

# Zwei- und dreidimensionale Modelle von Wurzelsystemen

Masterarbeit  
von  
Janina Zurstraßen



JOHANNES GUTENBERG  
UNIVERSITÄT MAINZ

Institut für Mathematik  
Fachbereich 08  
an der Johannes Gutenberg-Universität Mainz

Betreuer: Juniorprofessor Dr. rer. nat. Nikita Semenov  
Abgabedatum: 31.03.2015

# Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitung und Konstruktionen von <math>E_8</math></b>	<b>6</b>
1.1	Vorbereitung . . . . .	6
1.2	$E_8$ . . . . .	8
1.3	Konstruktion von $E_8$ mit Hilfe des Hamming Codes . . . . .	9
1.4	Konstruktion von $E_8$ mit Hilfe von ganzen Oktonionen . . . . .	14
1.5	Konstruktion von $E_8$ mit Hilfe von Del-Pezzo-Flächen . . . . .	15
<b>2</b>	<b>Die Coxeter-Ebene</b>	<b>21</b>
2.1	Vorbereitung . . . . .	21
2.2	Konstruktion . . . . .	22
2.3	Eigenschaften . . . . .	24
2.4	Folgerung . . . . .	26
<b>3</b>	<b>Platonische Polychora</b>	<b>33</b>
3.1	Vorbereitung . . . . .	33
3.2	Die regelmäßigen Polychora . . . . .	37
3.2.1	Der 5-Zeller, auch 4-Simplex oder Hypertetraeder . . . . .	37
3.2.2	Der 8-Zeller, auch 4-Maßpolytop oder Hyperwürfel . . . . .	37
3.2.3	Der 16-Zeller, auch 4-Kreuzpolytop . . . . .	38
3.2.4	Der 24-Zeller . . . . .	38
3.2.5	Der 600-Zeller . . . . .	39
3.2.6	Der 120-Zeller . . . . .	39
3.3	Reguläre Polytope in $n$ Dimensionen . . . . .	40
<b>4</b>	<b>Dreidimensionale Bilder von <math>E_8</math></b>	<b>45</b>
4.1	Vorbereitung . . . . .	45
4.2	Die Ikosiane . . . . .	47
4.3	Die Coxeter Gruppen $H_4$ und $H_3$ . . . . .	49
4.4	Eine dreidimensionale Projektion von $E_8$ nach D. Richter . . . . .	52
4.4.1	Das Bild . . . . .	52
4.4.2	Symmetrie . . . . .	53
4.5	Eine dreidimensionales Projektion von $E_8$ , nach G. Moxness . . . . .	54

- A Programm zur Projektion der Wurzeln eines Wurzelsystems  
in seine Coxeter Ebene 60
- B Programm zum dreidimensionalen Modell von  $E_8$  nach D.Richter 65
- C Programm zum dreidimensionalen Modell von  $E_8$  nach G.Moxness 71

# Einleitung

Wurzelsysteme sind Teilmengen des euklidischen Raumes mit speziellen geometrischen Eigenschaften. Die irreduziblen Wurzelsysteme klassifizieren unter anderem die eng miteinander verbundenen einfachen Lie-Gruppen und halbeinfachen Lie-Algebren über algebraisch abgeschlossenen Körpern. Die Lie-Gruppe  $E_8$  findet viele Anwendungen, beispielsweise in der experimentellen Physik. Um 1990 studierten der Physiker Alexander Zamolodchikov und andere ein theoretisches Modell, das einen eindimensionalen Magnet, der zwei magnetischen Feldern ausgesetzt ist beschreibt. Dieses Modell machte einige numerischen Vorhersagen, die nun in einem Experiment getestet wurden. Die Ergebnisse waren wie vorhergesagt. Das Modell enthielt  $E_8$ , daher kann man sagen, dass das Experiment Hinweise auf  $E_8$ -Symmetrien liefert [21]. Des Weiteren behauptete Dr. A. Garrett Lisi in seinem pre-print “An Exceptionally Simple Theory of Everything“ [20], dass man alle vier Grundkräfte der Physik (Gravitation, Elektromagnetismus, schwache Wechselwirkung und starke Wechselwirkung) mit Hilfe der Lie-Algebra  $E_8$  vereinen kann [19]. Dies führte zu vielen Diskussionen und wurde von Jacques Distler und Skip Garibaldi in “There is no “Theory of Everything“ inside  $E_8$ “ [22] widerlegt. In diesem Artikel wird gezeigt, dass keine auf der  $E_8$ -Symmetriegruppe basierte Theorie eine “Theory of Everything“ sein kann. Seither gibt es viele Diskussionen rund um dieses Thema.

In dieser Arbeit sollen Projektionen von Wurzelsystemen, insbesondere von  $E_8$  erstellt werden, die die Symmetrie derselben sichtbar machen. Die ganze Symmetrie kann für Wurzelsysteme aus dem  $\mathbb{R}^n$  auch nur im  $n$ -dimensionalen aufgezeigt werden. Daher ist es wichtig die Räume, in die projiziert wird gut zu wählen um möglichst viel der Symmetrie zu erhalten.

In Kapitel 1 wird eine Einführung in Wurzelsysteme und die Theorie dahinter gegeben und  $E_8$ , auf das sich in der Arbeit hauptsächlich konzentriert wird, auf drei verschiedene Weisen konstruiert werden. Zunächst mittels Kodierungstheorie über den Hamming-Code, dann über ganze Oktonionen und schließlich über Del-Pezzo-Flächen.

In Kapitel 2 wird dann die sogenannte Coxeter-Ebene konstruiert und erklärt, warum sich diese Projektionsebene für zweidimensionale Modelle von Wurzelsystemen so gut eignet. Außerdem wird hieraus auch direkt eine einfachere Konstruktion dieser Ebene abgeleitet. Ein Maple-Programm, das diese

Projektion realisiert findet sich im Anhang wieder.

In Kapitel 3 geht es dann um reguläre Polytope, insbesondere um welche der Dimension vier, da in Kapitel 4 dann das Wurzelsystem  $E_8$  mit einem dieser Polytope identifiziert wird. Zunächst lernen wir eine Notation für  $n$ -dimensionale Polytope, das Schläfli-Symbol, kennen und geben eine Bedingung für die Existenz eines regulären Polytopes der Dimension vier an. Dann konstruieren wir diese. Schließlich geben wir noch eine Bedingung für die Existenz eines regulären  $n$ -dimensionalen Polytops an.

Im letzten Kapitel 4 wird dann zunächst die Verbindung zwischen dem Wurzelsystem  $E_8$ , dem 4-Polytop 600-Zeller und den sogenannten Ikosianen aufgezeigt. Schließlich benutzen wir dies alles um zwei verschiedene dreidimensionale Modelle von  $E_8$  mit Hilfe von Maple zu zeichnen und treffen einige Aussagen über deren Symmetrie.

# Kapitel 1

## Vorbereitung und Konstruktionen von $E_8$

In diesem Kapitel werden wir zunächst eine kurze Zusammenfassung über Wurzelsysteme und deren Weyl- und Coxeter Gruppen geben. Danach werden wir drei mögliche Konstruktionen des exzeptionellen Wurzelsystems  $E_8$  angeben.

### 1.1 Vorbereitung

Wir gehen nach [1] vor. Es sei  $V$  ein endlicher euklidischer Vektorraum und  $(\cdot | \cdot)$  ein Skalarprodukt auf  $V$ .

**Definiton 1.1.1.** Eine endliche Teilmenge  $R \subset V$  mit  $0 \notin R$  heißt *Wurzelsystem*, falls folgende Bedingungen erfüllt sind:

1.  $R \cap \mathbb{R}\alpha = \{\alpha, -\alpha\}$  für alle  $\alpha \in R$ ,
2. Für alle  $\alpha \in R$  erhält die Spiegelung  $s_\alpha(x) := x - 2\frac{(x|\alpha)}{(\alpha|\alpha)}\alpha$   $R$ , also  $s_\alpha(R) = R$ .

Die Elemente in  $R$  heißen *Wurzeln* und das Wurzelsystem heißt *kristallographisch*, falls zusätzlich gilt:

$$2\frac{(\alpha|\beta)}{(\beta|\beta)} \in \mathbb{Z} \text{ für alle } \alpha, \beta \in R.$$

Eine Teilmenge  $S = \{\alpha_1, \dots, \alpha_n\} \subset R$  heißt *Basis* von  $R$ , falls  $S$  eine (Vektorraum-) Basis des  $\mathbb{R}$ - Spans von  $R$  in  $V$  ist und alle  $\beta \in R$  als nicht-positive oder nichtnegative Linearkombination der Elemente aus  $S$  (*einfache Wurzeln*) dargestellt werden können. Für den Beweis, dass eine solche Basis existiert siehe [1], Seite 8. Der *Rang* des Wurzelsystems ist  $n$ .

Für ein kristallographische Wurzelsystem kann man jede Wurzel sogar als Linearkombination mit Koeffizienten aus  $\mathbb{Z}$  (alle mit dem gleichen Vorzeichen) ausdrücken. Die Matrix  $A$  mit Einträgen  $a_{ij} = 2 \frac{(\alpha_i | \alpha_j)}{(\alpha_j | \alpha_j)}$  heißt *Cartan Matrix*. Nimmt man den eventuellen Verlust der Kristallographiebedingung in Kauf, so kann man jedes Wurzelsystem so umskalieren, dass alle Wurzeln Einheitslänge haben. Als nächstes definieren wir die Coxeter Gruppe:

**Definiton 1.1.2.** Es sei  $W$  die Gruppe, die von den Spiegelungen an Hyperebenen orthogonal zu Wurzeln  $\alpha$  aus  $R$  erzeugt wird:

$$W := \langle s_\alpha \mid \alpha \in R \rangle.$$

In [1] wird in Abschnitt 1.5 gezeigt, dass  $W$  von den *einfachen Spiegelungen*  $S'$ , also Spiegelungen bezüglich einfachen Wurzeln, erzeugt werden kann. Wir definieren  $m(\alpha_i, \alpha_j) = m_{ij}$  für  $1 \leq i, j \leq n$  als die Ordnung von  $s_i s_j$  in  $W$  (schreibe  $s_i$  statt  $s_{\alpha_i}$ ). In Abschnitt 1.9 von [1] wird gezeigt, dass

$$W = \langle s_1, s_2, \dots \in S' \mid (s_i s_j)^{m_{ij}} = 1, 1 \leq i, j \leq n \rangle$$

eine Präsentation von  $W$  ist.

Eine Gruppe, die eine solche Präsentation besitzt, nennen wir *Coxeter Gruppe*. Es muss gefordert werden, dass  $m_{ii} = 1$  ist, aber eine Relation  $(s_i s_j)^{m_{ij}} = 1$  kann wegfallen um die Möglichkeit zu lassen, dass das Produkt unendliche Ordnung hat. Die Matrix  $M := m_{ij}$  heißt *Coxeter Matrix* und  $(W, S)$  heißt Coxeter-System.

Ist  $R$  kristallographisch, so heißt die von den Spiegelungen  $s_\alpha$  mit  $\alpha \in R$  erzeugte Gruppe die *Weyl Gruppe* von  $R$ . Weyl Gruppen sind ein Beispiel für Coxeter Gruppen.

Die Coxeter Gruppe  $W$  ist bis auf Isomorphie durch die Menge der Zahlen  $m_{ij}$  für  $\alpha_i, \alpha_j \in S$  bestimmt. Wir codieren diese Information in dem sogenannten Coxeter-Dynkin Diagramm:

**Definiton 1.1.3.** Ein *Coxeter-Dynkin Diagramm* ist ein endlicher, ungerichteter Graph, dessen Kanten mit Zahlen  $\geq 3$  oder dem Symbol  $\infty$  beschriftet sind und dessen Knotenmenge  $K$  in einer 1-1-Korrespondenz mit den einfachen Spiegelungen  $S'$  steht. Die Kante, die  $s$  und  $s'$  aus  $K$  mit  $s \neq s'$  verbindet, wird mit  $m(s, s')$  beschriftet. Ist  $m(s, s') = 2$ , so werden die Knoten  $s$  und  $s'$  nicht verbunden und ist  $m(s, s') = 3$ , so wird die Kante nicht beschriftet.

Ist das Coxeter Diagramm eines Coxeter-Systems zusammenhängend, so nennen wir es irreduzibel. Das zugehörige Wurzelsystem heißt dann ebenfalls irreduzibel.

**Definiton 1.1.4.** Ist  $R$  ein irreduzibles, kristallographisches Wurzelsystem, so gibt es nur 2 verschiedene Längen der Wurzeln. Diese Information fügt man im Coxeter-Dynkin Diagramm ein, indem man an die Kante, die zwei unterschiedlich lange Wurzeln verbindet, einen Pfeil in Richtung der kurzen Wurzel setzt. Ersetzt man dann noch Kanten, die mit 4 oder 6 beschriftet sind durch doppelte oder dreifache Kanten ohne Beschriftung, so erhält man ein Diagramm namens *Dynkin Diagramm*.

## 1.2 $E_8$

Die Lie-Algebra  $E_8$  hat Dimension 248. Die vorliegende Arbeit beschäftigt sich hauptsächlich mit dem Wurzelsystem  $E_8$ . Dieses hat Rang 8 und besteht aus 240 Wurzeln. Diese normiert man so, dass sie die Länge  $\sqrt{2}$  haben. Man nimmt die Menge aller Vektoren des  $\mathbb{R}^8$  mit dieser Länge, sodass alle Koordinaten in  $\mathbb{Z}$  oder alle in  $\mathbb{Z} + \frac{1}{2}$  liegen und die Summe aller Koordinaten gerade ist. Ist  $e_1, \dots, e_8$  die Standardbasis des  $\mathbb{R}^8$ , so besteht  $E_8$  also aus den Wurzeln

$$\pm e_i \pm e_j \text{ für } i < j \text{ und } \frac{1}{2} \sum_{i=1}^8 \pm e_i \text{ mit gerade Anzahl von } + \text{ Zeichen.}$$

Eine Wahl der einfachen Wurzeln von  $E_8$  ist

$$S := \{e_1 - e_2, e_2 - e_3, e_3 - e_4, e_4 - e_5, e_5 - e_6, e_6 + e_7, \sum_{i=1}^8 -\frac{1}{2}e_i, e_6 - e_7\}.$$

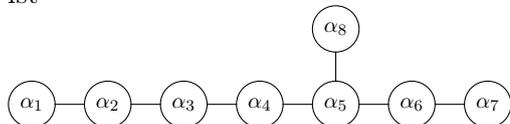
Die Cartan Matrix ist dann

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 2 \end{pmatrix}$$

und die Coxeter Matrix ist

$$M = \begin{pmatrix} 1 & 3 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 1 & 3 & 2 & 2 & 2 & 2 \\ 2 & 2 & 3 & 1 & 3 & 2 & 2 & 2 \\ 2 & 2 & 2 & 3 & 1 & 3 & 2 & 3 \\ 2 & 2 & 2 & 2 & 3 & 1 & 3 & 2 \\ 2 & 2 & 2 & 2 & 2 & 3 & 1 & 2 \\ 2 & 2 & 2 & 2 & 3 & 2 & 2 & 1 \end{pmatrix}.$$

Das zugehörige Dynkin Diagramm (gleich dem Coxeter-Dynkin Diagramm) ist



Die Weyl Gruppe von  $E_8$  hat Ordnung 696729600.

