musters eine Abbildung gemacht wird, wird das Muster je nach den Bedingungen neu definiert. In Wirklichkeit wird dann das Assembly-Volumen insgesamt achtmal definiert. Trotz dieses Hindernisses ist es sinnvoll, die Vorteile dieser Klasse von Geant4 zu nutzen.

In der Abbildung 4.8 wird die durch die Simulation erstellte Kugel aus der Sicht des Eingangskristall gezeigt.

## 4.4. Sensitives Detektorvolumen

Um das Verhalten eines Kalorimeters zu simulieren, wird das Modell des von Geant4 zur Verfügung gestellten Beispiels N03 benutzt. Die wichtigsten Einzelheiten über das Verfahren werden in den folgenden Klassen dargestellt:

- `SteppingAction`: Diese Klasse steuert die Strecke der Teilchen, die aus der Quelle heraus-

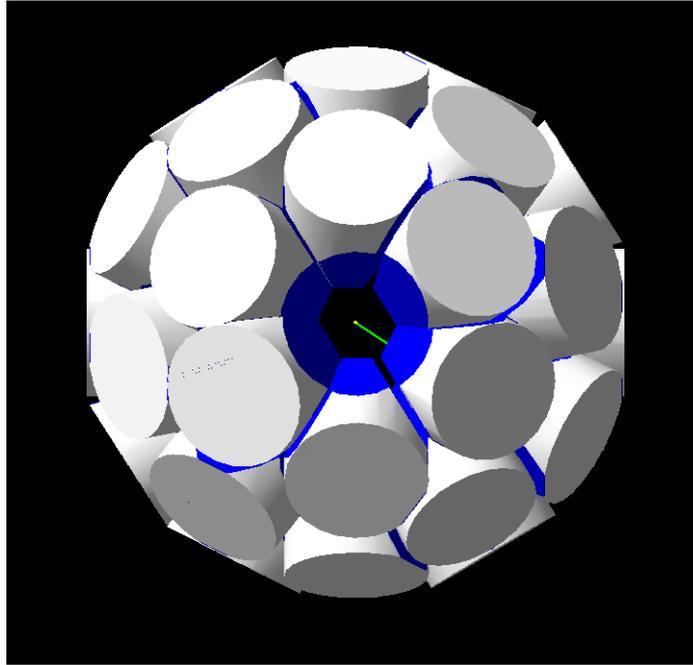


Abbildung 4.8.: OpenGL-Bild mit einer Ansicht des gesamten Kalorimeters aus Perspektive des fehlenden Eingangskristall.

geschlagen werden. Schritt für Schritt informiert sie u.a. über die deponierte Energie aus der Wechselwirkung mit der Umgebung nach den definierten physischen Prozessen.

- **EventAction:** Ein Event stellt das komplette Durchlaufen eines Teilchens im Kalorimeter dar. Jedes Mal, wenn in `SteppingAction` eine ausreichende Wechselwirkung mit dem  $\text{BaF}_2$ -Kristall erkannt wird, wird `EventAction` informiert und verarbeitet am Ende des Events die ganze Information dieses Events.
- **RunAction:** Ein sogenannter Run stellt die Gruppe der gesamten Events einer Simulation dar. `EventAction` stellt dieser Klasse am Ende eines Events alle nötigen Informationen dieses Events zur Verfügung und `RunAction` verarbeitet am Ende des Runs die ganze Information über alle Events.
- **PrimaryGeneratorAction:** Es gilt als Auslöser eines Events. In dieser Klasse werden die Quelle und alle ihre Eigenschaften definiert. Eine Methode dieser Klasse wird am Anfang jedes Events aufgerufen, um die Eigenschaften der Quelle wieder wie gewünscht einzustellen.
- **PhysicsLists:** In dieser Klasse werden sowohl alle Teilchen definiert, die in der Simulation auftreten, als auch die betrachteten physischen Prozesse, die mit den Teilchen verknüpft sind.
- **DetectorConstruction:** In dieser Klasse wird die Geometrie des Kalorimeters definiert, sowie die Materialien die dafür angewendet werden.

Die Klasse `AssemblyVolume`, die dem Aufbau des kugelförmigen Kalorimeters dient, verteilt selbst Namen für die bestehenden Teile, darunter die sensitiven Teile. Damit die Klasse `SteppingAction` einen sensitiven Teil des Kalorimeters erkennt, werden beim Entwurf der Geometrie in `DetectorConstruction` die Namen aller Kristalle verarbeitet, indem am Anfang der Namen eine Zahl steht.

Die Erkennung eines sensitiven Teils erfolgt dann mittels eines Vergleichs zwischen Daten vom Typ char, um die Rechnung am schnellsten zu schaffen.

## 4.5. Quelle

Es werden zwei Sorten von Quellen benutzt: Eine Punktquelle und eine Scheibe eines aktiven Materials. Die Materialien, die für diese Simulation zunächst von Interesse sind, sind  $^{60}\text{Co}$  und  $^{22}\text{Na}$ .

### 4.5.1. Punktquelle

Es wird in diesem Fall ein Photon als Quelle ausgewählt.

Mit einer Punktquelle kann man entweder eine vorprogrammierte Energie oder eine Anfangsenergie beliebig auswählen. Die schon vorhandenen Energiewerte, die für diese Simulation von Interesse sind, kommen aus dem Zerfall von  $^{60}\text{Co}$  und  $^{22}\text{Na}$ . Durch den Zerfall von  $^{60}\text{Co}$  werden 2 Photonen mit den Energien 1173 keV bzw. 1332 keV emittiert. Durch den Zerfall von  $^{22}\text{Na}$  werden auch zwei Photonen emittiert, die die Energien 511 keV und 1275 keV aufweisen. Jeder dieser vier Energiewerte darf als Anfangsenergie direkt ausgewählt werden.

Wird nun eine Punktquelle in die Mitte des Kalorimeters gesetzt, werden diese Teilchen in jedem Event in eine zufällige Richtung emittiert. Es wird hierfür die übliche Methode der Programmiersprache C++ statt der Geant4-Implementierung benutzt, um beliebige Richtungen in jedem Event zu generieren.

### 4.5.2. Aktive Scheibe

Das Material darf entweder  $^{60}\text{Co}$  oder  $^{22}\text{Na}$  sein. Statt eines Photons wird in diesem Fall ein Ion ausgewählt, das sich erst vor jedem Event als ein Ion von  $^{60}\text{Co}$  oder  $^{22}\text{Na}$  definiert. Für die Scheibe wird zunächst ein Radius von 1 cm und eine Dicke von  $1\ \mu\text{m}$  ausgewählt, sodass man eine ausgedehnte Quelle erhält.

Es wird in diesem Fall 0 als Anfangsenergie gewählt und Geant4 berechnet nach den definierten physischen Prozessen (siehe Abschnitt 4.6) sowohl die vorhandenen Energien als auch die Flugrichtungen.

## 4.6. Physische Prozesse

Es werden elektromagnetische Standardprozesse für die verschiedenen Teilchensorten ausgewählt. Insbesondere den Photonen, die für den Fall einer Punktquelle angewendet werden, werden der Photoelektrische Effekt, die Compton-Streuung und die Paarbildung zugewiesen.

Den Ionen hingegen werden mehrfache Streuung und Ionisation zugewiesen. In diesem Fall werden aber nicht nur elektromagnetische Wechselwirkungen definiert, sondern auch Zerfälle, insbesondere radioaktive Zerfall.

## 4.7. Antwort der Simulation

Die ausgegebene Information pro Event ist im Standard-Output vorhanden, aber im Prinzip kommentiert, da sie für eine hohe Anzahl von Events einerseits unlesbar wird und sich andererseits der Run unnötigerweise verlangsamt. Diese Information besteht aus: der deponierten Energie, der mittleren Weglänge und der Multiplizität.

Pro Run wird zuerst die Information pro  $\text{BaF}_2$ -Kristall gezeigt und dann vom ganzen Kalorimeter. Beim Bericht pro Kristall haben die Kristalle dieselbe Reihenfolge wie in der Abbildung 4.7, aber die Aufzählung fängt mit 0 an und ignoriert die fehlenden Kristalle. Es wird dabei keine Anpassung an die Abbildung vorgenommen, um den Run nicht langsamer zu machen. Die Information pro Kristall besteht aus dem Typ des Kristalls und dem entsprechenden Prozentsatz der totalen deponierten Energie, die in diesem Kristall deponiert wurde.