Es gibt einige Möglichkeiten  $E_8$  zu konstruieren. Im Folgenden möchte ich zwei davon aufzeigen:

### 1.3 Konstruktion von $E_8$ mit Hilfe des Hamming Codes

$E_8$  kann mittels Kodierungstheorie über den Hamming Code konstruiert werden. Man kann aus einem Code ein Gitter erhalten, wenn man in das Gitter alle diejenigen Vektoren  $(v_1, \dots, v_n)$  aufnimmt, die modulo 2 ein Codewort des Codes ergeben. Zunächst aber etwas über die Hintergründe und eine kleine Einführung in Codetheorie. Wir folgen in diesem Abschnitt [6] Kapitel 3, §2, beschränken uns hier allerdings auf binäre Codes.

Ein *binärer Code*  $C$  der Länge  $n$  ist ein Untervektorraum von  $\mathbb{F}_2^n$ . Ein *binäres Codewort* ist also ein  $n$ -Tupel aus 1 und 0,

$$u = (u_1, \dots, u_n) \text{ mit } u_i \in \mathbb{F}_2 = \{0, 1\}.$$

Natürlich kann man statt  $\mathbb{F}_2^n$  genauso  $\mathbb{F}_q^n$  nehmen, wobei  $q$  eine Primzahl oder eine Primzahlpotenz ist.

Passiert nun beim Übermitteln des Codewortes ein Fehler (mit Wahrscheinlichkeit  $p < 1/2$ ), wird also eine 1 gesendet aber eine 0 empfangen oder umgekehrt, so ist es gut wenn das fehlerhafte Codewort kein Wort des Codes ist, und der Empfänger so erkennt, dass überhaupt ein Fehler passiert ist. Der Empfänger kann dann das ursprüngliche Signal rekonstruieren, indem er das Codewort findet, das am nächsten am Empfangenen liegt, sprich, welches sich an den wenigsten Stellen vom Empfangenen unterscheidet. War die Störung aber groß und es sind mehrere Fehler passiert, wird so nicht unbedingt das richtige Wort rekonstruiert. Wir möchten in unserem Code also am besten Wörter haben, die relativ weit voneinander entfernt sind, um sie leichter richtig übersetzen zu können. Wir definieren dafür den *Hamming Abstand* zwischen zwei Vektoren  $u$  und  $v$  als die Anzahl der Stellen, in denen sie sich unterscheiden:

$$d(u, v) := |\{i : u_i \neq v_i\}|$$

Das *Hamming-Gewicht* von  $u$   $\text{wt}(u)$  wird definiert als die Anzahl der Koordinaten von  $u$ , die ungleich Null sind. Damit gilt

$$d(u, v) = \text{wt}(u - v).$$

Der *kleinste Abstand*  $d$  eines Codes ist also:

$$d = \min\{d(u, v) \mid u, v \in C, u \neq v\}.$$

Der Empfänger kann in dem Code  $C$  somit

$$\rho = \lfloor 1/2(d-1) \rfloor$$

Fehler richtig korrigieren. Weiter definieren wir  $A_i(u)$  als die Anzahl der Codewörter, die den Hamming Abstand  $i$  von  $u$  haben. Die Menge  $\{A_i(u)\}$  nennt man die *Gewichtsverteilung* von  $C$  bezüglich  $u$ . Ein Code der Länge  $n$ , der  $M$  Codewörter enthält, die den kleinsten Abstand  $d$  voneinander haben, wird  $(n, M, d)$  Code genannt.

Ist unser Code nun ein linearer Untervektorraum von  $\mathbb{F}_2^n$ , ist die Menge der Codewörter also unter Vektoraddition und Skalarmultiplikation mit Elementen aus  $\mathbb{F}_2$  abgeschlossen, so sprechen wir von einem linearen Code. Die *Dimension  $k$  des Codes* ist dann als die Dimension dieses Untervektorraumes definiert und es existieren  $q^k$  Codewörter. Für den kleinsten Abstand eines linearen Codes gilt:

$$d = \min\{\text{wt}(u) \mid u \in C, u \neq 0\}.$$

Für lineare Codes ist  $A_i(u)$  unabhängig von  $u$  und wird nur mit  $A_i$  bezeichnet. Lineare Codes der Länge  $n$  mit Dimension  $k$  und kleinstem Abstand  $d$  werden  $[n, k, d]$  oder nur  $[n, k]$  Codes genannt.

**Definiton 1.3.1.** Ein Code  $C$  ist *zyklisch*, falls gilt: Ist  $c_0c_1 \dots c_{n-1}$  ein Codewort, so auch  $c_{n-1}c_0c_1 \dots c_{n-2}$ .

Es sei nun  $C$  ein  $[n, k, d]$ , zyklischer, linearer Code über  $\mathbb{F}_2$ . Man schreibt ein Codewort  $c = c_0 \dots c_{n-1}$  auch als Polynom  $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$  im Ring  $R_n(x) = \mathbb{F}_2[x]/x^n - 1$ . Dann beschreibt  $xc(x)$  einen zyklischen Shift von  $c$  und ein zyklischer Code ist ein Ideal in  $R_n(x)$ . Nach [10], S.190ff kann dieses Ideal von einem einzigen Polynom  $g(x) = g_0 + g_1x + \dots + g_{n-1}x^{n-1}$  erzeugt werden. Hieraus kann man eine *erzeugende Matrix*  $G$  erstellen:

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_{n-1} & 0 & \dots & \dots & 0 \\ 0 & g_0 & g_1 & g_2 & \dots & g_{n-1} & 0 & \dots & 0 \\ & & & & \dots & & & & \\ 0 & \dots & \dots & 0 & g_0 & g_1 & g_2 & \dots & g_{n-1} \end{pmatrix},$$

bei der die Zeilen  $g(x)$ ,  $xg(x)$  usw. entsprechen. Der Code besteht dann aus  $\mathbb{F}_2$ -Linearkombinationen der Zeilen von  $G$ . Nun möchten wir noch definieren, wann zwei Codes äquivalent sind (nach [11], S.50ff). Dafür benötigen wir zwei Arten von Permutationen:

**Definiton 1.3.2.** Es sei  $\sigma$  eine Permutation von  $(1, 2, \dots, n)$ . Eine *Positionsp permutation* sei eine Transformation, die wir auf jedes Codewort von  $C$  anwenden:

$$u \mapsto u'$$

mit

$$u'_i = u_{\sigma(i)}.$$

Ist  $\sigma$  eine Permutation der Symbole, das heißt in unserem Fall eine Permutation von  $\{0, 1\}$ , so ist eine *Symbolpermutation* für ein  $i$ ,  $1 \leq i \leq n$  und für jedes Codewort  $u \in C$  die Transformation

$$u \mapsto u'$$

mit

$$u'_j = u_j \text{ für } 1 \leq j \leq n, i \neq j \text{ und } u'_i = \sigma(u_i).$$

**Definiton 1.3.3.** Zwei Codes  $C$  und  $C'$  heißen *äquivalent*, falls  $C'$  durch Hintereinanderausführung von Symbol- oder Positionsp permutationen von  $C$  erhalten werden kann.

Man kann zu jedem Code  $C$  mit erzeugender Matrix  $G$  einen äquivalenten Code  $C'$  finden, dessen erzeugende Matrix die Form

$$G' = [\mathbb{I}_k \mid A]$$

hat.  $\mathbb{I}_k$  ist hierbei die  $k \times k$  Einheitsmatrix und  $A$  somit eine  $k \times (n - k)$  Matrix. Bei der Überführung von  $G$  nach  $G'$  sind die folgenden Transformationen erlaubt:

1. Permutieren der Zeilen
2. Permutieren der Spalten
3. Multiplizieren einer Reihe mit einem Skalar ungleich 0
4. Multiplizieren einer Spalte mit einem Skalar ungleich 0
5. Addieren eines skalaren Vielfachen einer Zeile zu einer anderen Zeile.

Wir führen nun den Hamming Code  $\mathcal{H}_7$  ein, um aus ihm dann gleich den erweiterten Hamming Code  $\mathcal{H}_8$  herzuleiten.

**Beispiel 1.3.4.** Der Hamming Code  $\mathcal{H}_7$  ist ein binärer, zyklischer, linearer  $[7, 4, 3]$  Code der Länge 7 und hat das erzeugende Polynom

$$g(x) = 1 + x + x^3.$$

Die erzeugende Matrix ist also

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Der Code besteht somit aus folgenden Wörtern:

(0000000)  
 (1101000)  
 (0110100)  
 (0011010)  
 (0001101)  
 (1011100)  
 (1110010)  
 (1100101)  
 (0101110)  
 (0111001)  
 (0010111)  
 (1000110)  
 (1010001)  
 (1111111)  
 (0100011)  
 (1001011)

Ein äquivalenter Code wird von der Matrix

$$G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

erzeugt. Die Gewichtsverteilung dieses Codes ist

$$A_0 = 1, A_3 = A_4 = 7, A_7 = 1.$$

Hieraus bilden wir nun den erweiterten Hamming Code  $\mathcal{H}_8$ :

**Beispiel 1.3.5.** Wir fügen an  $\mathcal{H}_7$  eine Prüfziffer an, die angibt, ob die Anzahl der 1en im Codewort gerade oder ungerade ist. Damit erhalten wir den erweiterten Hamming Code  $\mathcal{H}_8$ , in dem jedes Codewort eine gerade Anzahl

an 1en hat. Dieser Code ist ein  $[8, 4, 4]$  Code und eine erzeugende Matrix ist:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Eine andere erzeugende Matrix ist

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

$\mathcal{H}_8$  hat die Gewichtsverteilung

$$A_0 = 1, A_4 = 14, A_8 = 1.$$

Nach [6], S.137 kann man ein Gitter aus einem  $(n, M, d)$  Code  $C$  folgendermaßen konstruieren:

$$\Lambda = \frac{1}{\sqrt{2}} \{x \in \mathbb{Z}^n \mid x \bmod 2 \in C\}.$$

Hiebei gilt die folgende Definition

**Definiton 1.3.6.** Für einen Punkt  $x = (x_1, \dots, x_n)$  im  $\mathbb{R}^n$  sei

$$x \bmod 2 = (x_1 \bmod 2, \dots, x_n \bmod 2).$$

So kann man das Wurzelgitter  $E_8$  aus dem erweiterten Hamming Code  $\mathcal{H}_8$  konstruieren. In dem folgenden Beispiel möchte ich zeigen, wie viele Vektoren mit Norm  $\sqrt{2}$  man erhält.

**Beispiel 1.3.7.**  $\mathcal{H}_8$  besteht aus einem Wort mit Gewicht 0, aus 14 Wörtern mit Gewicht 4 und aus einem Wort mit Gewicht 8.

Die Vektoren mit dem kleinsten Betrag (außer dem Nullvektor), die modulo 2 gleich dem Wort  $(00000000) \in \mathcal{H}_8$  sind, haben die Form  $(\pm 2, 0, 0, 0, 0, 0, 0, 0)$  und alle Permutationen davon, also insgesamt 16 Stück. Multipliziert man diese mit  $\frac{1}{\sqrt{2}}$  wie in der Konstruktion oben, so erhält man 16 Vektoren mit Norm  $\sqrt{2}$ .

Die kleinsten Vektoren, die modulo 2 gleich einem Wort mit Gewicht 4, also nach Umordnung gleich einem Wort  $(11110000)$  sind, haben nach Multiplikation mit  $\frac{1}{\sqrt{2}}$  die Form  $\frac{1}{\sqrt{2}}(\pm 1, \pm 1, \pm 1, \pm 1, 0, 0, 0, 0)$ . Bei freier Vorzeichenwahl und 14 Permutationen, macht das insgesamt  $16 \cdot 14 = 224$  weitere Vektoren mit Norm  $\sqrt{2}$ .

Betrachten wir schließlich das Wort mit dem Gewicht 4, also  $(11111111)$ , so fällt auf, dass die zugehörigen Vektoren im  $\mathbb{Z}^n$  mindestens Norm 2 haben und somit nicht gezählt werden.

Also haben wir insgesamt  $16 + 224 = 240$  Vektoren mit Norm  $\sqrt{2}$ .

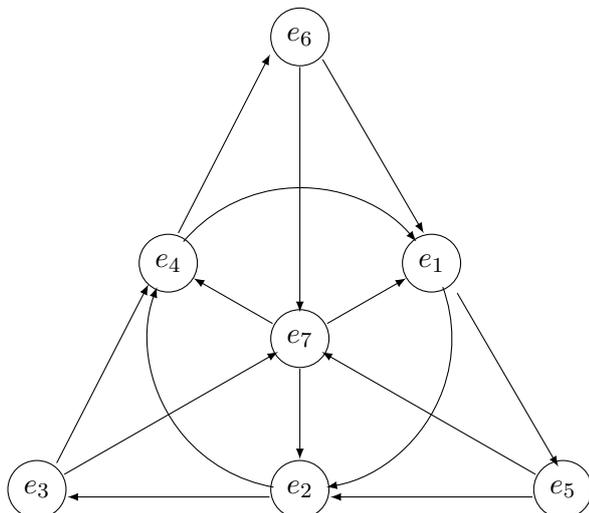
## 1.4 Konstruktion von $E_8$ mit Hilfe von ganzen Oktonionen

Ebenso kann das Wurzelgitter  $E_8$  aus den ganzen Oktonionen konstruiert werden. Hierfür müssen wir natürlich erst einmal klären, wann ein Oktonion als ganz bezeichnet wird. In diesem Kapitel folgen wir [12], S.99ff und [13].

**Definiton 1.4.1.** Die *Oktonionen* sind als Vektorraum

$$\mathbb{O} = \{a = a_\infty + \sum_{i=1}^7 a_i e_i \mid a_\infty, a_1, \dots, a_7 \in \mathbb{R}\}.$$

Sie bilden eine nichtassoziative Algebra mit neutralem Element 1, wenn die  $e_i$  einige Gleichungen erfüllen, die man an dem folgenden Diagramm, der Fano Ebene, ablesen kann:



Liegen  $e_i, e_j, e_k$  auf einer Linie (wobei der Kreis auch als Linie zählt), und sind so geordnet, wie die Pfeile es angeben, dann müssen  $e_i, e_j, e_k$  die folgenden Gleichungen und alle zyklischen Permutationen davon erfüllen:

$$e_i^2 = e_j^2 = e_k^2 = -1$$

$$e_i e_j = e_k = -e_j e_i$$

Ein Konjugiertes von  $a = a_\infty + \sum_{i=1}^7 a_i e_i$  sei  $\bar{a} = a_\infty - \sum_{i=1}^7 a_i e_i$ . Wir definieren eine Norm auf  $\mathbb{O}$  via

$$|a|^2 = a\bar{a} = a_\infty^2 + \sum_{i=1}^7 a_i^2.$$

Die Oktonionen, bei denen alle Koeffizienten ganze Zahlen sind, heißen *Graves'sche Oktonionen*.

Um aber auf die ganzen Oktonionen zu kommen muss noch etwas mehr getan werden. Wir bilden sogenannte Halbierungsmengen von Indizes von  $a$ .

**Definiton 1.4.2.** Wir betrachten ein Oktonion  $a$ , dessen Koordinaten in  $\frac{1}{2}\mathbb{Z}$  liegen. Wir nennen die Menge der Koordinaten, die nicht auch in  $\mathbb{Z}$  liegen, *Halbierungsmenge* der Indizes für  $a$ .