Die Information über das Kalorimeter ist die folgende:

- **Zeitdauer:** Sie wird vom Anfang bis Ende des Runs gemessen. Die Geometrie des Detektors wird vor dem ersten Event aufgebaut und ihre Zeitaufwendung wird somit nicht mitberechnet. Die Zeitdauer wird mit der üblichen Methode der Programmiersprache C++ berechnet.
- **Mittlere Energie:** Ein Durchschnittswert der deponierten Energien aller Events.
- **Mittlere Weglänge:** Auch von Interesse in derartigen Simulationen kann die Weglänge sein, die das Teilchen durch das Kalorimeter durchlaufen hat.
- **Anzahl getroffener Kristalle:** Mit wie vielen  $\text{BaF}_2$ -Kristallen hat die Quelle insgesamt gewechselwirkt.
- **Mittlere Multiplizität:** Neben der deponierten Energie ist auch die Multiplizität von Interesse. Sie gibt an, wie viele Kristalle mit der Quelle pro Event in Wechselwirkung treten. Die zusätzliche Anwendung pro Event, die die Erkennung der Multiplizität braucht, wird mit Indizes vom Datentyp short implementiert, um den Run für eine hohe Anzahl von Events möglichst schnell zu machen.

Falls eine Punktquelle untersucht wird, gibt es eine bekannte Anfangsenergie. Daher ist auch die folgende zusätzliche Information, die in diesem Fall mitberechnet wird, von Interesse:

- **Effizienz:** Prozentsatz der Events, bei denen überhaupt Energie deponiert wurde. Erwartet werden nach dem Modell in Ref. [5] präsentiert rund 90%.
- **Volleffizienz:** Prozentsatz der Events, bei denen die gesamte Anfangsenergie deponiert wurde. Erwartet werden in diesem Fall etwa 50% der Events.

- Energieprozentsatz: Prozentsatz der Anfangsenergie, der deponiert wurde.

Für jeden Run aller Arten werden ROOT-Dateien erstellt. Zum einen werden Histogramme für jeden einzelnen Detektor sowie für das gesamte Kalorimeter erstellt. Dabei ist der Unterschied zwischen den Peaks und anderen Werten groß, weshalb die Aufzählung logarithmisch angezeigt wird.

Zum anderen wird eine TTree-Struktur mit der gesammelten Information aller Events und aller Detektoren gespeichert. Eine README-Datei mit allen Anweisungen und erklärten Versuchsoptionen wurde erstellt.

## 5. Ergebnisse

Für eine komplexe Geometrie wie diese werden nicht mehrere Millionen Events simuliert, was zum Vergleich mit den durchgeführten Messungen noch besser wäre, jedoch ist die Analyse einer akzeptablen Anzahl von Events möglich. Hierfür wird zusätzlich die Zeit berechnet, die die Simulation benötigt. Für eine Punktquelle werden im Durchschnitt etwa 3 Sekunden pro 1000 Events gebraucht. Für eine reale Quelle, bei der Zerfälle simuliert werden, werden im Durchschnitt etwa 6 Sekunden pro 1000 Events für  $^{60}\text{Co}$  und etwa 7 Sekunden pro 1000 Events für  $^{22}\text{Na}$  gebraucht. Nach wenigen hunderttausenden Events bricht aber die Simulation aus Mangel an Speicherplatz im Rechner ab. Für die folgenden Analysen werden zunächst immer 10000 Events ausgewählt. Dabei werden immer alle 40 Detektoren getroffen.

Die Eigenschaften des Rechners sind entscheidend für den Zeitaufwand. In diesem Fall handelt es sich um die folgenden:

- Prozessor: Intel Core i5-3450. 3.10 GHz (x4)
- RAM Speicherplatz: 3.8 GB
- Vorhandener Speicherplatz im Moment der Analyse: etwa 850 GB
- Betriebssystem: Ubuntu 12.04 64-bit, Kernel Linux 3.2.0-54-generic, GNOME 3.4.2

**Punktquelle:** Zunächst werden vier Fälle mit einer Punktquelle betrachtet: Energien der Photonen aus dem Zerfall von  $^{60}\text{Co}$  und aus dem Zerfall von  $^{22}\text{Na}$ . Die Ergebnisse dieser Versuche werden in den Tabellen 5.1 bzw. 5.2 gezeigt. Die Effizienz liegt, wie erwartet, um die 90 %. Die Volleffizienz ist eher unterschiedlich, jedoch liegt sie üblicherweise über den vorhergesagten 50 %.

In der Abbildung 5.1 sieht man die Energiespektren des gesamten Kalorimeters für die vier Fälle.

Eigenschaft	$^{60}\text{Co} - \gamma 1$ (1173 keV)	$^{60}\text{Co} - \gamma 2$ (1332 keV)
Zeitdauer	36 Sek.	28 Sek.
Mittlere Energie	(902.39 $\pm$ 449.83) keV	1.02 MeV $\pm$ 508.96 keV
Mittlere Weglänge	(980.67 $\pm$ 535.21) $\mu\text{m}$	1.16 mm $\pm$ 631.27 $\mu\text{m}$
Mittlere Multiplizität	2.40	2.46
Effizienz	85 %	86 %
Volleffizienz	63 %	49 %
Energieprozentsatz	76.93 %	76.73 %

Tabelle 5.1.: Ergebnisse der Simulation einer Punktquelle mit den Energien von  $^{60}\text{Co}$ . 10000 Events

Eigenschaft	$^{22}\text{Na} - \gamma 1$ (511 keV)	$^{22}\text{Na} - \gamma 2$ (1275 keV)
Zeitdauer	25 Sek.	28 Sek.
Mittlere Energie	$(433.69 \pm 167.87)$ keV	$(978.99 \pm 487.22)$ keV
Mittlere Weglänge	$(338.09 \pm 157.11)$ $\mu\text{m}$	$1.09 \text{ mm} \pm 595.21 \mu\text{m}$
Mittlere Multiplizität	2.22	2.46
Effizienz	89 %	86 %
Volleffizienz	75 %	51 %
Energieprozentsatz	84.87 %	76.78 %

Tabelle 5.2.: Ergebnisse der Simulation einer Punktquelle mit den Energien von  $^{22}\text{Na}$ . 10000 Events

Eigenschaft	$^{60}\text{Co}$	$^{22}\text{Na}$
Zeitdauer	64 Sek.	66 Sek.
Mittlere Energie	$1.92 \text{ MeV} \pm 699.24 \text{ keV}$	$1.87 \text{ MeV} \pm 638.22 \text{ keV}$
Mittlere Weglänge	$2.13 \text{ mm} \pm 847.23 \mu\text{m}$	$1.79 \text{ mm} \pm 698.54 \mu\text{m}$
Mittlere Multiplizität	4.86	7.33

Tabelle 5.3.: Ergebnisse der Simulation realer Quellen im gesamten Kalorimeter. 10000 Events

Die Anfangsenergien weisen einen eindeutigen Peak auf. Andere Energiewerte unter dem Peak werden nur leicht aufgenommen und alle etwa gleich wahrscheinlich. Dies entspricht der Erwartung, dass der Peak bei der vollen Energie liegt und außerdem das Compton-Kontinuum sichtbar wird.

**Reale Quellen:** Dann werden beide realen Quellen getestet. In der Tabelle 5.3 sieht man die Ergebnisse für das gesamte Kalorimeter und in der Tabelle 5.4 die deponierte Energie in jedem einzelnen Detektor. Alle Detektoren empfangen nah beieinander liegende Energiewerte.

In der Abbildung 5.2 sieht man die Energiespektren des gesamten Kalorimeters für die beiden realen Quellen. In diesem Fall erkennt man einen Peak nicht nur bei den Energien der zugehörigen Photonen aus den Zefällen, sondern noch wahrscheinlicher tritt die Summe beider Energien auf. Dies entspricht dem erwarteten Summing-Effekt, da die Photonen mit geringerem zeitlichem Abstand als der Zeitauflösung des Detektors emittiert werden.

Im Fall von  $^{22}\text{Na}$  treten zusätzlich andere Summen auf, die nur dank des bedeckten Raumwinkels möglich nachzuweisen sind. Sie erfordern mindestens einen  $180^\circ$  Raumwinkel, da die beiden Photonen mit 511 keV unter diesem Winkel bei der Positronenannihilation emittiert werden. Das Compton-Kontinuum ist in diesem Fall ebenso zu erkennen.

Nach den höchsten Summenlinien sind zusätzlich Energiewerte zu erkennen, bei denen es sich möglicherweise um die emittierten Elektronen aus dem Zerfall handelt, die im Kalorimeter auch nachgewiesen werden.

In den Abbildungen 5.3 und 5.4 werden typische Flugbahnen für nur 10 Events im Fall von  $^{60}\text{Co}$  bzw.  $^{22}\text{Na}$  dargestellt.

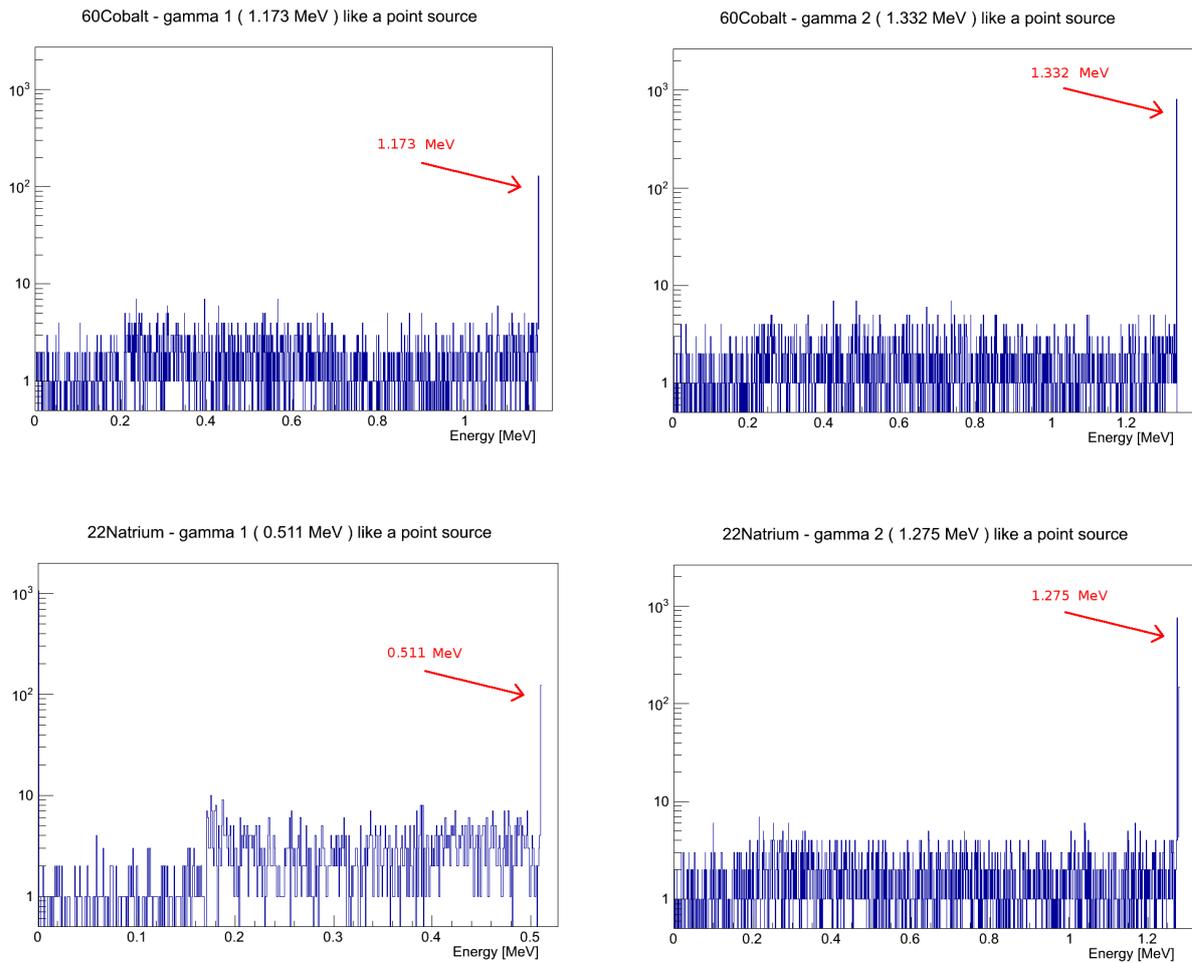
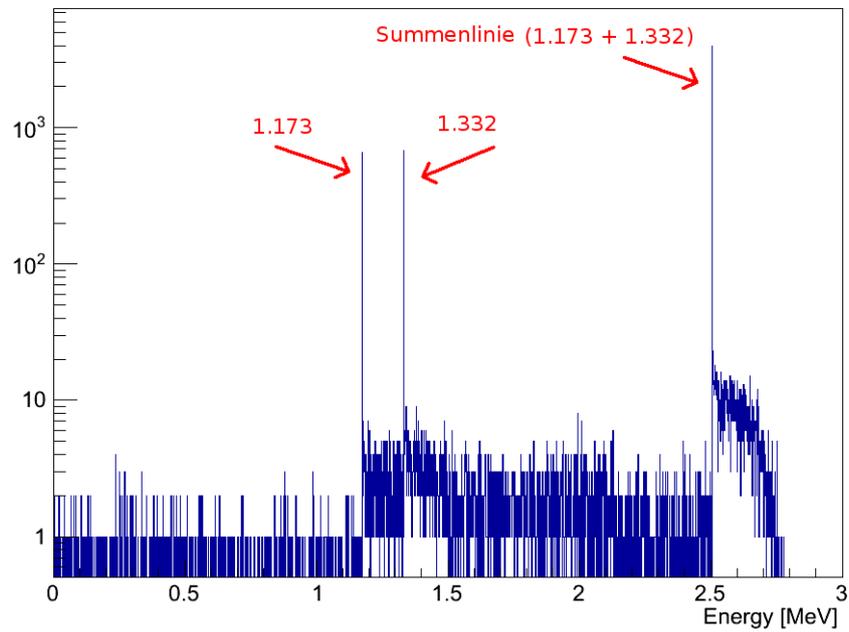


Abbildung 5.1.: Energiespektren der Photonen aus dem Zerfall von  $^{60}\text{Co}$  und  $^{22}\text{Na}$  als Punktquellen. Rot gekennzeichnet sind die Anfangsenergien, die einen Peak im Spektrum verursachen. Das Compton-Kontinuum ist dabei auch zu erkennen sowie die Kontinuität bis zum Peak aufgrund der Mehrfachstreuung.

Detektor (BaF <sub>2</sub> -Kristall)	<sup>60</sup> Co [%]	<sup>22</sup> Na [%]
1	2.66	2.61
2	2.69	2.51
3	2.76	2.60
4	2.75	2.65
5	2.74	2.55
6	2.43	2.47
7	2.49	2.45
8	2.65	2.59
9	2.40	2.45
10	2.48	2.21
11	2.31	2.53
12	2.68	2.61
13	Ausgangskristall	Ausgangskristall
14	2.60	2.42
15	2.52	2.37
16	2.59	2.46
17	2.22	2.46
18	2.40	2.60
19	2.61	2.34
20	2.41	2.52
21	2.64	2.40
22	2.18	2.44
23	2.61	2.61
24	2.34	2.45
25	2.23	2.38
26	Eingangskristall	Eingangskristall
27	2.44	2.52
28	2.56	2.67
29	2.42	2.57
30	2.21	2.49
31	2.50	2.62
32	2.39	2.47
33	2.26	2.49
34	2.55	2.52
35	2.68	2.51
36	2.24	2.51
37	2.71	2.31
38	2.53	2.41
39	2.21	2.35
40	2.50	2.54
41	2.41	2.44
42	2.78	2.68

Tabelle 5.4.: Prozentsatz der deponierten Energie pro Detektor mit den realen Quellen. Die Reihenfolge der Aufzählung der Detektoren ist dieselbe wie in der Abbildung 4.7.

### $^{60}\text{Co}$ by radioactive decay



### $^{22}\text{Na}$ by radioactive decay

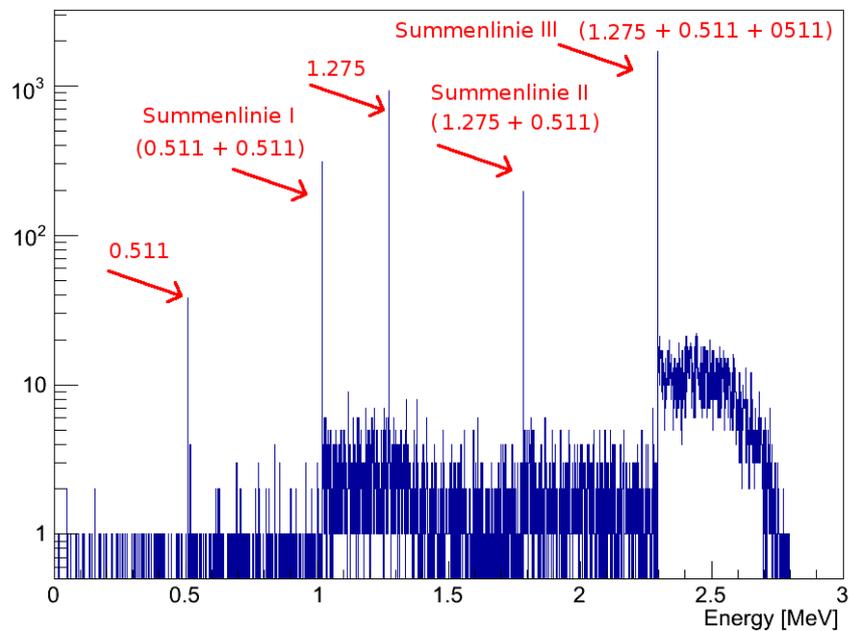


Abbildung 5.2.: Energiespektren aus dem Zerfall von  $^{60}\text{Co}$  und  $^{22}\text{Na}$ . Rot gekennzeichnet sind die Peaks, die den zwei emittierten Photonen aus den Zerfällen sowie der Summe zwischen ihnen entsprechen.