Weiter betrachten wir die Halbierungsmengen  $\{\infty, i, j, k\}$ , wo  $e_i, e_j, e_k$  auf einer Linie in der Fano Ebene liegen, sowie die Halbierungsmengen  $\{l, m, n, o\}$ , wo  $e_l, e_m, e_n, e_o$  nicht auf dieser Linie liegen. Diese Mengen, zusammen mit der leeren Menge und  $\{\infty, 1, 2, 3, 4, 5, 6, 7\}$  bilden die  $\infty$ -Mengen. Diese sind also die Mengen

$$\begin{array}{cccccccc} \emptyset & \infty 124 & \infty 235 & \infty 346 & \infty 156 & \infty 137 & \infty 457 & \infty 267 \\ \infty 1234567 & 3567 & 1467 & 1257 & 2347 & 2456 & 1236 & 1345 \end{array}$$

Natürlich entsprechen die Oktonionen mit Halbierungsmenge  $\emptyset$  den Graves'schen Oktonionen. Die Oktonionen, deren Halbierungsmenge eine  $\infty$ -Menge ist, nennen wir  $\infty$ -Ganzzahl. Die  $\infty$ -Ganzzahlen sind multiplikativ jedoch nicht abgeschlossen. Ein kurzes Beispiel belegt das:

**Beispiel 1.4.3.**

$$\left(\frac{1}{2} + \frac{1}{2}e_1 + \frac{1}{2}e_3 + \frac{1}{2}e_7\right) \cdot \left(\frac{1}{2} + \frac{1}{2}e_1 + \frac{1}{2}e_2 + \frac{1}{2}e_4\right) = \left(\frac{1}{2}e_1 + \frac{1}{2}e_3 + \frac{1}{2}e_4 + \frac{1}{2}e_6\right)$$

Vertauscht man nun in der Definition von  $\infty$ -Mengen und  $\infty$ -Ganzzahlen  $\infty$  mit 1, erhält man die 1-Mengen und 1-Ganzzahlen. Diese sind erstaunlicherweise multiplikativ abgeschlossen (siehe [12], S.103). Die 1-Mengen sind

$$\begin{array}{cccccccc} \emptyset & \infty 124 & 1235 & 1346 & \infty 156 & \infty 137 & 1457 & 1267 \\ \infty 1234567 & 3567 & \infty 467 & \infty 257 & 2347 & 2456 & \infty 236 & \infty 345. \end{array}$$

Die *ganzen Oktonionen* seien nun die 1-Ganzzahlen. Diese bilden ein  $E_8$ -Wurzelgitter. Hier haben die Wurzeln allerdings alle Länge 1. Wie schon bei der Konstruktion von  $E_8$  aus dem Hamming Code, möchte ich auch hier zur Überprüfung die Anzahl der ganzen Oktonionen mit minimaler Norm ausrechnen:

**Beispiel 1.4.4.** Die Graves'schen Oktonionen mit minimaler Norm ungleich Null sind  $\pm 1$  und  $\pm e_i$  für  $i = 1, \dots, 7$ . Das sind 16 Stück der Länge 1. Bleiben noch  $\frac{1}{2}(1 + e_i + e_j + e_k)$  und  $\frac{1}{2}(e_l + e_m + e_n + e_o)$ , wo  $\infty ijk$  und  $lmno$  1-Mengen sind. Diese haben ebenfalls Norm 1 und es sind  $16 \cdot (7 + 7) = 224$  Stück. Damit kommen wir auf  $224 + 16 = 240$  ganze Oktonionen mit minimaler Norm.

## 1.5 Konstruktion von $E_8$ mit Hilfe von Del-Pezzo-Flächen

Eine dritte Möglichkeit und die letzte, die wir hier aufführen möchten, ist der Weg über del Pezzo Flächen und deren Picard-Gruppen. Wir definieren zunächst wie in [14], Seite 114:

**Definiton 1.5.1.** Sei  $r \geq 1$ . Es seien

1.  $N_r := \mathbb{Z}^{r+1} = \bigoplus_{i=0}^r \mathbb{Z}l_i$  mit  $(l_i)$  eine gewählte Basis,
2.  $\omega_r := (-3, 1, \dots, 1) \in N_r$ ,
3.  $(-, -)$  eine Bilinearform  $N_r \times N_r \rightarrow \mathbb{Z}$  mit

$$(l_0, l_0) = 1, \quad (l_i, l_i) = -1 \text{ für } i \geq 1, \quad (l_i, l_j) = 0 \text{ für } i \neq j,$$

4.  $R_r := \{l \in N_r \mid (l, \omega_r) = 0, (l, l) = -2\}$ .

**Definiton 1.5.2.** Eine *Del-Pezzo-Fläche* ist eine glatte projektive, rationale Fläche  $X$ , auf der die antikanonische Klasse  $\Omega_V^{-1}$  ampel ist.

In [14], Theorem 24.4 wird gezeigt, dass jede Del-Pezzo-Fläche vom Grad  $d$  bis auf Isomorphie entweder  $\mathbb{P}^1 \times \mathbb{P}^1$ ,  $\mathbb{P}^2$  oder eine Aufblasung von  $\mathbb{P}^2$  in  $9 - d$  Punkten in allgemeiner Lage ist. Hierbei sind Punkte in allgemeiner Lage, falls keine drei kollinear sind und keine sechs auf einem Kegelschnitt liegen. Sei nun  $V$  eine Del-Pezzo-Fläche.

**Definiton 1.5.3.** Die *Picardgruppe*  $\text{Pic}(X)$  ist die abelsche Gruppe der Isomorphieklassen von Geradenbündeln auf  $X$ , mit dem Tensorprodukt als Gruppenoperation. Weiter sei  $\omega_V$  die Klasse der Bündel von Differentialen  $\Omega_V$ .

Nach Lemma 24.3.1 und Proposition 25.1 aus [14] ist die Picardgruppe einer Del-Pezzo-Fläche  $V$  vom Grad  $d$  eine freie abelsche Gruppe und wenn  $V$  nicht isomorph zu  $\mathbb{P}^1 \times \mathbb{P}^1$  ist, gibt es eine freie Basis  $(l_0, \dots, l_r)$  in  $\text{Pic}(V)$ , mit  $r = 9 - d$ , sodass gilt

1.  $\omega_V = -3l_0 + \sum_{i=1}^r l_i$ .
2.  $(l_0, l_0) = 1, \quad (l_i, l_i) = -1 \text{ für } i \geq 1, \quad (l_i, l_j) = 0 \text{ für } i \neq j$ .

Somit stimmt das Objekt  $\{\text{Pic}(V), \omega_V, \text{Schnittprodukt}\}$  mit dem Objekt  $\{N_r, \omega_r, (-, -)\}$  überein.

Kommen wir nun zum Zusammenhang mit Wurzelsystemen:

**Proposition 1.5.4.** Stattet man das orthogonale Komplement  $\omega_r^\perp$  von  $\omega_r$  in  $\mathbb{R} \otimes N_r$  mit dem Skalarprodukt von  $N_r$  mit umgedrehtem Vorzeichen aus, so ist  $\omega_r^\perp$  für  $r \leq 8$  ein euklidischer Vektorraum der Dimension  $r$ . Die Vektoren aus  $R_r$  in  $\omega_r^\perp$  bilden dann ein Wurzelsystem vom Rang  $r$ .

*Beweis.* (nach [14], S.127)

Zunächst zeigen wir, dass  $\omega_r^\perp$  zusammen mit dem Skalarprodukt von  $N_r$  mit umgedrehten Vorzeichen einen euklidischen Vektorraum bildet. Hierfür zeigen wir, dass das Schnittprodukt selbst, das wir ja mit der Bilinearform

auf  $N_r$  identifiziert haben, auf  $\omega_r^\perp$  negativ definit ist.

Offensichtlich wird  $\mathbb{R} \otimes N_r$  von  $\omega_r, l_1, \dots, l_r$  erzeugt. Ein Vektor habe also die Form  $a\omega_r + \sum_{i=1}^r b_i l_i$ . Zunächst schauen wir uns an, wie ein Vektor aus  $\mathbb{R} \otimes N_r$ , der bezüglich der Bilinearform auf  $N_r$  orthogonal zu  $\omega_r$  ist, aussieht.

$$\begin{aligned} \left( \omega_r, a\omega_r + \sum_{i=1}^r b_i l_i \right) &= \left( \left( -3l_0 + \sum_{i=1}^r l_i \right), \left( -3al_0 + \sum_{i=1}^r (b_i + a) l_i \right) \right) \\ &= 9a \underbrace{(l_0, l_0)}_{=1} + \sum_{i=1}^r (b_i + a) \underbrace{(l_r, l_r)}_{=-1} = (9 - r)a - \sum_{i=1}^r b_i. \end{aligned}$$

Beim ersten Schritt sind alle Summanden mit Faktor  $(l_i, l_j) = 0$  für  $i \neq j$ . Weiter ist also

$$\left( \omega_r, a\omega_r + \sum_{i=1}^r b_i l_i \right) = 0 \Leftrightarrow a = \frac{1}{9 - r} \sum_{i=1}^r b_i.$$

Nun berechnen wir die Länge eines solchen Vektors mit umgedrehtem Vorzeichen und erhalten

$$\begin{aligned} 9a^2 (l_0, l_0) + \sum_{i=1}^r (b_i + a)^2 (l_i, l_i) &= (9 - r)a^2 - 2a \sum_{i=1}^r b_i - \sum_{i=1}^r b_i^2 \\ &= -\frac{1}{9 - r} \left( \sum_{i=1}^r b_i \right)^2 - \sum_{i=1}^r b_i^2. \end{aligned}$$

Nun müssen wir noch zeigen, dass  $R_r = \{l \in N_r \mid (l, \omega_r) = 0, (l, l) = 2\}$  ein Wurzelsystem vom Rang  $r$  ist.  $R_r$  ist endlich, da  $N_r \cap \omega_r^\perp$  ein Gitter in  $\omega_r^\perp$  ist und es daher nur endlich viele Vektoren einer festen Länge enthalten kann. Die  $r$  Vektoren  $l_1 - l_2, \dots, l_1 - l_r, l_0 - l_1 - l_2 - l_3$  sind linear unabhängig, liegen in  $\omega_r^\perp$  und haben Länge 2. Die Spiegelung an einer Hyperebene orthogonal zu  $l \in R_r$  eines  $x \in \omega_r^\perp$  ist

$$s_l(x) = x + (x, l)l.$$

Diese Formel wird aus der gewöhnlichen erhalten, wenn man das Vorzeichen des Skalarproduktes tauscht und  $(l, l) = -2$  benutzt. Es ist  $s_l(R_r) = R_r$ , da  $s_l$  Längen und Orthogonalität zu  $\omega_r$  erhält und  $N_r$  auf sich selbst abbildet. Außerdem ist für alle  $l, m \in R_r$  die Zahl  $2 \frac{(l, m)}{(l, l)} \in \mathbb{Z}$ . Damit ist die Proposition bewiesen.  $\square$

Welche Wurzelsysteme sind dies nun aber? Um diese Frage beantworten zu können benötigen wir noch eine Eigenschaft: Es sei  $Q(R_r) \subset \omega_r^\perp$  die Gruppe, die von  $R_r$  erzeugt wird und  $P(R_r)$  das duale Gitter:

$$P(R_r) = \{l \in \omega_r^\perp \mid (l, m) \in \mathbb{Z} \text{ für alle } m \in Q(R_r)\}.$$

Die Quotientengruppe  $P(R_r)/Q(R_r)$  ist endlich, da  $Q(R) \subset P(R)$ . Die Ordnung der Quotientengruppe wird *Zusammenhangsindex* von  $R_r$  genannt. Nach Proposition 25.3 in [14] ist für alle  $3 \leq r \leq 8$ :  $P(R_r)/Q(R_r) \cong \mathbb{Z}_{9-r}$ . Nun können wir den Zusammenhang mit Wurzelsystemen, insbesondere mit  $E_8$  herstellen:

**Proposition 1.5.5.** Es sei  $R_r$  ein Wurzelsystem mit Rang  $r$  für  $3 \leq r \leq 8$ , wo alle Wurzeln die gleiche Länge haben und  $P(R_r)/Q(R_r) \cong \mathbb{Z}_{9-r}$ . Dann ist  $R_r$  isomorph zu einem der Systeme

$$A_1 \times A_2, A_4, D_5, E_6, E_7, E_8.$$

*Beweis.* Die Tabellen in [2] auf den Seiten 250-275 zeigen, dass Wurzelsysteme der Typen  $B_l, C_l, F_4$  und  $G_2$  die erste Bedingung nicht erfüllen. Die verbleibenden Wurzelsysteme haben folgenden Index:

$$\begin{array}{cccccc} & A_l & D_l & E_6 & E_7 & E_8 \\ \text{Index} & l+1 & 4 & 3 & 2 & 1 \end{array}$$

Ist  $R_r = R_{r_1} \times R_{r_2}$  ein zerlegbares Wurzelsystem vom Zusammenhangsindex  $i$  und  $R_{r_1}$  und  $R_{r_2}$  vom Index  $i_1$  bzw.  $i_2$ , so gilt  $i = i_1 \cdot i_2$  und  $\text{Rang}(R_r) = \text{Rang}(R_{r_1}) + \text{Rang}(R_{r_2})$ .

Sei nun  $r = 8$ . Von den verbleibenden hat nur  $E_8$  den Zusammenhangsindex 1. Daher ist  $R_8$  isomorph zu  $E_8$ .

Für  $r = 4, 5, 6$  wird der Zusammenhangsindex  $9-r$  eine Primzahl. Daher sind die zugehörigen Wurzelsysteme  $R_r$  unzerlegbar und es bleibt nur  $R_7 \cong E_7$ ,  $R_6 \cong E_6$  und  $R_4 \cong A_4$  übrig.

Ist  $r = 5$  so benötigen wir ein Wurzelsystem vom Index 4 mit Rang 5. Ist  $R_5$  unzerlegbar, so bleibt nur  $D_5$  als Möglichkeit übrig. Wäre  $R_5$  zerlegbar, so hätten beide Summanden Zusammenhangsindex 2 oder einer Index 4 und einer Index 1. Da die Ränge sich aber auch zu 5 aufaddieren müssen, fallen die  $E_i$  von vornherein als Summanden raus. Die  $D_l$  können aber auch keine Summanden sein, da der andere Summand Index 1 haben müsste und Rang  $5-l$  haben müsste. Einen solchen gibt es nicht.  $A_1 \times A_1$  hat zwar Index 4 aber nur Rang 2. Für  $r = 3$  muss  $R_3$  Index 6 haben. Wäre  $R_3$  unzerlegbar, so käme nur  $A_5$  in Frage, dies hat aber offensichtlich den falschen Rang. Also ist  $R_3$  zerlegbar und es kommt auf Grund des Ranges nur  $A_1 \times A_2$  in Frage.  $\square$

Nun aber zur Konstruktion von  $E_8$ :

Es sei

$$l = al_0 - \sum_{i=1}^r b_i l_i \in N_r.$$

Dieser Vektor liegt in  $R_r$ , falls er orthogonal zu  $\omega_r$  ist und Länge 2 hat (bezüglich des Skalarproduktes in  $\omega_r^\perp$ ). Für die Koeffizienten  $a, b_i \in \mathbb{Z}$  muss

also gelten:

$$0 = (\omega_r, l) = 3a - \sum_{i=1}^r b_i \quad (1.1)$$

$$2 = (l, l) = -a^2 + \sum_{i=1}^r b_i^2. \quad (1.2)$$

Jetzt setzen wir  $r = 8$ .