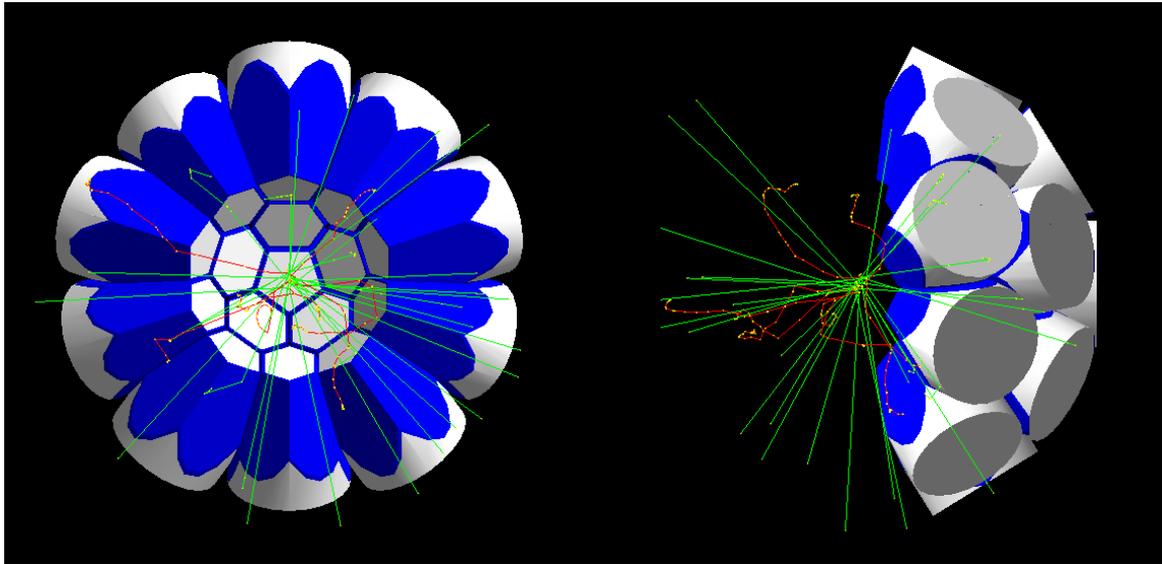


Abbildung 5.3.: Typische Flugbahn beim Zerfall von  $^{60}\text{Co}$ . Die Farben bezeichnen die Ladung des Teilchens. Grün: 0 (Photon), rot: -1 (Elektron), blau: +1 (Positron). 10 Zerfälle.

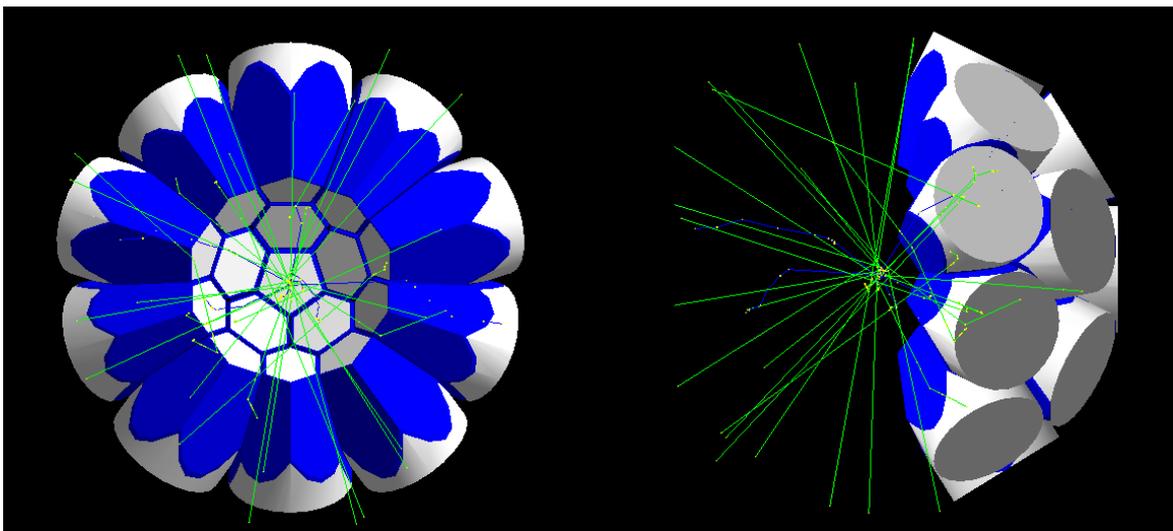


Abbildung 5.4.: Typische Flugbahn beim Zerfall von  $^{22}\text{Na}$ . Die Farben bezeichnen wie in der Abbildung 5.3 die Ladung des Teilchens. 10 Zerfälle.

## 5.1. Vergleich mit den gemessenen Daten aus der B.Sc.-Arbeit von Clemens Wolf

In der Abbildung 5.5 werden die Energiespektren von Clemens Wolf im gesamten Kalorimeter, bei denen allerdings vier Detektoren fehlen, mit denen von dieser Simulation in einer Grafik übereinandergelegt, um diese miteinander zu vergleichen. Die Peaks der Energien stimmen mit denen aus dieser Arbeit und den theoretisch erwarteten sehr gut überein.

Von diesen Spektren wurde zudem der Hintergrund abgezogen, was die entstehenden negativen Häufigkeiten und starken Schwankungen erklären kann. Die hohe Anzahl naheliegender Werte an den Peaks liegt an der Energieauflösung, die nicht simuliert wird.

Die Multiplizität wird zeitlich bestimmt. Dabei wird bei den ersten drei Millionen Events im Durchschnitt mit Abstand die höchste Multiplizität erreicht und diese nimmt anschließend kontinuierlich ab [2]. Nach dieser Beschreibung ist diese anfängliche Multiplizität mit der hier errechneten zu vergleichen, die nur bei 10000 Events angegeben wird.

In der Abbildung 5.6 ist ein Event-Histogramm zu sehen, in dem ein Durchschnittswert nicht explizit angegeben wird, aber maximal um 2 eingeschätzt werden kann. Dieses Resultat ist klein im Vergleich mit den in dieser Arbeit erhaltenen Multiplizitäten. Die für  $^{60}\text{Co}$  liegt um 5. Für den Unterschied sollten nicht nur die Energieverluste betrachtet werden, sondern auch die Effizienz der Detektoren, die im Rahmen dieser B.Sc.-Arbeit durchschnittlich bei kaum 67,01 % für eine Energie von 1275 keV bestimmt wurde. Dafür stimmt dieses Ergebnis aber besser mit den von Anfang an vorhergesagten Multiplizitäten in Ref. [5] überein.