**Proposition 1.5.6.** Die Wurzeln von  $R_8$  können aus den Wurzeln  $l$  mit folgenden Koeffizienten durch Vorzeichenwechsel und Permutation der  $b_i$  erhalten werden:

$a$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$
0	1	0	0	0	0	0	0	-1
1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	1	1

*Beweis.* Es ist klar, dass die Tabelle alle ganzzahligen Lösungen des Gleichungssystems anzeigt, die die Bedingungen  $0 \leq a \leq 3$  und  $b_1 \geq \dots \geq b_8$  erfüllen. Nun müssen wir noch zeigen, dass das trotzdem alle Lösungen von 1.1 und 1.2 sind. Hierfür geben wir eine Spiegelung bezüglich einer Wurzel an, die bei allen Wurzeln  $a$  verändert und zeigen dann, dass  $|a|$  trotzdem nicht größer als 3 wird. Es sei also  $(a, b_1, \dots, b_8)$  eine beliebige Lösung des obigen Gleichungssystems. Wir vertauschen falls nötig, die Vorzeichen und permutieren die  $b_i$  so, dass  $a > 0$  und  $b_1 \geq \dots \geq b_8$  gilt. Dann ist mit 1.1

$$a = \frac{1}{3} \sum_{i=1}^r b_i \leq \frac{1}{3} (b_1 + b_2 + b_3 + b_1 + b_2 + b_3 + b_1 + b_2) < b_1 + b_2 + b_3.$$

Setze  $l := (1, 1, 1, 1, 0, 0, 0, 0, 0)$  und  $s_l$  die Spiegelung bezüglich  $l$ . Dann ist

$$\begin{aligned} s_l((a, b_1, \dots, b_r)) &= (a, b_1, \dots, b_r) - 2 \overbrace{\frac{(a - b_1 - b_2 - b_3)}{2}}^{0 > c :=} l \\ &= (a + c, b_1 + c, b_2 + c, b_3 + c, b_4, b_5, b_6, b_7, b_8). \end{aligned}$$

Also wird  $a$  bei der Spiegelung bezüglich  $l$  reduziert. Daher kann man alle Wurzeln durch Permutationen der  $b_i$ , Vorzeichenwechsel und Spiegelung bezüglich  $l$  aus der Tabelle erhalten. Da aber  $|a + c| = |2a - b_i - b_j - b_k| \leq 3$  für alle in der Tabelle aufgeführten Lösungen und deren Negativen gilt, folgt, dass man nie eine Lösung  $(a, b_1, \dots, b_8)$  erhält, mit  $|a| > 3$ .  $\square$

Wieder werden wir die Zahl der kürzesten Wurzeln berechnen:

**Beispiel 1.5.7.** Nach Tabelle 1.5.6 sind die Wurzeln

$$\pm (l_i - l_j), 1 \leq i < j \leq 8 \qquad \text{macht } 2 \cdot \binom{8}{2} = 56$$

$$\pm (l_0 + l_i + l_j + l_k), 1 \leq i < j < k \leq 8 \qquad \text{macht } 2 \cdot \binom{8}{3} = 112$$

$$\pm (2l_0 + l_1 + \cdots + \hat{l}_i + \cdots + \hat{j} + \cdots + l_8), 1 \leq i < j \leq 8 \qquad \text{macht } 2 \cdot \binom{8}{2} = 56$$

$$\pm (3l_0 + l_1 + \cdots + 2l_i + \cdots + l_8), 1 \leq i \leq 8 \qquad \text{macht } 2 \cdot \binom{8}{1} = 16$$

Also insgesamt  $56 + 112 + 56 + 16 = 240$ .

## Kapitel 2

# Die Coxeter-Ebene

Mit Hilfe einer Projektion auf die Coxeter-Ebene möchte man ein zweidimensionales Bild von Wurzelsystemen erzeugen, das möglichst viel der ganzen Symmetrie erhält. Wir werden in allen Abschnitten alle Konstruktionen und Berechnungen am Beispiel von  $A_2$  nachvollziehen. In Abschnitt 1 werden grundlegende Vorbereitungen getroffen und einige wichtige Definitionen aufgeführt. Die eigentliche Konstruktion der Coxeter-Ebene nach Humphreys [1] und Bourbaki [2] folgt dann in Abschnitt 2. In Abschnitt 3 werden wichtige Eigenschaften aufgeführt, aus welchen dann in Abschnitt 4 als Folgerung eine einfachere Konstruktion (nach Stembridge [3]) dieser Ebene gegeben werden kann.

### 2.1 Vorbereitung

Im ganzen Kapitel sei  $e_1, \dots, e_l$  die Standardbasis von  $\mathbb{R}^l$  und  $V$  ein  $n$ -dimensionaler reeller Vektorraum.  $R \subset V$  sei ein nicht notwendigerweise kristallographisches Wurzelsystem (wir erlauben also Koeffizienten aus  $\mathbb{R}$ ) und  $S = \{\alpha_1, \dots, \alpha_n\}$  seine einfachen Wurzeln. Weiter seien  $s_i = s_{\alpha_i}$ , für  $i = 1, \dots, n$  die einfachen Spiegelungen an den zu  $\alpha_i$  orthogonalen Hyperebenen  $H_i$  und  $W = \langle s_1, \dots, s_n \rangle \subset \text{GL}(V)$  die zugehörige Spiegelungsgruppe. Wir können annehmen, dass  $W$  irreduzibel und wesentlich ist und dass alle Wurzeln in  $R$  Vektoren mit Länge 1 sind.

Zunächst benötigen wir die Definition vom Coxeter-Element und der Coxeter-Zahl. Beide werden eine fundamentale Rolle für Eigenschaften der Coxeter-Ebene spielen:

**Definiton 2.1.1.** Das Element  $w := s_1 \cdots s_n \in W$  heißt ein *Coxeter-Element* von  $W$ .

Alle Coxeter-Elemente sind nach [1], Proposition 3.16 unter  $W$  zueinander konjugiert. Daher haben sie die gleiche Ordnung  $h$ . Man kann die Nummerierung der einfachen Wurzeln so wählen, dass  $\alpha_1, \dots, \alpha_r$  und  $\alpha_{r+1}, \dots, \alpha_n$  jeweils paarweise orthogonal sind ([2], S.123).

**Definiton 2.1.2.** Wir nennen  $h = \text{ord}(w)$  die *Coxeter-Zahl*.

**Beispiel 2.1.3.** Sei  $V = \{v \in \mathbb{R}^3 \mid v_1 + v_2 + v_3 = 0\}$  und  $R = \{v \in V \mid |v| = 1\}$ , also ein Wurzelsystem vom Typ  $A_2$ . Es sei weiter  $S = \left\{ \frac{1}{\sqrt{2}}(e_1 - e_2), \frac{1}{\sqrt{2}}(e_2 - e_3) \right\}$  die Basis von  $R$ . Zunächst benötigen wir die Matrizen, die die Spiegelungen an  $\alpha_1$  und  $\alpha_2$  darstellen.

Für  $s_{\alpha_1} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & g & i \end{pmatrix}$  muss gelten:

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & g & i \end{pmatrix} \cdot \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{pmatrix} = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix}, \text{ also: } \begin{pmatrix} a & = & b - 1 \\ d & = & e + 1 \\ g & = & h \end{pmatrix},$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & g & i \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{pmatrix}, \text{ also: } \begin{pmatrix} b & = & c + 1 \\ e & = & f \\ h & = & i - 1 \end{pmatrix}.$$

Also ist

$$s_{\alpha_1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Genauso berechnet man

$$s_{\alpha_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

und somit ist ein Coxeter-Element

$$w = s_{\alpha_1} \cdot s_{\alpha_2} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Die Coxeter-Zahl ist  $h = 3$ .

## 2.2 Konstruktion

In diesem Abschnitt folgen wir [1], §3.18. Von nun an sei  $w$  das Coxeter-Element mit  $w = yz$ , wobei  $y = s_1 \dots s_r$  und  $z = s_{r+1} \dots s_n$  und sowohl  $\alpha_1, \dots, \alpha_r$ , als auch  $\alpha_{r+1}, \dots, \alpha_n$  paarweise orthogonal sind. Es sei weiter  $\omega_1, \dots, \omega_n$  die Basis in  $V$ , sodass  $\langle \omega_i \mid \alpha_j \rangle = \delta_{ij}$ .

Wir benötigen nun

$$Y := \langle \omega_{r+1}, \dots, \omega_n \rangle \text{ und } Z := \langle \omega_1, \dots, \omega_r \rangle.$$

Aus Dimensionsgründen muss gelten:

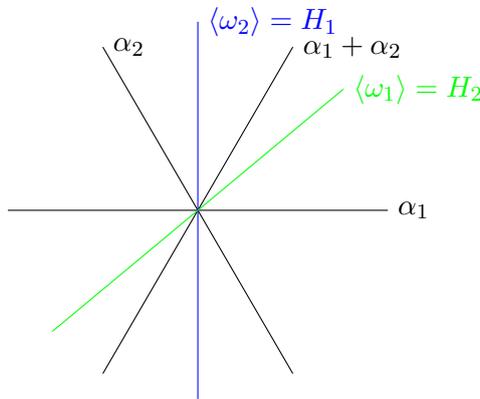
$$Y^\perp = \langle \alpha_1, \dots, \alpha_r \rangle \text{ und } Z^\perp = \langle \alpha_{r+1}, \dots, \alpha_n \rangle.$$

Außerdem ist

$$Y \subset Y' := H_1 \cap \dots \cap H_r \text{ und } Z \subset Z' := H_{r+1} \cap \dots \cap H_n.$$

Da nur 0 orthogonal zu allen  $\alpha_i$  ist, gilt  $Y' \cap Z' = 0$ . Außerdem ist  $V = Y \oplus Z$ , also folgern wir, dass  $Y = Y'$  und  $Z = Z'$ . Zur Veranschaulichung, hier ein Beispiel:

**Beispiel 2.2.1.**  $A_2$ :



**Bemerkung 2.2.2.** Ist ein Vektor  $z'$  (bzw.  $y'$ ) orthogonal zu  $Z$  (bzw.  $Y$ ), so gilt  $z(z') = -z'$  (bzw.  $y(y') = -y'$ ). Liegt  $z'$  (bzw.  $y'$ ) in  $Z$  (bzw.  $Y$ ), so ist  $z(z') = z'$  (bzw.  $y(y') = y'$ ).

Nun sei  $a$  der Endomorphismus auf  $V$ , der jedes  $\omega_i$  auf das entsprechende  $\alpha_i$  abbildet. Die Matrix bezüglich der Basis  $\omega_1, \dots, \omega_n$  ist  $A = ((\alpha_i, \alpha_j))$

Also ist  $A$  positiv definit, symmetrisch mit  $a_{ij} \leq 0$  für  $i \neq j$  und hat somit nach [1], Proposition 2.6 einen positiven Eigenwert  $c$  mit zugehörigem Eigenvektor  $(c_1, \dots, c_n)$  in  $\mathbb{R}^n$ , wo alle  $c_i > 0$ .

Definiere

$$\lambda := \sum_{i=1}^r c_i \omega_i \text{ und } \mu := \sum_{j=r+1}^n c_j \omega_j.$$

Es sei  $P$  die Ebene, die von  $\mathbb{R}\lambda$  und  $\mathbb{R}\mu$  aufgespannt wird.  $P$  heißt *Coxeter-Ebene*.

**Beispiel 2.2.3.** Für  $A_2$  ist:

$$A = \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{pmatrix}.$$

Diese hat einen Eigenwert  $c = \frac{1}{2}$ , der zugehörige Eigenvektor ist  $(1, 1)$ . Damit wird die Coxeter-Ebene von  $\lambda = \omega_1$  und  $\mu = \omega_2$  aufgespannt. Diese können leicht berechnet werden, da  $(\omega_i | \alpha_j) = \delta_{ij}$  gelten muss und die Koordinaten von  $\omega_i$  aufsummiert null ergeben müssen. Also ist

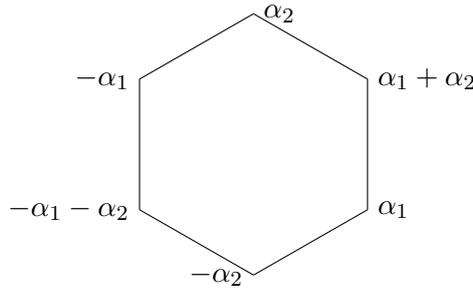
$$P = \left\langle \left( \begin{array}{c} 2\sqrt{2}/3 \\ -\sqrt{2}/3 \\ -\sqrt{2}/3 \end{array} \right), \left( \begin{array}{c} \sqrt{2}/3 \\ \sqrt{2}/3 \\ -2\sqrt{2}/3 \end{array} \right) \right\rangle = \left\langle \left( \begin{array}{c} 2 \\ -1 \\ -1 \end{array} \right), \left( \begin{array}{c} 1 \\ 1 \\ -2 \end{array} \right) \right\rangle = \left\langle \left( \begin{array}{c} \sqrt{2}/\sqrt{3} \\ -1/\sqrt{2}\sqrt{3} \\ -1/\sqrt{2}\sqrt{3} \end{array} \right), \left( \begin{array}{c} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{array} \right) \right\rangle,$$

wobei die letzte Gleichheit nach dem Gram-Schmidtschen Orthonormalisierungsverfahren gilt. Nun können wir die Projektion aller Wurzeln auf  $P = \langle p_1, p_2 \rangle$  direkt berechnen.

Hierbei ist  $\text{pr}(\alpha) = (\alpha | p_1) \begin{pmatrix} 1 \\ 0 \end{pmatrix} + (\alpha | p_2) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ , also:

$$\begin{aligned} \text{pr}(\alpha_1) &= \left( \left( \begin{array}{c} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{array} \right) \middle| \left( \begin{array}{c} \sqrt{2}/\sqrt{3} \\ -1/\sqrt{2}\sqrt{3} \\ -1/\sqrt{2}\sqrt{3} \end{array} \right) \right) \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \left( \left( \begin{array}{c} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{array} \right) \middle| \left( \begin{array}{c} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{array} \right) \right) \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \sqrt{3}/2 \\ -1/2 \end{pmatrix} \\ \text{pr}(\alpha_2) &= \left( \left( \begin{array}{c} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{array} \right) \middle| \left( \begin{array}{c} \sqrt{2}/\sqrt{3} \\ -1/\sqrt{2}\sqrt{3} \\ -1/\sqrt{2}\sqrt{3} \end{array} \right) \right) \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \left( \left( \begin{array}{c} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{array} \right) \middle| \left( \begin{array}{c} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{array} \right) \right) \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

Da die Projektion linear ist, kann man nun ein zweidimensionales Modell von  $A_2$  angeben. Hierbei sei erwähnt, dass zwei Punkte mit einer Linie verbunden werden, falls die zugehörigen Wurzeln nächste Nachbarn sind, das heißt falls die Differenz der Wurzeln wieder eine Wurzel ist:



## 2.3 Eigenschaften

Wir werden nun zeigen, dass die eben konstruierte Coxeter-Ebene unter dem oben festgelegten Coxeter-Element fest bleibt. Außerdem werden wir die Wirkung des Coxeter-Elements auf  $P$  genauer analysieren.