## 5.2. Vergleich mit Simulationen in Geant3

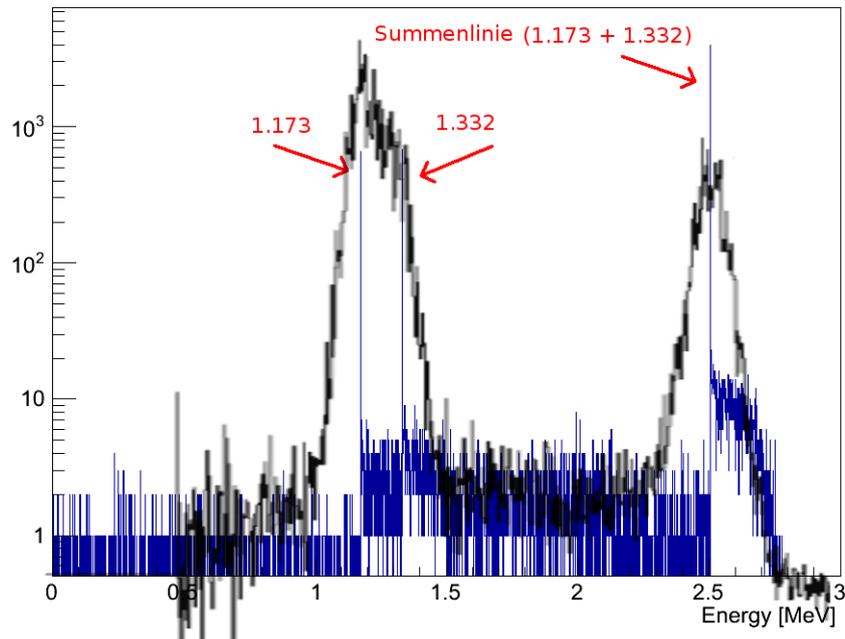
Was die Geometrie des Kalorimeters betrifft, gibt es ein paar Unterschiede. In [5] wurde am Anfang geplant, die fehlenden Kristalle durch ein zylindrisches Loch zu ersetzen, um den bedeckten Raumwinkel zu steigern. Das wurde bisher noch nicht unternommen. Einerseits sind die fehlenden Kristalle so simuliert wie im gegenwärtigen Zustand des Kalorimeters und andererseits werden in dieser Simulation sowohl der Schlitz als auch die zusätzlichen Materialien in diesem Bereich mitsimuliert. Außerdem sind beliebige Dimensionierungen wie im Abschnitt 6 erklärt, erlaubt.

Für den Versuch mit einzelnen Photonen wurde der Protonenstrahl aus dem Hochfrequenz-Beschleuniger direkt in eine Probe von  $^{26}\text{Mg}$  geschossen. Eine sehr gute Übereinstimmung der Simulation mit den Messungen, wie in der Abbildung 5.7 zu sehen ist, wurde dabei erhalten.

Im Gegensatz zu den Spektren aus dieser Simulation wurde jedoch eine bessere Übereinstimmung mit der Realität erreicht, indem eine stark zunehmende Wahrscheinlichkeit schon in der Nähe der Peaks aufgenommen wird. Wahrscheinlich wurde die typische Energieauflösung des Detektors mit dem simulierten Spektrum gefaltet, um diese Übereinstimmung zu erreichen.

Ein Zusammenhang verschiedener Anfangsenergien mit den errechneten Eigenschaften des Kalorimeters sind auch in [3] zu betrachten. Ein Vergleich mit den Ergebnissen aus dieser Simulation ist im Kapitel 6, zusammen mit anderen einstellbaren Eigenschaften, zu sehen.

### $^{60}\text{Co}$ by radioactive decay



### $^{22}\text{Na}$ by radioactive decay

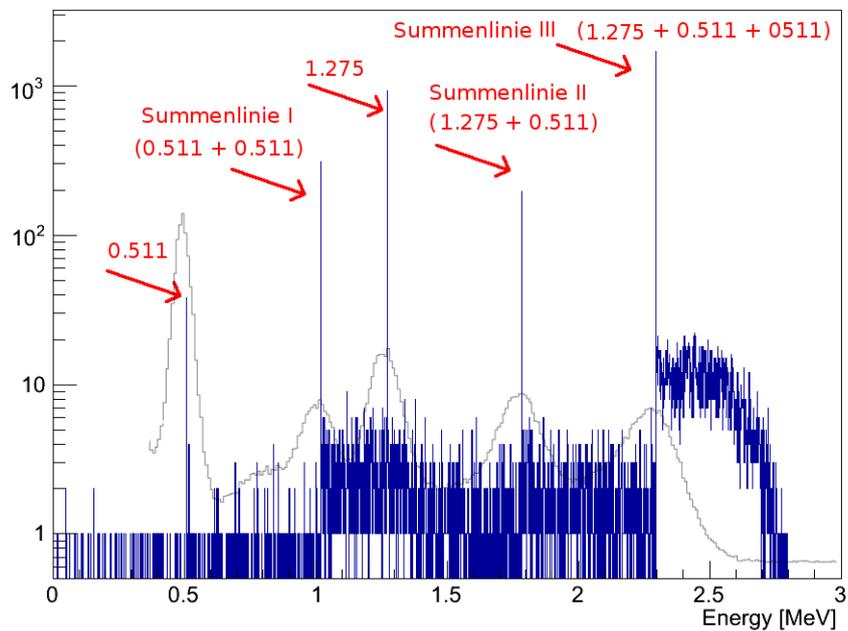


Abbildung 5.5.: Vergleich der Spektren aus der Abbildung 5.2 mit den Messungen aus der B.Sc.-Arbeit von Clemens Wolf. In Blau sind die Ereignisse dieser Simulation und Schwarz die Messungen mit abgezogenem Hintergrund dargestellt. Die Messungen sind mit derselben Energieskala versehen und unterscheiden sich nur in der Anzahl der Ereignisse.

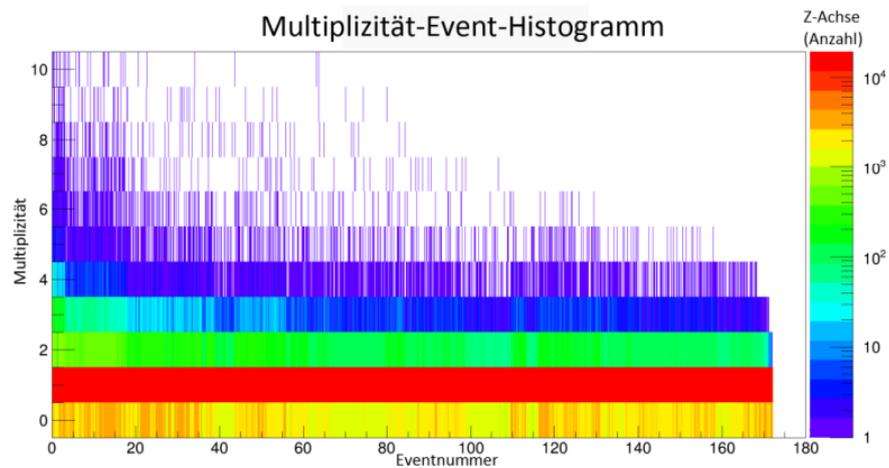


Abbildung 5.6.: Multiplizität-Histogramm. Die Anzahl der Events sollte in Millionen angegeben sein. Die Z-Achse zeigt die Häufigkeit, mit der die entsprechende Multiplizität bei dieser Anzahl von Events auftaucht [2].

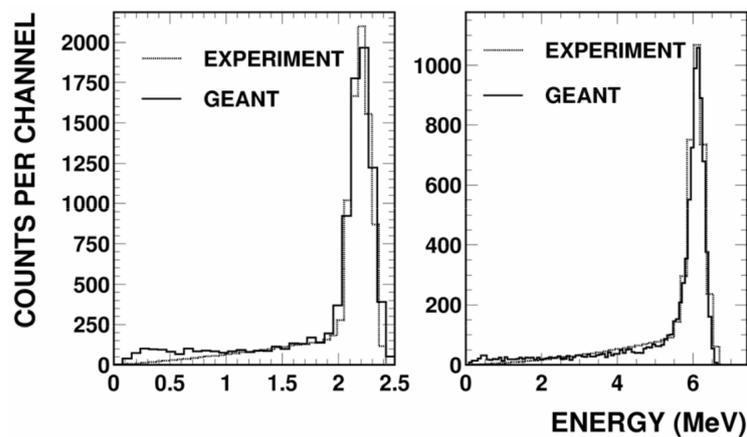


Abbildung 5.7.: Vergleich der Geant3-Spektren mit realen Messungen. Links sind mono-energetische  $\gamma$ -Strahlen mit 2.209 MeV und rechts mit 6.146 MeV dargestellt. Die durchgezogenen Linien zeigen die Ergebnisse aus den Simulationen und die gestrichelten Linien, die aus den Messungen [3].

## 6. Weitere Möglichkeiten der Simulation

Die Simulation bietet die Möglichkeit, einige Parameter zu verändern, um deren Einfluss auf die Ergebnisse untersuchen zu können. Dies erfolgt sowohl durch die Standardmethode von Geant4, die Bearbeitung einer Datei, als auch durch direkte Angabe beim Start des Programms. Alle Einzelheiten sind in der README-Datei enthalten. Die folgenden Parameter sind einstellbar:

- Dimensionierung des inneren Radius
- Dimensionierung der Detektorbreite
- Dimensionierung des Schlitzes
- Anwesenheit von Zwischenräumen zwischen den Kristallen
- Dimensionierung des Quellenradius
- Dimensionierung der Quellenbreite
- Art der Quelle
- Wahl des Materials
- Wahl der zu untersuchenden Photonen
- monoenergetische Anfangsenergie

**Eigenschaften der Geometrie:** Alle Dimensionierungen erlauben eine positive reale Zahl als Angabe. Nachdem alle angegeben wurden, wird untersucht, ob der Schlitz maximal das Fünffache des inneren Radius beträgt. Damit würde der Schlitz schon kurz davor stehen, den nächsten Kristall zu schneiden.

Allerdings gibt es Meldungen über Visualisationsprobleme bei manchen booleschen Operationen, die einen von den Entwicklern schon erkannten Fehler in der Implementierung von Geant4 betrifft. Dieser sollte in zukünftigen Versionen von Geant behoben sein. Zunächst darf man das Kalorimeter maximal mit einer Detektorbreite zweimal größer als ursprünglich richtig visualisieren.

In der Abbildung 6.1 kann man eine Darstellung des Kalorimeters mit dimensioniertem innerem Radius und Detektorbreite.

Wenn eine Oberfläche ohne Dicke in Geant4 angegeben wird, kann man sie trotzdem visualisieren. Daher gibt es eine Option, um das Kalorimeter ohne Zwischenraum zwischen den Kristallen zu betrachten.

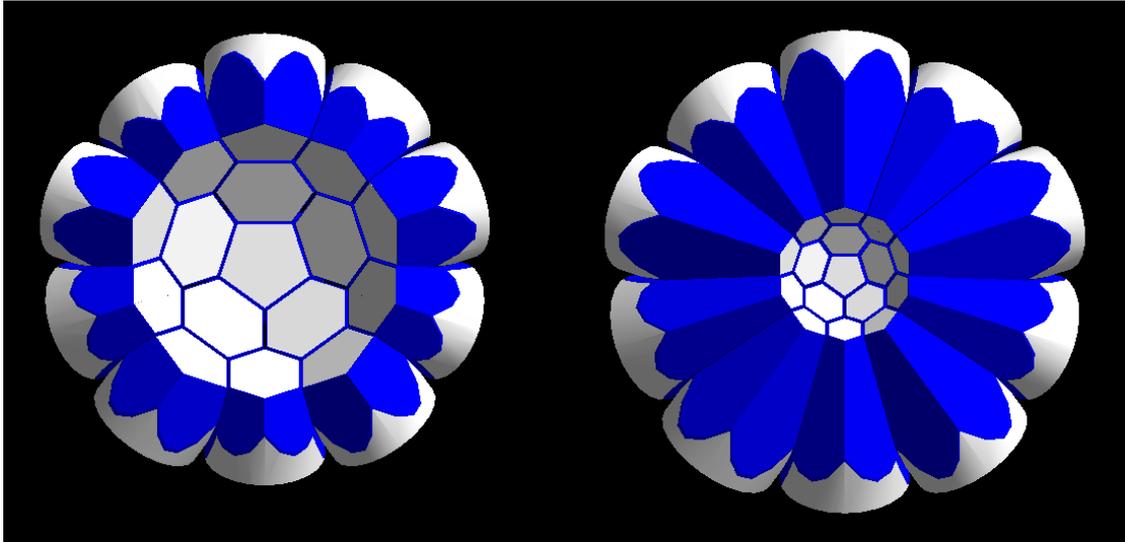


Abbildung 6.1.: Links die Ansicht eines halben Kalorimeter mit einem inneren Radius zweimal so groß wie die ursprünglichen 10 cm. Rechts mit einer Detektorbreite zweimal so groß wie die ursprünglichen 15 cm.

**Eigenschaften der Quelle:** Die Dimensionierungen der Quelle erlauben auch eine positive reale Zahl als Angabe. Nachdem alle angegeben wurden, wird diesmal untersucht, ob der Quellenradius nicht größer ist als zweieinhalbmals der innere Radius. Damit würde die Quelle kurz davor stehen, breiter als der Eingangskristall zu sein. Bei der Quellenbreite wird untersucht, ob sie nicht größer ist als  $2 \cdot 10^4$  Quellenradius. Damit würde die Quelle nicht innerhalb der inneren Kugel einpassen.

Die anderen vier Eigenschaften erlauben, die Optionen zwischen Runs einzustellen. Die Art von Quelle darf entweder Punktquelle oder eine reale Quelle sein. Als Material darf entweder  $^{60}\text{Co}$  oder  $^{22}\text{Na}$  ausgewählt werden. Als Photon kann man zwischen dem ersten und dem zweiten Photon aus dem Zerfall des ausgewählten Materials entscheiden.

Durch eine einstellbare Anfangsenergie als Punktquelle kann man zudem eine Analyse über die Eigenschaften des Kalorimeters mit verschiedenen Energien durchführen. In der Tabelle 6.1 sieht man die Ergebnisse dieser Analyse aus der Simulation in Geant3. Die Tabelle 6.2 zeigt zum Vergleich die Evolution der Antwort der in dieser Arbeit entwickelten Simulation.

Die Effizienz sinkt mit steigender Energie in beiden Simulationen, ebenso die Volleffizienz, allerdings stärker als die Effizienz. Die mittlere Multiplizität steigt dagegen in beiden Simulationen. In dieser Simulation, wie in der Geant3-Simulation, sinkt die Volleffizienz mit steigender Energie schneller. Allgemein weist diese Simulation leicht kleinere Effizienzen und leicht größere Multiplizitäten auf.

<b>Energie [MeV]</b>	<b>Effizienz [%]</b>	<b>Volleffizienz [%]</b>	<b>~ Multiplizität</b>
0.5	94	90	1.2
1.0	92	83	1.3
2.0	89	73	1.4
3.0	87	70	1.5
4.0	87	68	1.6
5.0	86	64	1.7
6.0	86	63	1.8
8.0	87	62	1.9
10.0	87	62	2.0

Tabelle 6.1.: Entwicklung der simulierten Effizienz, Volleffizienz und der mittleren Multiplizität in der Geant3 Simulation für Photonen verschiedener Energien [3].

<b>Energie [MeV]</b>	<b>Effizienz [%]</b>	<b>Volleffizienz [%]</b>	<b>~ Multiplizität</b>
0.5	88	74	2.22
1.0	86	65	2.40
2.0	83	52	2.57
3.0	80	31	2.76
4.0	79	24	2.94
5.0	80	24	3.24
6.0	79	16	3.46
8.0	79	18	3.91
10.0	80	13	4.37

Tabelle 6.2.: Entwicklung der Multiplizität, Effizienz, Volleffizienz und der mittleren deponierten Energie im gesamten Kalorimeter für Photonen verschiedener Energien in dieser Simulation. 10000 Events.

# Literaturverzeichnis

- [1] <http://exp-astro.physik.uni-frankfurt.de>. Information über das Franz-Projekt, physikalische Grundlagen über hochenergetische Reaktionen, vollendete Bachelor- und Master-Arbeiten. Aufgerufen im April 2013.
- [2] C. Wolf, "Charakterisierung des 42-fach segmentierten  $4\pi$  BaF<sub>2</sub>-Detektors für FRANZ," Mai 2013. Bachelor Abschluss Arbeit an der J. W. Goethe Universität, Frankfurt am Main.
- [3] R. Reifarh, R. Plag, M. Heil, F. Käppeler, F. Voss, and K. Wisshak, "Geant Simulations of the Karlsruhe 4p BaF2 detector," tech. rep., Los Alamos National Laboratory (LANL), Los Alamos, New Mexico, 87545, USA, Oktober 2004. In der LANL Bibliothek eingereicht.
- [4] <http://de.wikipedia.org/>. Photoelektrischer Effekt / Compton-Effekt / Paarbildung / Szintillator / Photomultiplier. Aufgerufen im November 2013.
- [5] K. Wisshak, K. Guber, F. Käppeler, J. Krisch, H. Müller, G. Kupp, and F. Voss, "The Karlsruhe  $4\pi$  barium fluoride detector," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 292, pp. 595–618, März 1990. Kernforschungszentrum Karlsruhe, Institut für Kernphysik III, P.O.B. 3640, 7500 Karlsruhe, Deutschland.
- [6] <http://geant4.cern.ch>. Source Code, Installation- und User-Handbuch von Geant4. Aufgerufen im April 2013.
- [7] <http://www.slac.stanford.edu/~perl/heprep/>. Beschreibung vom HepRep Format. Aufgerufen im Juni 2013.
- [8] <http://geant4.slac.stanford.edu/Presentations/vis/G4HepRAppTutorial/G4HepRAppTutorial.html>. Geant4 Visualisierungstutorial mit dem HepRApp HepRep-Browser. Aufgerufen im Juni 2013.
- [9] <http://geant4.slac.stanford.edu/Presentations/vis/G4OpenGLTutorial/G4OpenGLTutorial.html>. Geant4 Visualisierungstutorial mit dem System OpenGL. Aufgerufen im Oktober 2013.
- [10] <http://root.cern.ch>. Anweisungen über die Software root. Aufgerufen im Oktober 2013.