**Satz 2.3.1.** *Sei  $w$  das Coxeter-Element und  $P$  die Coxeter-Ebene. Dann ist  $w(P) = P$ .*

*Beweis.* Seien  $c$  und  $(c_1, \dots, c_n)$  wie eben positiver Eigenwert und Eigenvektor von  $A$ . Es gilt:

$$\sum_{j=1}^n c_j \alpha_j = \sum_{j=1}^n c_j \cdot \sum_{i=1}^n a_{ij} \omega_i \stackrel{c_j, a_{ij} \in \mathbb{R}}{=} \sum_{i=1}^n \omega_i \sum_{j=1}^n a_{ij} c_j = \sum_{i=1}^n \omega_i c c_i.$$

Nun nehmen wir auf jeder Seite das innere Produkt mit  $\alpha_i$  für ein  $i \in \{1, \dots, r\}$ :

$$\begin{aligned} \left( \sum_{j=1}^n c_j \alpha_j \mid \alpha_i \right) &= \left( \sum_{j=1}^n c c_j \omega_j \mid \alpha_i \right) \\ \iff \sum_{j=1}^n c_j (\alpha_j \mid \alpha_i) &= \sum_{j=1}^n c c_j (\omega_j \mid \alpha_i) \\ \xrightarrow[\substack{(\alpha_j \mid \alpha_i) = \delta_{ij} \text{ für } j \in \{1, \dots, r\} \\ (\omega_j \mid \alpha_i) = \delta_{ij}}]{\substack{(\alpha_j \mid \alpha_i) = \delta_{ij} \text{ für } j \in \{1, \dots, r\} \\ (\omega_j \mid \alpha_i) = \delta_{ij}}} c_i + \sum_{j=r+1}^n c_j (\alpha_j \mid \alpha_i) &= c c_i. \end{aligned}$$

Damit berechnen wir:

$$\begin{aligned} (c-1)\lambda &= (c-1) \sum_{i=1}^r c_i \omega_i \\ &= \sum_{i=1}^r \left( \sum_{j=r+1}^n c_j (\alpha_j \mid \alpha_i) \right) \omega_i \\ &= \sum_{j=r+1}^n c_j \left( \sum_{i=1}^r (\alpha_j \mid \alpha_i) \omega_i \right) \\ &= \sum_{j=r+1}^n c_j \left( \underbrace{\sum_{i=1}^n (\alpha_j \mid \alpha_i) \omega_i - \omega_j}_{= \alpha_j} \right) \\ &= \underbrace{\sum_{j=r+1}^n c_j \alpha_j}_{=: \nu} - \mu \end{aligned}$$

Hier sieht man, dass man die Ebene  $P$  auch mit  $\lambda$  und  $\nu$  aufspannen kann. Nun ist  $\nu$  orthogonal zu allen  $\omega_1, \dots, \omega_r$ , da die  $\omega_i$  duale Basis zu den  $\alpha_i$  bilden. Daher ist  $\nu$  auch orthogonal zu  $Z = \langle \omega_1, \dots, \omega_n \rangle$ . Wie in 2.2.2 gilt somit  $z(\nu) = -\nu$ . Außerdem gilt  $z(\lambda) = \lambda$ , da  $\lambda \in Z$ . Damit ist  $z(P) = P$ . Genauso zeigt man  $y(P) = P$  und erhält somit  $w(P) = P$ .  $\square$

**Korollar 2.3.2.** Sei  $w = yz$  das Coxeter-Element wie oben und  $P$  die Coxeter-Ebene. Dann operiert  $w$  auf  $P$  als Drehung.

*Beweis.* Da  $\mu \in P \cap Y$  ist  $y(\mu) = \mu$  und  $y|_P$  eine Spiegelung bezüglich  $\mathbb{R}\mu$ . Genauso ist  $z|_P$  eine Spiegelung bezüglich  $\mathbb{R}\lambda$  (da  $\lambda \in P \cap Z$ ). Damit folgt sofort die Behauptung.  $\square$

Nun ist es aber interessant zu wissen, um welche Drehung es sich handelt.

**Satz 2.3.3.** Es sei  $w$  das Coxeter-Element,  $h$  die Coxeter-Zahl und  $P$  die Coxeter-Ebene. Dann hat  $w$  die Ordnung  $h$  auf  $P$ .

*Beweis.* Nach [1], Theorem 1.12, ist

$$\bar{D} := \left\{ \beta \in V \mid (\beta|\alpha) \geq 0 \ \forall \alpha \in S \right\} = \left\{ \sum_{i=1}^n a_i \omega_i \in V \mid a_i \geq 0 \ \forall i \right\}$$

der Fundamentalbereich von  $W$ .  $\lambda$  und  $\mu$  liegen beide in  $\bar{D}$ . Daher ist

$$P \cap D = \{a\lambda + b\mu \mid a, b \geq 0\} \neq \emptyset.$$

Nun sei  $t$  die Ordnung von  $w$  auf  $P$ . Dann ist  $w^t(p) = p$  für alle  $p \in P$ . Daher existiert ein  $p' \in P \cap D$ , sodass  $w^t(p') = p'$ . Also ist  $w^t = id$  auf ganz  $V$  und somit ist  $t = h$ .  $\square$

Zusammen mit Korollar 2.3.2 sieht man sofort, das folgende Korollar:

**Korollar 2.3.4.** Sei  $w$  das Coxeter-Element,  $h$  die Coxeter-Zahl und  $P$  die Coxeter-Ebene. Dann operiert  $w$  auf  $P$  als Drehung um  $\frac{2\pi}{h}$ .

Es folgt:

**Korollar 2.3.5.** Die primitive  $h$ -te Einheitswurzel  $e^{\frac{2\pi i}{h}}$  ist ein Eigenwert mit Multiplizität 1 von  $w$  auf  $P$ .

*Beweis.* Siehe Korollar 2 in [2], Kapitel 5, § 6, Abschnitt 2.  $\square$

## 2.4 Folgerung

Über das letzte Korollar 2.3.5 kommt man auf eine einfachere Definition der Coxeter-Ebene  $P$ . Diese wollen wir in diesem Abschnitt herleiten.

Sei von nun an  $w$  irgendein Coxeter-Element. Es ist egal, welches wir wählen, da ja wie in Abschnitt 2.1 erwähnt, alle Coxeter-Elemente konjugiert zueinander sind. Es sei weiter  $z$  der Eigenvektor von  $w$  zum Eigenwert  $e^{\frac{2\pi i}{h}}$ . Da  $w \in W$  ist, gilt:

$$(w(\alpha)|z) = (\alpha|w(z)) = e^{\frac{2\pi i}{h}} (\alpha|z) \ \forall \alpha \in V.$$

Das heißt, wenn auf  $z$  projiziert wird, operiert  $w$  als Drehung um  $\frac{2\pi}{h}$  auf allen Wurzeln. Wir können  $P$  also durch  $\text{Re}(z)$  und  $\text{Im}(z)$  aufspannen ([3]).

**Beispiel 2.4.1.** Für  $A_2$  war ein Coxeter-Element

$$w = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Da die Coxeter-Zahl  $h = 3$  war, benötigen wir nun den Eigenvektor  $v$  zum Eigenwert  $e^{\frac{2\pi i}{3}} = \cos\left(\frac{2\pi}{3}\right) + \sin\left(\frac{2\pi}{3}\right) \cdot i = -\frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i$ . Dieser ist

$$z = \begin{pmatrix} 2 \\ -1 - \sqrt{3} \cdot i \\ -1 + \sqrt{3} \cdot i \end{pmatrix}$$

also ist

$$P = \left\langle \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -\sqrt{3} \\ \sqrt{3} \end{pmatrix} \right\rangle.$$

Man sieht gleich, dass diese Ebene gleich derjenigen ist, die wir in 2.2.3 berechnet haben.

Für ein festes Coxeter Element  $w$  ist die Coxeter Ebene eindeutig, da wir in 2.3.5 gesehen haben, dass die primitive  $h$ -te Einheitswurzel ein Eigenwert mit Multiplizität 1 von  $w$  ist. Wählen wir ein anderes Coxeter Element, so kann auch die Coxeter Ebene eine andere sein:

**Beispiel 2.4.2.** Die Weyl Gruppe von  $A_3$  ist nach [1], 2.10  $S_4$ . Diese hat die Erzeuger

$$s_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, s_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, s_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Das Coxeter Element  $w_1 = s_1 s_2 s_3$  hat zum Eigenwert  $e^{\frac{2\pi i}{4}} = i$  den Eigenvektor

$$v_1 = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} + i \cdot \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

und das Coxeter Element  $w_2 = s_1 s_3 s_2$  den Eigenvektor

$$v_2 = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + i \cdot \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}.$$

$v_1$  und  $v_2$  spannen offensichtlich nicht die gleiche Ebene auf.

Im Anhang befindet sich ein Maple Programm, mit dem man sich die Projektionen aller Wurzelsysteme auf die Coxeter-Ebene ausgeben lassen kann. Alle Kanten, deren äußerste Endpunkte auf ein und demselben Ring um den Ursprung liegen, haben die gleiche Farbe. Falls keine zwei unterschiedliche Wurzeln auf den gleichen Punkt projiziert werden, so liegen die projizierten Wurzeln auf konzentrischen Kreisen um den Ursprung, und zwar auf jedem genau  $h$  Stück.