# A. Darstellung der Größen

In den Abbildungen A.1, A.2, A.3 und A.4 werden die für die Implementation oft verwendeten Größen veranschaulicht.

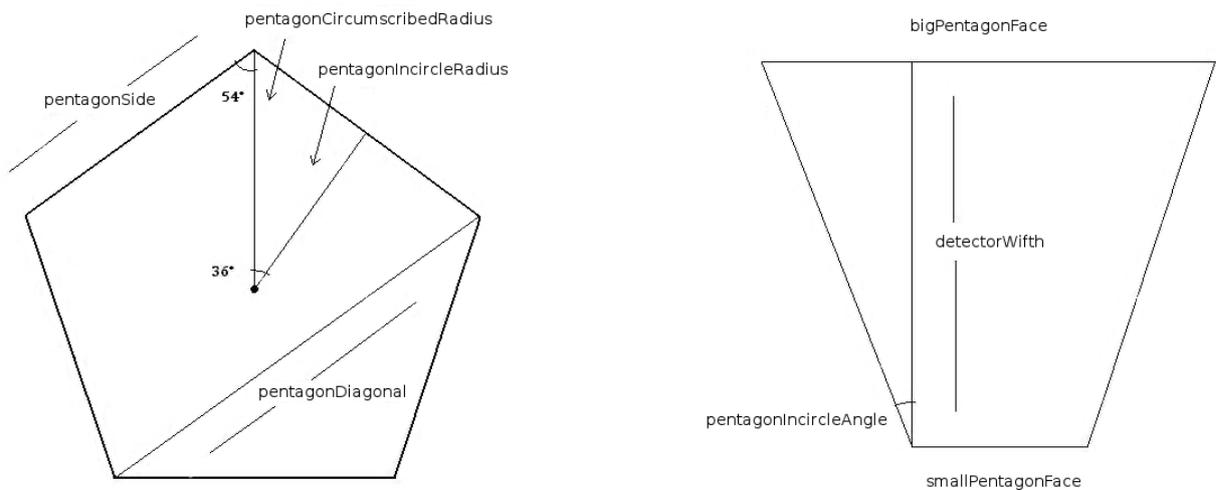


Abbildung A.1.: Größen des Kristalls A. Links die Grundfläche der Pyramide und rechts die transversale Sicht.

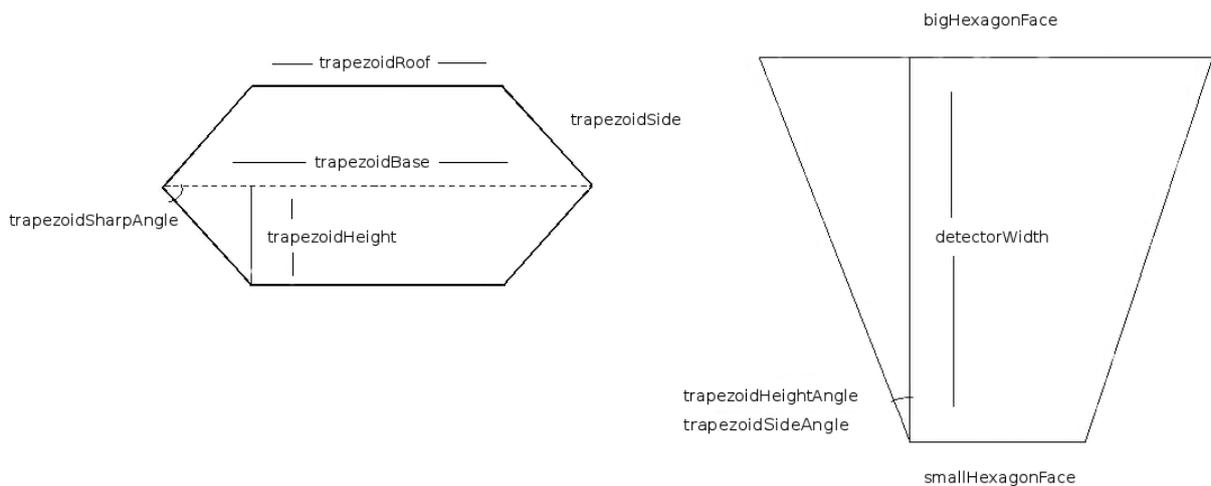


Abbildung A.2.: Größen des Kristalls B. Links die Grundfläche der Pyramide und rechts die transversale Sicht. Im Vergleich mit dem Kristal A, können zwei verschiedene Winkel entstehen, rechts im Bild, je nachdem ob die diagonale Linie einer Seite des Trapezoide oder seinem Dach entspricht.

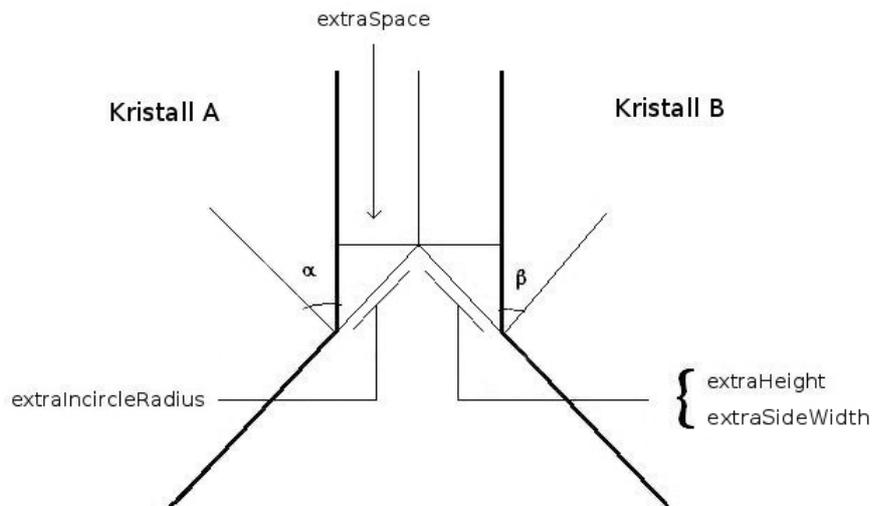


Abbildung A.3.: Abdeckungbezogene Größen. Damit der Aufbau der Kugel mit den Abdeckungen immer noch gelingt, müssen je nach dem erwünschten Wert vom Zwischenraum (`extraSpace` im Bild) die gekennzeichneten Verlängerungen der entsprechenden Größen addiert werden.  $\alpha$  und  $\beta$  im Bild stellen die in Abbildung A.1 und A.2 bei der transversalen Sicht gezeigten Winkel dar: `pentagonsIncircleAngle` im Fall vom Kristall A und `trapezoidHeightAngle` bzw. `trapezoidsSideAngle` im Fall von Kristall B.

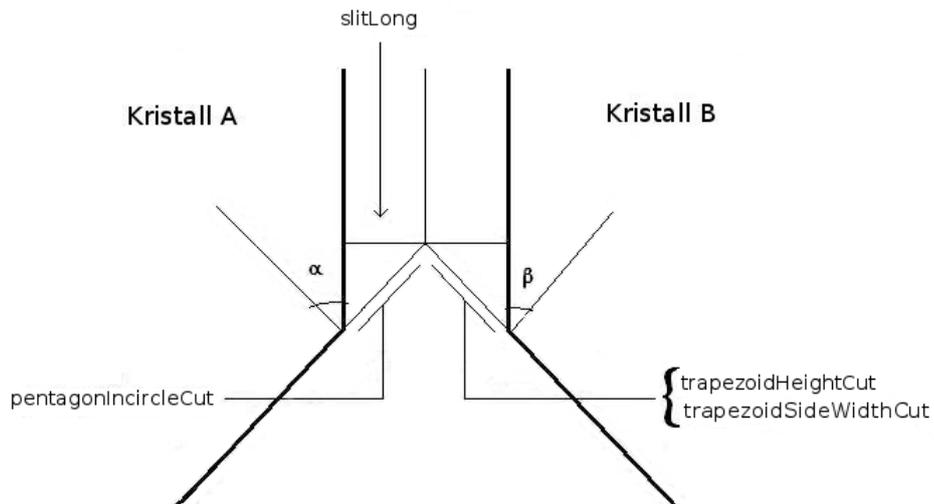


Abbildung A.4.: Schlitzbezogene Größen. Damit der Aufbau der Kugel mit dem Schlitz immer noch gelingt, müssen je nach seinem erwünschten Wert (`slitLong` im Bild) die gekennzeichneten Verkürzungen der entsprechenden Größen abgezogen werden.  $\alpha$  und  $\beta$  im Bild stellen dieselben Winkel wie in der Abbildung A.3 dar.