Abbildung 2.1: A5

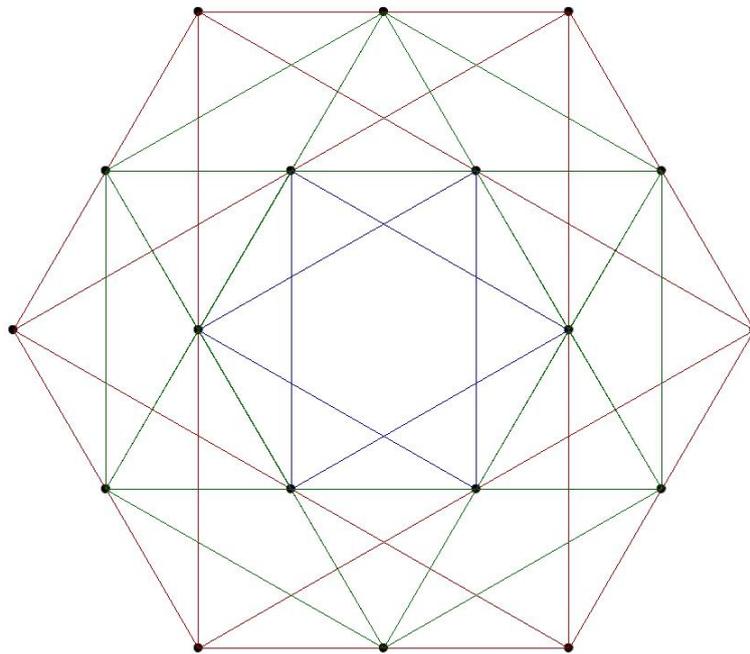


Abbildung 2.2: B4

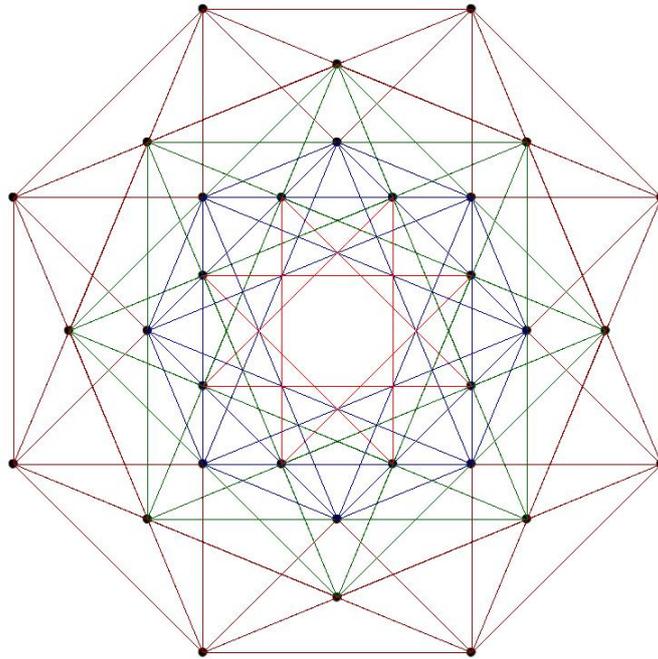


Abbildung 2.3: C4

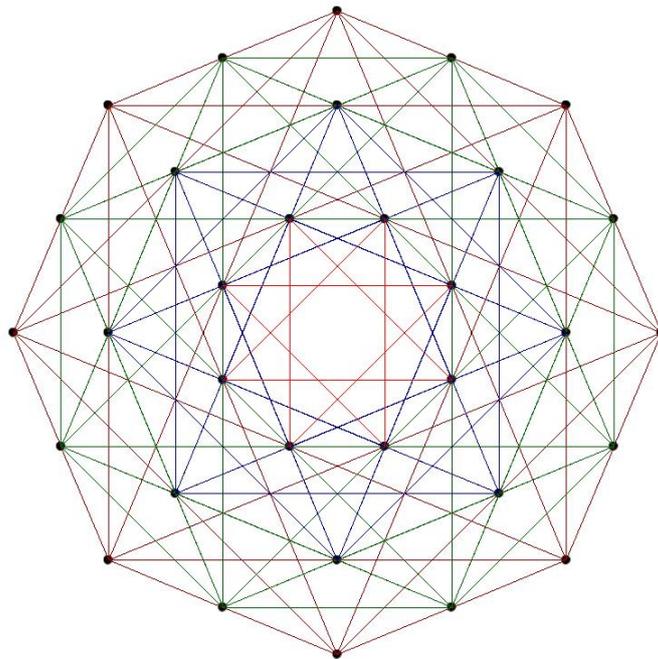


Abbildung 2.4: D6

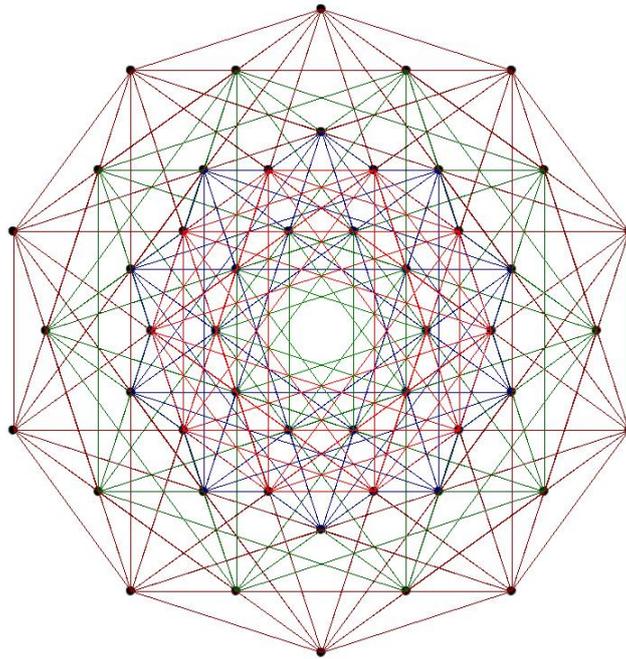


Abbildung 2.5: E6

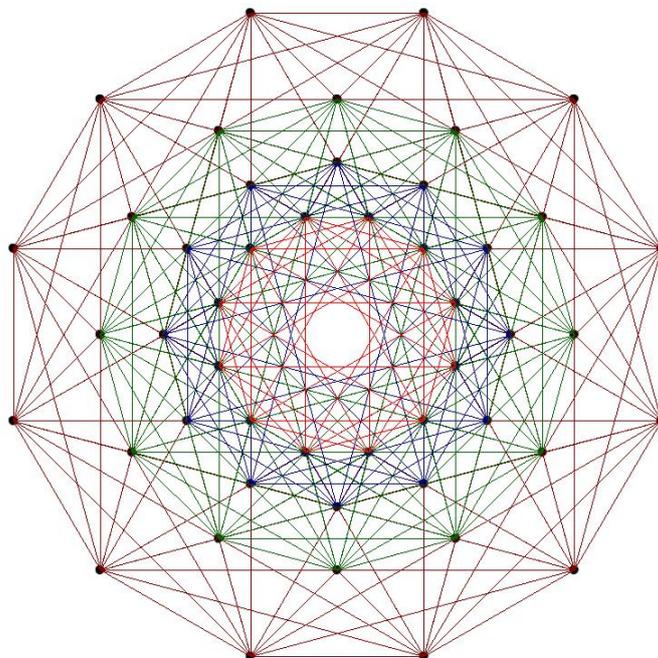


Abbildung 2.6: E7

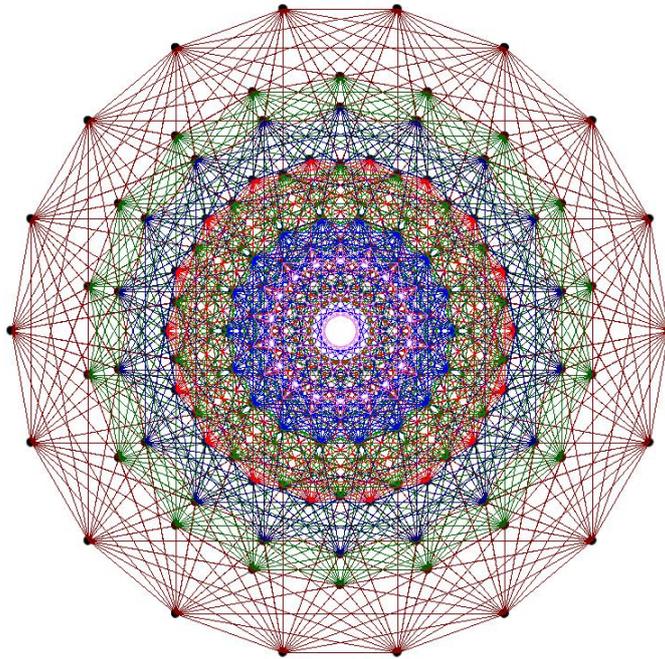
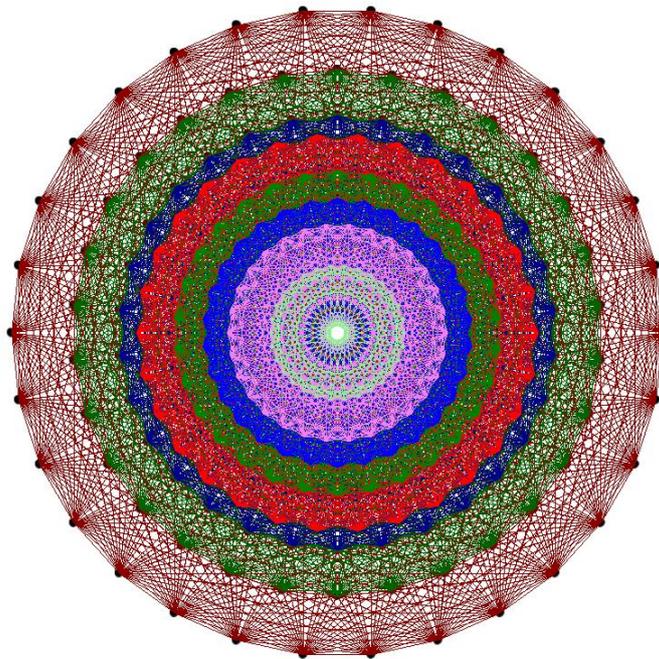
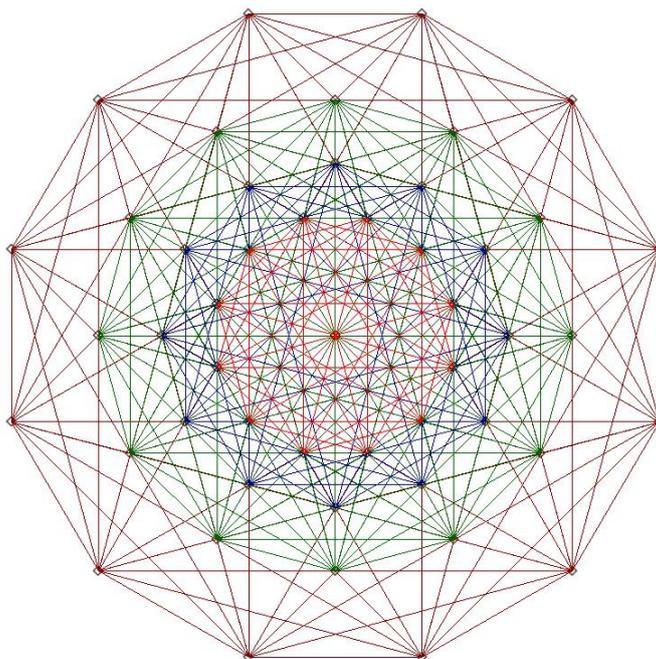


Abbildung 2.7: E8



Man kann die Wurzeln eines Wurzelsystems auch in die Coxeter-Ebene einer Unteralgebra projektieren. So sieht die Projektion von  $E_8$  auf die Coxeter-Ebene von  $E_6$  folgendermaßen aus:

Abbildung 2.8:  $E_8$  in der Coxeter Ebene von  $E_6$



## Kapitel 3

# Platonische Polychora

Die platonischen Polychora sind reguläre, konvexe, vierdimensionale Polytope. Sie bilden sozusagen das vierdimensionale Analogon zu den platonischen Körpern. So entsteht der sogenannte 5-Zeller beispielsweise, wenn man eine Pyramide  $ABCDE$  auf einer Basis  $ABCD$  in Tetraeder-Form baut. Der 8-Zeller ist ein Prisma mit einem Würfel als Basis. Der sogenannte 600-Zeller  $C_{600}$ , spielt eine wichtige Rolle bei der Projektion von  $E_8$  in den dreidimensionalen Raum.

Wir werden in diesem Kapitel zunächst das Schläfli-Symbol<sup>1</sup> einführen, welches reguläre Polytope und Parkettierungen (beliebiger Dimension) beschreibt. Dann werden wir eine notwendige Bedingung dafür angeben, dass ein Polychor regulär ist und anschließend die sechs möglichen Polychora konstruieren und ihre kartesischen Koordinaten angeben. Im ganzen Kapitel folgen wir [5], 22.1-22.3.

### 3.1 Vorbereitung

**Definiton 3.1.1.** Ein *Schläfli-Symbol* hat die Form  $\{p, q, r, \dots\}$  und ist eine rekursive Beschreibung von regulären Polytopen. So stellt  $\{p\}$  ein reguläres Polygon mit  $p$  Ecken dar. Ein reguläres Polyeder ist eine Verklebung gleicher regulärer  $p$ -Ecke, sodass jede Kante zu zwei Flächen gehört und an jeder Ecke  $q$  reguläre  $p$ -Ecke anliegen. Dieses stellt man mit  $\{p, q\}$  dar. So hat zum Beispiel ein Tetraeder an jeder Ecke 3 aneinander liegende gleichseitige Dreiecke und hat somit das Schläfli-Symbol  $\{3, 3\}$ . Bei einem Würfel liegen an jeder Ecke drei Quadrate. Er wird deshalb mit dem Symbol  $\{4, 3\}$  beschrieben. Ein Polychor  $\{p, q, r\}$  ist analog eine Verklebung gleicher regulärer Polyeder  $\{p, q\}$ , die so aneinander liegen, dass jede Fläche  $\{p\}$  zu zwei Zellen  $\{p, q\}$  gehört und an jeder Kante  $r$  Zellen anliegen. So kann man rekursiv alle  $\{p, q, r, \dots, w\}$  definieren.

---

<sup>1</sup>benannt nach dem Schweizer Mathematiker Ludwig Schläfli, 1814-1895

Zur Veranschaulichung möchte ich hier die Schläfli-Symbole der Dimension 3 aufführen:

- Beispiel 3.1.2.**
1. Tetraeder: An jeder Ecke liegen 3 Dreiecke an. Somit hat das Tetraeder Schläfli-Symbol  $\{3, 3\}$ . Die Selbstdualität des Tetraeders erkennt man daran, dass  $p = q$  im Schläfli-Symbol ist.
  2. Würfel: An jeder Ecke liegen 3 Vierecke an. Somit hat der Würfel Schläfli-Symbol  $\{4, 3\}$ .
  3. Oktaeder: An jeder Ecke liegen 4 Dreiecke an. Somit hat der Oktaeder Schläfli-Symbol  $\{3, 4\}$ . Man erkennt an den vertauschten  $p$  und  $q$ , dass Würfel und Oktaeder dual zueinander sind.
  4. Dodekaeder: An jeder Ecke liegen 3 Fünfecke an. Somit hat der Dodekaeder Schläfli-Symbol  $\{5, 3\}$ .
  5. Ikosaeder: An jeder Ecke liegen 5 Dreiecke an. Somit hat der Ikosaeder Schläfli-Symbol  $\{3, 5\}$ . Auch hier erkennt man wieder die Dualität von Dodekaeder und Ikosaeder.

Des weiteren werden wir noch die Eckfigur eines  $\{p, q, r, \dots, w\}$  benötigen. Eine Eckfigur von  $\{p, q\}$  ist die Schnittfläche die entsteht, wenn man eine Ecke des Polyeders so abschneidet, dass man alle angrenzenden Kanten halbiert hat. Die Ecken der Eckfigur sind also die Mittelpunkte der angrenzenden Kanten. Sie bilden ein  $\{q\}$ .

**Definiton 3.1.3.** Die Ecken der Eckfigur eines Polychor  $\{p, q, r\}$  sind die Mittelpunkte der Kanten an einer Ecke des  $\{p, q, r\}$ . Die Eckfigur ist ein  $\{q, r\}$ . Des weiteren ist jede Fläche der Eckfigur selbst Eckfigur einer Zelle, die an der gleichen Ecke des  $\{p, q, r\}$  anliegt. Die Eckfigur eines  $\{p, q, r, \dots, v, w\}$  ist ein  $\{q, r, \dots, v, w\}$ .

**Satz 3.1.4.** *Bezeichnen wir mit  $N_0$  die Anzahl der Ecken, mit  $N_1$  die Anzahl der Kanten, mit  $N_2$  die Anzahl der Flächen und mit  $N_3$  die Anzahl der Zellen so gilt:*

$$N_0 - N_1 + N_2 - N_3 = 0.$$

*Beweis.* Wir benötigen zunächst einige Beobachtungen über Polyeder. Seien hierfür  $V$ ,  $E$  und  $F$  die Anzahl der Knoten, Kanten und Flächen eines Polyeders  $\{p, q\}$ . Dann gilt  $V - E + F = 2$  (vergleiche zum Beispiel [5] S. 152). Außerdem ist

$$qV = 2E = pF. \tag{3.1}$$

Wenn man alle  $q$  Kanten an jedem der  $V$  Knoten zählt hat man jede Kante zwei mal gezählt, da jede Kante zwei angrenzende Knoten hat. Wenn man

alle  $p$  Kanten von jedem der  $F$   $p$ -Ecke zählt, so hat man auch jede Kante zwei mal gezählt, da jede Kante zwei angrenzende Flächen hat. Damit gilt:

$$\frac{V}{1/q} = \frac{E}{1/2} = \frac{F}{1/p} = \frac{V - E + F}{1/q - 1/2 + 1/p} = \frac{2}{1/q + 1/p - 1/2} = \frac{4pq}{2p + 2q - pq}$$

und daher

$$E = \frac{2pq}{2p + 2q - pq} \quad \text{und} \quad \frac{1}{E} = \frac{1}{p} + \frac{1}{q} - \frac{1}{2}. \quad (3.2)$$

Erinnern wir uns nun, dass die Zellen die Form  $\{p, q\}$  haben und sei weiter  $V'$ ,  $E'$  und  $F'$  die Anzahl der Knoten, Kanten und Flächen der Eckfiguren  $\{q, r\}$ .

Die  $F$  Flächen aller  $N_3$  Zellen sind alle  $N_2$  Flächen des gesamten  $\{p, q, r\}$ . Jede Fläche gehört aber zu zwei Zellen und somit gilt:

$$FN_3 = 2N_2. \quad (3.3)$$

Damit ist

$$EN_3 \stackrel{3.1}{=} \frac{1}{2}pFN_3 \stackrel{3.3}{=} pN_2. \quad (3.4)$$

Die Anzahl  $V'$  der Knoten der Eckfigur an einem Knoten sind definitionsgemäß die Anzahl der Kanten an diesem Knoten. Somit sind die  $V'$  Knoten aller  $N_0$  Eckfiguren alle  $N_1$  Kanten von  $\{p, q, r\}$  aber zwei mal gezählt, da jede Kante an zwei Knoten angrenzt. Somit gilt:

$$V'N_0 = 2N_1. \quad (3.5)$$

Wenn wir  $E$  Kanten von jeder der  $N_3$  Zellen zählen, so haben wir jede Kante  $r$  mal gezählt da  $r$  Zellen um jede Kante liegen:

$$EN_3 = rN_1. \quad (3.6)$$

Damit gilt:

$$rN_1 \stackrel{3.5}{=} \frac{1}{2}rV'N_0 \stackrel{3.1}{=} E'N_0. \quad (3.7)$$

Also ist

$$E'N_0 = rN_1 = pN_2 = EN_3$$

und mit 3.2 sind die Zahlen  $N_0 : N_1 : N_2 : N_3$  proportional zu

$$\frac{1}{q} + \frac{1}{r} - \frac{1}{2} : \frac{1}{r} : \frac{1}{p} : \frac{1}{p} + \frac{1}{q} - \frac{1}{2}.$$

Daher gilt

$$N_0 - N_1 + N_2 - N_3 = 0$$

□

**Bemerkung 3.1.5.** Diese Formel hängt, wie auch schon für den dreidimensionalen Fall eng mit der Eulercharakteristik der Einheitskugel  $\chi(S^3) = 0$  zusammen. Nach [23], Kapitel 7 ist die Eulercharakteristik eines CW Komplexes als die alternierende Summe der Zellenanzahlen definiert, also

$$\chi(X) := \sum_{i=0}^n (-1)^i \cdot (\text{Anzahl der } i\text{-Zellen}).$$

Hierbei ist eine  $i$ -Zelle ein topologischer, zu  $\mathbb{R}^i$  homöomorpher Raum. Die  $i$ -Sphäre  $S^i$  kann in eine 0-Zelle (ein Punkt  $a$ ) und eine  $i$ -Zelle ( $S^i \setminus a$ ) zerlegt werden. Daher ist

$$\chi(S^n) = 1 + (-1)^n.$$

Allgemein kann man zeigen [15], dass für einen CW-Komplex  $X$  gilt:

$$\chi(X) = \sum_{i=0}^{\infty} (-1)^i \dim(H_i(X)),$$

wo  $H_i(X)$  die  $i$ -te Homologiegruppe von  $X$  bezeichnet.

**Satz 3.1.6.** *Die einzig möglichen endlichen, vierdimensionalen, regulären Polytope sind*

$$\{3, 3, 3\}, \{3, 3, 4\}, \{3, 3, 5\}, \{4, 3, 3\}, \{3, 4, 3\} \text{ und } \{5, 3, 3\}.$$

*Beweis.* Der Beweis beruht auf zwei Überlegungen:

1.: Die Summe der Raumwinkel der Polyeder, die um eine Kante liegen muss kleiner als  $360^\circ$  sein.

2.: Es müssen mindestens 3 Polyeder um eine Kante liegen. Der Raumwinkel eines Tetraeders ist etwa  $70,53^\circ$ . Daher kann man nur drei, vier oder fünf Tetraeder um eine gemeinsame Kante legen. Also sind  $\{3, 3, 3\}$ ,  $\{3, 3, 4\}$  und  $\{3, 3, 5\}$  die einzigen Möglichkeiten.

Die Raumwinkel von Würfels, Oktaeder und Dodekaeder liegen zwischen  $90^\circ$  und  $120^\circ$ . Daher kann man nur drei um eine gemeinsame Kante legen und  $\{4, 3, 3\}$ ,  $\{3, 4, 3\}$  und  $\{5, 3, 3\}$  sind die einzigen Möglichkeiten.