## B. Rechnung der Größen

Die folgenden Gleichungen zeigen, wie man jede Größe, die in den Abbildungen aus Anhang A gezeigt werden, sowie andere nützliche Größe für die Implementation, berechnet.

### Pentagonbezogene Größen:

$$pentagonIncircleRadius = \frac{pentagonSide}{2} \cdot \tan(54^\circ) \quad (B.1)$$

$$pentagonDiagonal = \frac{pentagonSide}{2} \cdot (1 + \sqrt{5}) \quad (B.2)$$

$$pentagonSurface = \frac{5}{2} \cdot pentagonCircumscribedRadius^2 \cdot \sin(72^\circ) \quad (B.3)$$

$$pentagonsIncircleAngle = \arctan\left(\frac{bigPentagonIncircleRadius - smallPentagonIncircleRadius}{detectorWidth}\right) \quad (B.4)$$

### Hexagonbezogene Größen:

$$trapezoidBaseExpansion = \frac{trapezoidBase - trapezoidRoof}{2} \quad (B.5)$$

$$trapezoidHeight = \sqrt{trapezoidSide^2 - trapezoidBaseExpansion^2} \quad (B.6)$$

$$trapezoidSharpAngle = \arctan\left(\frac{trapezoidHeight}{TrapezoidBaseExpansion}\right) \quad (B.7)$$

$$trapezoidSurface = trapezoidHeight \cdot \frac{trapezoidBase + trapezoidRoof}{2} \quad (B.8)$$

$$hexagonSurface = 2 \cdot trapezoidSurface \quad (B.9)$$

$$trapezoidsBaseDifference = bigTrapezoidBase - smallTrapezoidBase \quad (B.10)$$

$$trapezoidsHeightAngle = \arctan\left(\frac{bigTrapezoidHeight - smallTrapezoidHeight}{detectorWidth}\right) \quad (B.11)$$

$$trapezoidsSideAngle = \arctan\left(\frac{\sin(trapezoidSharpAngle) \cdot \frac{trapezoidsBaseDifference}{2}}{detectorWidth}\right) \quad (B.12)$$

### Abdeckungbezogene Größen:

Pentagon:

$$extraTeflonIncircleRadius = \frac{teflonLayer}{\sin(pentagonsIncircleAngle)} \quad (B.13)$$

$$extraAluminiumIncircleRadius = \frac{aluminiumLayer}{\sin(pentagonsIncircleAngle)} \quad (B.14)$$

$$extraLayersIncircleRadius = extraTeflonIncircleRadius + extraAluminiumIncircleRadius \quad (B.15)$$

$$extraAirIncircleRadius = \frac{airLayer}{\sin(pentagonsIncircleAngle)} \quad (B.16)$$

$$extraIncircleRadius = extraLayersIncircleRadius + extraAirIncircleRadius \quad (B.17)$$

$$extraTeflonCircumscribedRadius = extraTeflonIncircleRadius \cdot \cos(36^\circ) \quad (B.18)$$

$$extraAluminiumCircumscribedRadius = extraAluminiumIncircleRadius \cdot \cos(36^\circ) \quad (B.19)$$

$$extraLayersCircumscribedRadius = extraTeflonCircumscribedRadius + extraAluminiumCircumscribedRadius \quad (B.20)$$

$$extraAirCircumscribedRadius = extraAirIncircleRadius \cdot \cos(36^\circ) \quad (B.21)$$

$$extraCircumscribedRadius = extraLayersCircumscribedRadius + extraAirCircumscribedRadius \quad (B.22)$$

$$pentagonSurfaceWithSpace = \frac{5}{2} \cdot (pentagonCircumscribedRadius + extraCircumscribedRadius)^2 \cdot \sin(72^\circ) \quad (B.23)$$

Hexagon:

$$extraTeflonHeight = \frac{teflonLayer}{\sin(trapezoidsHeightAngle)} \quad (B.24)$$

$$extraAluminiumHeight = \frac{aluminiumLayer}{\sin(trapezoidsHeightAngle)} \quad (B.25)$$

$$extraLayersHeight = extraTeflonHeight + extraAluminiumHeight \quad (B.26)$$

$$extraAirHeight = \frac{airLayer}{\sin(trapezoidsHeightAngle)} \quad (B.27)$$

$$extraHeight = extraLayersHeight + extraAirHeight \quad (B.28)$$

$$extraTeflonSideWidth = \frac{teflonLayer}{\sin(trapezoidsSideAngle)} \quad (B.29)$$

$$extraAluminiumSideWidth = \frac{aluminiumLayer}{\sin(trapezoidsSideAngle)} \quad (B.30)$$

$$extraLayersSideWidth = extraTeflonSideWidth + extraAluminiumSideWidth \quad (B.31)$$

$$extraAirSideWidth = \frac{airLayer}{\sin(trapezoidsSideAngle)} \quad (B.32)$$

$$extraSideWidth = extraLayersSideWidth + extraAirSideWidth \quad (B.33)$$

$$extraTeflonBase = \frac{extraTeflonSideWidth}{\sin(trapezoidSharpAngle)} \quad (B.34)$$

$$extraAluminiumBase = \frac{extraAluminiumSideWidth}{\sin(trapezoidSharpAngle)} \quad (B.35)$$

$$extraLayersBase = extraTeflonBase + extraAluminiumBase \quad (B.36)$$

$$extraAirBase = \frac{extraAirSideWidth}{\sin(trapezoidSharpAngle)} \quad (B.37)$$

$$extraBase = extraLayersBase + extraAirBase \quad (B.38)$$

$$extraTeflonRoof = \frac{trapezoidBaseExpansion + extraTeflonBase \cdot trapezoidHeight + extraTeflonHeight}{\tan(trapezoidSharpAngle)} \quad (B.39)$$

$$extraAluminiumRoof = \frac{trapezoidBaseExpansion + extraAluminiumBase \cdot trapezoidHeight + extraAluminiumHeight}{\tan(trapezoidSharpAngle)} \quad (B.40)$$

$$extraLayersRoof = extraTeflonRoof + extraAluminiumRoof \quad (B.41)$$

$$extraAirRoof = trapezoidBaseExpansion + extraAirBase - \frac{trapezoidHeight + extraAirHeight}{\tan(trapezoidSharpAngle)} \quad (B.42)$$

$$extraRoof = extraLayersRoof + extraAirRoof \quad (B.43)$$

$$trapezoidSurfaceWithSpace = \frac{(trapezoidHeight + extraHeight) \cdot (trapezoidBase + trapezoidRoof + 4 \cdot extraBase)}{2} \quad (B.44)$$

$$hexagonSurfaceWithSpace = 2 * trapezoidSurfaceWithSpace; \quad (B.45)$$

### Schlitzbezogene Größen:

$$pentagonIncircleCut = \frac{\frac{slitLong}{2}}{\sin(90^\circ - pentagonsIncircleAngle)} - extraAirIncircleRadius \quad (B.46)$$

$$pentagonCircumscribedCut = \frac{pentagonIncircleCut}{\sin(54^\circ)} \quad (B.47)$$

$$trapezoidHeightCut = \frac{\frac{slitLong}{2}}{\sin(90^\circ - trapezoidsHeightAngle)} - extraAirHeight \quad (B.48)$$

$$trapezoidSideWidthCut = \frac{\frac{slitLong}{2}}{\sin(90^\circ - trapezoidsSideAngle)} - extraAirSideWidth \quad (B.49)$$

$$trapezoidBaseCut = \frac{trapezoidSideWidthCut}{\sin(trapezoidSharpAngle)} \quad (B.50)$$

### Aufbaubezogene Größen:

$$totalSmallSurface = nPentagon \cdot smallPentagonSurface + nHexagon \cdot smallHexagonSurface \quad (B.51)$$

$$totalSmallSurfaceWithSpace = nPentagon \cdot smallPentagonSurfaceWithSpace + nHexagon \cdot smallHexagonSurfaceWithSpace \quad (B.52)$$

$$extraRadius = innerDetectorRadius \cdot \sqrt{\frac{totalSmallSurfaceWithSpace}{totalSmallSurface}} - innerDetectorRadius \quad (B.53)$$

$$patternTheta = pentagonsIncircleAngle + trapezoidsHeightAngle \quad (B.54)$$

wobei nPentagon und nHexagon 12 bzw. 30 ist.