Das Ikosaeder hat einen Raumwinkel größer als  $120^\circ$  und kann deshalb so nicht verwendet werden.  $\square$

**Satz 3.1.7.** *Die notwendige Bedingung, damit ein vierdimensionales Polychoron mit Schläfli Symbol  $\{p, q, r\}$  regulär und endlich ist, ist*

$$\cos\left(\frac{\pi}{q}\right) < \sin\left(\frac{\pi}{p}\right) \sin\left(\frac{\pi}{r}\right).$$

*Beweis.* Die Raumwinkel eines platonischen Körpers  $\{p, q\}$  ist  $2 \cdot \arcsin\left(\frac{\cos(\pi/q)}{\sin(\pi/p)}\right)$ . Da  $r$  solcher Winkel zusammen kleiner als  $2\pi$  sein müssen, muss jeder kleiner als  $2\pi/r$  sein. Somit muss  $\arcsin\left(\frac{\cos(\pi/q)}{\sin(\pi/p)}\right) < \frac{\pi}{r}$  sein und es folgt direkt die Behauptung.  $\square$

Es bleibt noch zu zeigen, dass die Bedingung in 3.1.7 auch hinreichend ist. Das macht man, indem man die vierdimensionalen regulären Polychora tatsächlich konstruiert. Siehe hierzu [4], S.145-153.

## 3.2 Die regelmäßigen Polychora

### 3.2.1 Der 5-Zeller, auch 4-Simplex oder Hypertetraeder

Der selbstduale 5-Zeller  $\{3, 3, 3\}$  kann konstruiert werden, indem man ein Tetraeder ABCD im  $\mathbb{R}^4$  betrachtet und einen fünften Punkt E außerhalb des 3-Raumes wählt, in dem das Tetraeder lebt. Der Punkt E muss den gleichen Abstand zu jedem der Punkte A, B, C und D haben, wie A, B, C und D untereinander. Dann verbindet man den Punkt E mit jedem der anderen Punkte. Man konstruiert also eine vierdimensionale Pyramide mit einem Tetraeder als Basis. Da man jede der Ecken A,B,C, D und E als Spitzen betrachten kann und jeder Punkt mit jedem verbunden ist sieht man leicht, dass der 5-Zeller aus fünf Tetraedern, zehn Dreiecken, zehn Kanten und fünf Ecken besteht. Die Koordinaten eines 5-Zellers, der wie beschrieben auf dem gleichseitigen Dreieck ABC aufgebaut wurde, sind  $A = (2, 0, 0, 0)$ ,  $B = (0, 2, 0, 0)$ ,  $C = (2, 2, 0, 0)$ ,  $D = (1, 1, 2\sqrt{2} - 2, 0)$  und  $E = \left(1, 1, \frac{7\sqrt{2}+9}{2}, \frac{\sqrt{118\sqrt{2}-187}}{2}\right)$ . Um jede Kante liegen drei und um jede Ecke vier Tetraeder.

### 3.2.2 Der 8-Zeller, auch 4-Maßpolytop oder Hyperwürfel

Der 8-Zeller  $\{4, 3, 3\}$  kann konstruiert werden, indem man einen Würfel AB-CDEFGH entlang der vierten Dimension verschiebt, so wie man einen (dreidimensionalen) Würfel erhält, wenn man ein Quadrat senkrecht zu der Ebene verschiebt, in der es liegt. Zu den acht Ecken des Würfels kommen also noch 8 Ecken des verschobenen Würfels hinzu. Man erhält durch diese Verschiebung insgesamt acht Würfel als Zellen, nämlich den ursprünglichen Würfel, den verschobenen Würfel und sechs, die durch die Verschiebung der sechs Seiten des ursprünglichen Würfels entstehen. Es ergeben sich 24 Quadrate und 32 Kanten. Man sieht leicht, dass ein so konstruierter 8-Zeller, der seinen Mittelpunkt im Ursprung haben soll die Koordinaten  $(\pm 1, \pm 1, \pm 1, \pm 1)$  hat. Um jede Kante liegen drei und um jede Ecke vier Würfel.

### 3.2.3 Der 16-Zeller, auch 4-Kreuzpolytop

Wir werden den 16-Zeller anhand von Analogien zum Dreidimensionalen konstruieren. Man kann einen Tetraeder in einen Würfel einschreiben, indem man als Ecken des Tetraeders Ecken des Würfels so auswählt, dass die Ecken im Würfel nicht verbunden sind.

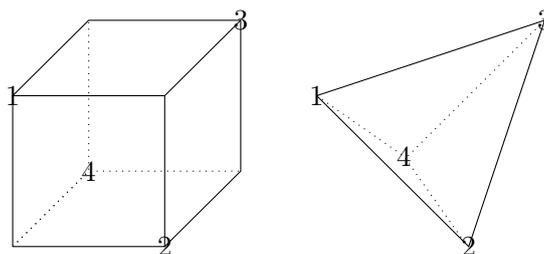


Abbildung 3.1

Die Seiten des Tetraeders können den verbleibenden Ecken des Würfels zugeordnet werden und die sechs Kanten bilden je eine der Diagonalen der Seiten des Würfels. Somit bilden die Mittelpunkte der Kanten des Tetraeders die Eckpunkte des in den Würfel eingeschriebenen Oktaeders.

Verfahren wir nun analog mit dem 8-Zeller  $\{4, 3, 3\}$ : Wir wählen zunächst alternierende Ecken von  $\{4, 3, 3\}$ . Man erhält ein Polytop mit acht Ecken. Die Zellen des eingeschriebenen Polytops (Tetraeder) können den ausgelassenen Ecken des 8-Zellers zugeordnet werden. Zusätzlich zu diesen acht Tetraedern erhält man noch acht weitere, nämlich je eines, das in je eine Zelle (Würfel) des  $\{4, 3, 3\}$  eingeschrieben ist. Man erhält 16 Zellen für dieses Polychor, daher heißt es 16-Zeller. Die Kanten des 16-Zellers sind 24 Diagonalen der 24 Flächen des 8-Zellers. Jede dieser Kanten gehört zu zwei Tetraedern des ersten und zwei Tetraedern des zweiten Typs. Das Schläfli-Symbol des 16-Zellers ist  $\{3, 3, 4\}$ . Er ist dual zum Hyperwürfel.

Sieht man sich die Anzahl der Ecken, Kanten, Flächen und Zellen des 8-Zellers noch einmal an, so sieht man, dass der 16-Zeller dual zu diesem ist. Man hätte die Ecken des 16-Zellers also auch als die Zellenmittelpunkte des 8-Zellers wählen können. Tut man dies, so erhält man, dass die Koordinaten der Ecken des 16-Zellers  $(\pm 1, 0, 0, 0)$ ,  $(0, \pm 1, 0, 0)$ ,  $(0, 0, \pm 1, 0)$  und  $(0, 0, 0, \pm 1)$  sind. Wir werden später aber auch noch eine leichtere Methode zur Konstruktion eines Kreuzpolytopes kennen lernen.

### 3.2.4 Der 24-Zeller

Der selbstduale 24-Zeller  $\{3, 4, 3\}$  wird aus dem 16-Zeller konstruiert. Man nimmt als Ecken die Mittelpunkte der 24 Kanten. Als Zellen ergeben sich zum einen die acht Oktaeder-Eckfiguren der Ecken des 16-Zellers und zum anderen die 16 Oktaeder, die in die 16 Tetraeder eingeschrieben sind. Der

24-Zeller besteht also aus 24 Oktaedern, 96 Dreiecken und 96 Kanten. Berechnet man die Ecken als Mittelpunkte der 24 Kanten des 16-Zellers mit Ecken  $(\pm 2, 0, 0, 0)$  mit allen Permutationen, so erhält man alle Permutationen von  $(\pm 1, \pm 1, 0, 0)$ , die Ecken des 24-Zellers. Der 24-Zeller ist selbstdual und die Ecken des dualen sind  $(\pm 1, 0, 0, 0)$  und alle Permutationen und  $(\pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2)$ . Die Wurzeln des Wurzelsystems  $F_4$  sind gerade die Ecken des 24-Zellers und seinem dualen.

### 3.2.5 Der 600-Zeller

Um den 600-Zeller  $\{3, 3, 5\}$  konstruieren zu können, benötigt man zunächst ein semireguläres Polychor, den schrägen 24-Zeller. Die Ecken des  $s\{3, 4, 3\}$  erhält man, wenn man die 96 Kanten des 24-Zellers im Verhältnis  $\tau : 1 = \frac{\sqrt{5}+1}{2} : 1$  teilt. Er hat 24 Ikosaeder und 120 Tetraeder als Zellen, 480 Dreiecke, 432 Kanten und 96 Ecken mit Koordinaten  $(\pm\tau, \pm 1, \pm\sigma, 0)$  und geraden Permutationen. Hierbei sei  $\sigma = \tau^{-1} = \frac{\sqrt{5}-1}{2}$ . Für eine ausführliche Konstruktion des schrägen 24-Zellers siehe [4], Seite 151f.

Da der Umkugelradius eines Ikosaeders kleiner als seine Kantenlänge ist, kann man im Vierdimensionalen eine Pyramide konstruieren, die einen Ikosaeder als Basis und 20 reguläre Tetraeder als verbleibende Zellen hat. Wir setzen so eine Pyramide auf jedes Ikosaeder des  $s\{3, 4, 3\}$ . So ersetzen wir jedes Ikosaeder durch ein Cluster von 20 Tetraedern mit einer neuen Ecke (Spitze der Pyramide), zwölf neuen Kanten und 30 neuen Dreiecken. So erhalten wir ein Polytop mit  $96 + 1 \cdot 24 = 120$  Ecken,  $432 + 12 \cdot 24 = 720$  Kanten,  $480 + 30 \cdot 24 = 1200$  Dreiecken und  $120 + 20 \cdot 24 = 600$  Tetraedern. Die 24 "neuen" Ecken entsprechen den Zellen des  $\{3, 4, 3\}$  und sind die Ecken des dualen  $\{3, 4, 3\}$ . Sie haben also die Koordinaten  $(\pm 2, 0, 0, 0)$ , alle Permutationen und  $(\pm 1, \pm 1, \pm 1, \pm 1)$ .

### 3.2.6 Der 120-Zeller

Den 120-Zeller  $\{5, 3, 3\}$  konstruieren wir als duales Polychor zum 600-Zeller. Die 600 Ecken sind die Mittelpunkte der 600 Tetraeder. Der 120-Zeller besteht aus 120 Dodekaedern, 720 Fünfecken, 1200 Kanten und 600 Ecken. Die

kartesischen Koordinaten der Ecken sind

$(\pm 2, \pm 2, 0, 0)$  und alle Permutationen ergibt 24 Vektoren  
 $(\pm\sqrt{5}, \pm 1, \pm 1, \pm 1)$  und alle Permutationen ergibt 64 Vektoren  
 $(\pm\tau, \pm\tau, \pm\tau, \pm\sigma^2)$  und alle Permutationen ergibt 64 Vektoren  
 $(\pm\tau^2, \pm\sigma, \pm\sigma, \pm\sigma)$  und alle Permutationen ergibt 64 Vektoren  
 $(\pm\tau^2, \pm\sigma^2, \pm 1, 0)$  und alle geraden Permutationen ergibt 96 Vektoren  
 $(\pm\sqrt{5}, \pm\sigma, \pm\tau, 0)$  und alle geraden Permutationen ergibt 96 Vektoren  
 $(\pm 2, \pm 1, \pm\tau, \pm\sigma)$  und alle geraden Permutationen ergibt 192 Vektoren.

Um jede Kante liegen drei und um jede Ecke vier Dodekaeder.

### 3.3 Reguläre Polytope in $n$ Dimensionen

Für  $n \geq 5$  gibt es nur noch drei reguläre Polytope. Diese möchten wir hier konstruieren und folgen hierbei [4], Kapitel 7.

#### Der reguläre $n$ -Simplex $\alpha_n$

Wie schon der Hypertetraeder (oder reguläre 4-Simplex) aus dem Tetraeder (oder regulären 3-Simplex) konstruiert worden ist, kann man dies natürlich beliebig oft fortführen. Je  $n + 1$  Punkte, die nicht in einem  $(n - 1)$ -Raum liegen und gleichen Abstand zueinander haben, bilden einen regulären  $n$  dimensionalen Simplex  $\alpha_n$ . Die Elemente sind Simplexe, die von Teilmengen der  $n + 1$  Punkte geformt werden. Die Anzahl der  $k$ -dimensionalen Flächen ist

$$N_k = \binom{n+1}{k+1}.$$

Im  $n + 1$ -dimensionalen Raum kann man in einem Punkt  $O$   $n + 1$  paarweise orthogonale Linien anlegen. Dann bilden die  $n + 1$  Punkte entlang dieser Linien mit gleichem Abstand zu  $O$  einen regulären  $n$ -Simplex. Ein so erzeugter Simplex mit Koordinaten  $(1, 0, \dots, 0)$  und allen Permutationen hat Kantenlänge  $\sqrt{2}$ . Der reguläre  $n$ -Simplex für  $n > 1$  hat Schläfli-Symbol  $\{3, \dots, 3\} = \{3^{n-1}\}$ .

#### Das reguläre Kreuzpolytop $\beta_n$

Legt man nun im  $n$ -dimensionalen Raum durch einen Punkt  $O$   $n$  paarweise orthogonale Geraden und markiert von  $O$  aus entlang jeder dieser Geraden in beiden Richtungen  $2n$  Punkte, alle mit dem gleichen Abstand zu  $O$ ,

so erhält man ein reguläres Kreuzpolytop  $\beta_n$ . Die Zellen bestehen aus  $2^n$   $(n-1)$ -Simplexen. Die Permutationen von  $(\pm 1, 0, \dots, 0)$  bilden ein reguläres Kreuzpolytop mit Kantenlänge  $\sqrt{2}$ . Für  $n=2$  erhält man so ein Quadrat, die Zellen bestehen also aus 4 Punkten.  $\beta_3$  ist ein Oktaeder, seine 2-Zellen bestehen aus 8 gleichseitigen Dreiecken.  $\beta_n$  ist eine Dipyramide mit einem  $\beta_{n-1}$  als Basis. Daher sind alle Elemente entweder Elemente von  $\beta_{n-1}$  oder Pyramiden mit solchen Elementen als Basis. Daher sind alle Elemente Simplexe und die Anzahl  $N_k$  der  $\alpha_k$  in  $\beta_n$  ist die Anzahl  $N'_k$  der  $\alpha_k$  in  $\beta_{n-1}$  plus zwei mal die Anzahl der  $\alpha_{k-1}$  in  $\beta_{n-1}$ :

$$N_k = N'_k + 2N'_{k-1}.$$

Die Anzahl der Punkte  $N_0$  ist wie oben schon erwähnt  $2n$  und man zeigt leicht mit Induktion, dass

$$N_k = 2^{k+1} \binom{n}{k+1}.$$

Das reguläre  $n$ -Kreuzpolytop hat Schläfli-Symbol  $\{3, \dots, 3, 4\} = \{3^{n-2}, 4\}$ .

### Das reguläre Maßpolytop $\gamma_n$

Eine dritte Konstruktionsmöglichkeit ist die, wie wir schon den Würfel aus dem Quadrat und den Hyperwürfel aus dem Würfel erhalten haben. Wir verschieben  $\gamma_{n-1}$  so, dass an jeder Ecke alle anliegenden  $n$  Kanten paarweise orthogonal zueinander sind und die gleiche Länge haben und erhalten so  $\gamma_n$ .  $\gamma_0$  ist wieder nur ein Punkt. Das Maßpolytop hat  $2^n$  Ecken. Sei  $N_k$  die Anzahl von  $\gamma_k$ 's in  $\gamma_n$ . Diese Anzahl berechnet sich aus zwei mal der Anzahl von  $\gamma_k$ 's in  $\gamma_{n-1}$  (einmal für das  $\gamma_{n-1}$ , mit dem gestartet wird und einmal für das, in das es verschoben wird) und der Anzahl der  $\gamma_{k-1}$  in  $\gamma_{n-1}$ , die durch das Verschieben zu  $\gamma_k$ 's in  $\gamma_n$  werden. So ist

$$N_k = 2N'_k + N'_{k-1},$$

also wieder mit Induktion:

$$N_k = 2^{n-k} \binom{n}{k}.$$

Das reguläre  $n$ -Maßpolytop hat Schläfli-Symbol  $\{4, 3, \dots, 3\} = \{4, 3^{n-2}\}$ .

### Klassifikation

Wie schon in drei und in vier Dimensionen gibt es eine Bedingung, dass ein  $n$ -dimensionales Polytop existiert:

Nach [4], Seite 130 hat jedes reguläre Polytop  $\Pi_n$  ein Zentrum  $O_n$ , um das wir Sphären mit Radius  $_jR$  durch die Zentren der in  $\Pi_n$  enthaltenen  $\Pi_j$  (für

$j \in \{0, \dots, n-1\}$ ) legen können.  $O_0$  ist somit eine Ecke, eine Kante wird durch  $O_1$  in zwei gleiche Teile geteilt, ein reguläres  $p$ -Eck  $\{p\}$  hat Zentrum  $O_2$  und so weiter. Der Radius  ${}_0R$  ist natürlich der Umkugelradius und  ${}_{n-1}R$  der Inkugelradius. Der Winkel  $\Phi = O_0O_2O_1$  in einem  $\{p\}$  mit Kantenlänge  $2l$  ist  $\Phi = \pi/p$  und  ${}_0R \sin \Phi = l$ . Sei nun analog bei einem  $\Pi_n = \{p, q, r, \dots, w\}$   $\Phi = O_0O_nO_1$ . Weiterhin bleibt

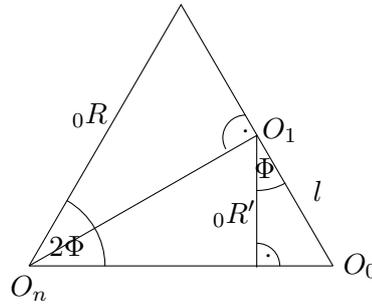
$${}_0R \sin \Phi = l.$$

Es seien  ${}_0R', l', \Phi'$  die entsprechenden Werte der Eckfigur  $\{q, \dots, v, q\}$  von  $\{p, q, \dots, v, w\}$ . Dann ist

$${}_0R' \sin \Phi' = l' = l \cos \pi/p$$

und  ${}_0R' = l \cos \Phi$ . Dies wollen wir kurz verdeutlichen:

Abbildung 3.2: Wie in [4], Seite 22 leicht verändert



Es gilt also

$$\begin{aligned} \cos \Phi &= \frac{\cos \pi/p}{\sin \Phi'} \\ \sin^2 \Phi &= 1 - \frac{\cos^2 \pi/p}{\sin^2 \Phi'}. \end{aligned}$$

Führen wir das so weiter aus und definieren wir  $\Phi^{(i)}$  als den entsprechenden Winkel in der  $i$ -ten Eckfigur, so erhalten wir

$$\sin^2 \Phi' = 1 - \frac{\cos^2 \pi/q}{\sin^2 \Phi''}, \dots, \sin^2 \Phi^{(n-3)} = 1 - \frac{\cos^2 \pi/v}{\sin^2 \Phi^{(n-2)}}.$$

Da  $\Phi^{(n-2)} = \pi/w$  gilt, ist daher

$$\begin{aligned} \sin^2 \Phi &= 1 - \frac{\cos^2 \pi/p}{1 - \frac{\cos^2 \pi/q}{1 - \dots \frac{\cos^2 \pi/v}{1 - \sin^2 \pi/w}}} \\ &= \frac{\Delta_{p,q,\dots,v,w}}{\Delta_{q,\dots,v,w}}, \end{aligned}$$

wobei die  $\Delta$ -Funktion folgendermaßen rekursiv definiert ist:

**Definiton 3.3.1.**

$$\Delta_{p,q,r,\dots,w} := \Delta_{q,r,\dots,w} - \Delta_{r,\dots,w} \cdot \cos^2 \pi/p \text{ mit } \Delta := 1 \text{ und } \Delta_p := \sin^2 \pi/p.$$

Nun erhalten wir also

$$\sin^2 \Phi \sin^2 \Phi' \dots \sin^2 \Phi^{(n-2)} = \Delta_{p,q,\dots,w}.$$

Für ein endliches Polytop muss also gelten:

$$\Delta_{p,q,\dots,w} > 0.$$

Gilt in dieser Gleichung Gleichheit, so stellt  $\{p, q, \dots, w\}$  eine Parkettierung des gesamten  $\mathbb{R}^n$  dar. Wir möchten nachvollziehen, dass diese Gleichung mit den uns bekannten für  $n = 3$  und  $n = 4$  stimmt und eine für  $n = 5$  angeben:

**Beispiel 3.3.2.** 1.  $n = 3$ : Haben wir  $\{p, q\}$ , so ist die Bedingung

$$\begin{aligned} \Delta_{p,q} &= \Delta_q - \Delta \cos^2 \pi/p = \sin^2 \pi/q - \cos^2 \pi/p = \sin^2 \pi/q - 1 + \sin^2 \pi/p > 0 \\ &\Leftrightarrow \sin^2 \pi/q + \sin^2 \pi/p > 1. \end{aligned}$$

Da  $p, q \geq 2$  ist diese wegen dem Verlauf der Funktion  $\sin^2 \pi/x$  genau dann erfüllt, wenn

$$\frac{1}{p} + \frac{1}{q} > \frac{1}{2}$$

gilt.

2.  $n = 4$ : Haben wir ein  $\{p, q, r\}$ , so ist die Bedingung

$$\begin{aligned} \Delta_{p,q,r} &= \Delta_{q,r} - \Delta_r \cos^2 \pi/p = \Delta_r - \Delta \cos^2 \pi/q - \Delta_r \cos^2 \pi/p \\ &= \sin^2 \pi/r - \cos^2 \pi/q - \sin^2 \pi/r \cos^2 \pi/p \\ &= \sin^2 \pi/r (1 - \cos^2 \pi/p) - \cos^2 \pi/q > 0 \\ &\Leftrightarrow \sin \pi/r \sin \pi/p > \cos \pi/q, \end{aligned}$$

welche wir auch in 3.1.7 angegeben haben.

3.  $n = 5$ : Haben wir ein  $\{p, q, r, s\}$ , so ist

$$\begin{aligned} \Delta_{p,q,r,s} &= \Delta_{q,r,s} - \Delta_{r,s} \cos^2 \pi/p \\ &= \sin^2 \pi/s - \cos^2 \pi/r - \sin^2 \pi/s \cos^2 \pi/q - (\Delta_s - \Delta \cos^2 \pi/r) \cos^2 \pi/p \\ &= \sin^2 \pi/s - \cos^2 \pi/r - \sin^2 \pi/s \cos^2 \pi/q - \sin^2 \pi/s \cos^2 \pi/p + \cos^2 \pi/r \cos^2 \pi/p \\ &= \sin^2 \pi/s \sin^2 \pi/p - \cos^2 \pi/r \sin^2 \pi/p - \sin^2 \pi/s \cos^2 \pi/q > 0 \\ &\Leftrightarrow 1 - \frac{\cos^2 \pi/r}{\sin^2 \pi/s} - \frac{\cos^2 \pi/q}{\sin^2 \pi/p} > 0 \\ &\Leftrightarrow \frac{\cos^2 \pi/r}{\sin^2 \pi/s} + \frac{\cos^2 \pi/q}{\sin^2 \pi/p} < 1. \end{aligned}$$

Damit kann man nun zeigen, dass die einzigen Polytope in 5 Dimensionen die folgenden sind:

$\alpha_5 = \{3, 3, 3, 3\}$ , der reguläre 5-Simplex

$\beta_5 = \{3, 3, 3, 4\}$ , das reguläre 5-Kreuzpolytop und

$\gamma_5 = \{4, 3, 3, 3\}$ , das reguläre 5- Maßpolytop.

Mit Induktion folgt dann, dass in mehr als fünf Dimensionen  $\alpha_n$ ,  $\beta_n$  und  $\gamma_n$  die einzigen regulären Polytope sind.

## Kapitel 4

# Dreidimensionale Bilder von $E_8$

Die Ecken des 600-Zellers können mit den sogenannten Ikosianen identifiziert werden. Diese sind spezielle Quaternionen, die, mit zwei verschiedenen Normen versehen, einmal zu den Ecken des 600 Zellers und einmal zu  $E_8$  isomorph sind.

### 4.1 Vorbereitung

Zunächst klären wir einige Notationen. Hierbei folgen wir [6], Seite 207ff und [7]

**Notation 4.1.1.** In diesem Kapitel sei  $\mathbb{F} = \mathbb{Q}(\sqrt{5})$  und

$$' : \mathbb{F} \rightarrow \mathbb{F} \quad (a + b\sqrt{5})' = a - b\sqrt{5}$$

die Konjugationsabbildung. Es seien weiter  $\tau$  und  $\sigma$  die reellen Nullstellen von  $x^2 - x - 1$ , also

$$\tau := \frac{1 + \sqrt{5}}{2} \quad \text{und} \quad \sigma := \frac{1 - \sqrt{5}}{2}.$$

Man kann leicht nachrechnen, dass  $\sigma + \tau = 1$ ,  $\sigma\tau = -1$ ,  $\sigma^2 = 1 + \sigma$  und  $\tau^2 = 1 + \tau$ .

Wir wollen nun Quaternionen mit Koeffizienten aus  $\mathbb{F}$  konstruieren.

**Notation 4.1.2.** Es sei  $\mathbb{H}$  die *Quaternionenalgebra* über  $\mathbb{R}$ , also

$$\mathbb{H} = \{z = z_1 + z_2i + z_3j + z_4k \mid z_1, z_2, z_3, z_4 \in \mathbb{R}, i^2 = j^2 = k^2 = -1 \\ \text{und } ij = -ji = k, jk = -kj = i, ki = -ik = j\}.$$

Auf Grund der Länge dieser Zahlen werden wir Quaternionen als 4-Tupel  $(z_1, z_2, z_3, z_4)$  geschrieben. Das Konjugierte  $\bar{z}$  von  $z$  ist wie gewohnt  $\bar{z} =$

$z_1 - z_2i - z_3j - z_4k$ . Nun werden wir noch das innere Produkt von zwei Quaternionen  $x := (x_1, x_2, x_3, x_4)$  und  $y := (y_1, y_2, y_3, y_4)$  auf  $\mathbb{H}$  definieren.

$$x \cdot y = \frac{1}{2} (x\bar{y} + y\bar{x}) = x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4.$$

Die hierdurch definierte *Quaternionen-Norm* von  $z$  ist

$$\text{QN}(z) = z\bar{z} = z_1^2 + z_2^2 + z_3^2 + z_4^2.$$

Des weiteren bezeichnen

$$\mathbb{H}^0 := \mathbb{R}i + \mathbb{R}j + \mathbb{R}k$$

die *reinen Quaternionen*. Sie haben die Eigenschaft, dass  $\bar{z} = -z$  für  $z \in \mathbb{H}^0$ . Außerdem seien  $\mathbb{H}_{\mathbb{F}}$  und  $\mathbb{H}_{\mathbb{Q}}$  die  $\mathbb{F}$ - und  $\mathbb{Q}$ -Quaternionen-Algebren in  $\mathbb{H}$ :

$$\mathbb{H}_{\mathbb{F}} := \mathbb{F} + \mathbb{F}i + \mathbb{F}j + \mathbb{F}k \text{ und } \mathbb{H}_{\mathbb{Q}} := \mathbb{Q} + \mathbb{Q}i + \mathbb{Q}j + \mathbb{Q}k.$$

Man kann die Konjugationsabbildung auf  $\mathbb{F}$  zu einem  $\mathbb{Q}$ -linearen Automorphismus auf  $\mathbb{H}_{\mathbb{F}}$  erweitern, indem man sie auf jeder Komponente von  $z$  anwendet:  $(z_1, z_2, z_3, z_4)' = (z_1', z_2', z_3', z_4')$ .

**Bemerkung 4.1.3.** Für  $x, y \in \mathbb{H}_{\mathbb{F}}$  ist  $x \cdot y \in \mathbb{F}$  und kann somit in der Form  $(a + \tau b)$  mit  $a, b \in \mathbb{Q}$  geschrieben werden.

Wir benötigen aber noch eine zweite Bilinearform auf  $\mathbb{H}_{\mathbb{F}}$  mit Werten in  $\mathbb{Q}$ .

**Definiton 4.1.4.** Wie eben in 4.1.3 bemerkt wurde, ist  $x \cdot y$  von der Form  $a + \tau b$ . Wir definieren ein zweites Skalarprodukt  $(x \cdot y)_{\tau}$ , die sogenannte *rationale Form* bezüglich  $\tau$ :

$$(x \cdot y)_{\tau} = a \text{ wenn } x \cdot y = a + \tau b.$$

Es lässt sich leicht zeigen, dass dies eine positiv definite, symmetrische Bilinearform ist. Bilinearität und Symmetrie folgen direkt aus der Bilinearität und Symmetrie des Standard-Skalarproduktes. Die positive Definitheit ist auch schnell gezeigt:

Sei  $z \in \mathbb{H}_{\mathbb{F}}$ ,  $z = (z_1, z_2, z_3, z_4) = (x_1 + y_1\tau, x_2 + y_2\tau, x_3 + y_3\tau, x_4 + y_4\tau)$  mit  $x_i, y_i \in \mathbb{Q}$ . Dann ist:

$$\begin{aligned} (z \cdot z)_{\tau} &= \left( (x_1 + y_1\tau)^2 + (x_2 + y_2\tau)^2 + (x_3 + y_3\tau)^2 + (x_4 + y_4\tau)^2 \right)_{\tau} \\ &\stackrel{\tau^2=1+\tau}{=} \left( x_1^2 + x_2^2 + x_3^2 + x_4^2 + y_1^2 + y_2^2 + y_3^2 + y_4^2 \right. \\ &\quad \left. + (2(x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4) + y_1^2 + y_2^2 + y_3^2 + y_4^2) \tau \right)_{\tau} \\ &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + y_1^2 + y_2^2 + y_3^2 + y_4^2 \\ &\geq 0. \end{aligned}$$

Offensichtlich ist  $(z \cdot z)_{\tau} = 0$  genau dann, wenn  $z = 0$  ist. Genauso kann man die rationale Form bezüglich  $\sigma$  definieren.

## 4.2 Die Ikosiane

In diesem Abschnitt möchte ich Ikosiane definieren, die die Ecken eines 600-Zellers bilden und einen Weg aufzeigen, diese mit einem  $E_8$ -Gitter zu identifizieren.

**Definiton 4.2.1.** Die 120 Quaternionen

$$\begin{aligned}
 & (\pm 1, 0, 0, 0) && \text{und alle Permutationen} \\
 & \frac{1}{2} (\pm 1, \pm 1, \pm 1, \pm 1) \\
 & \frac{1}{2} (0, \pm 1, \pm \sigma, \pm \tau) && \text{und alle geraden Permutationen}
 \end{aligned}$$

mit Quaternionen-Norm 1 heißen *Einheits-Ikosiane*. Sie bilden unter der Standard-Quaternionen-Multiplikation eine endliche Gruppe  $\mathcal{I}$  der Ordnung 120. Der Ring der Ikosiane  $\mathfrak{I}$  ist die Menge aller endlichen Summen von Elementen aus der Ikosianen-Gruppe.

**Bemerkung 4.2.2.** Ein typisches Ikosian liegt in  $\mathbb{H}_{\mathbb{F}}$ . Die Einheits-Ikosiane sind genau die Ikosiane mit Quaternionen-Norm 1.

Nun wollen wir einen isometrischen Isomorphismus zwischen dem  $\mathbb{Q}$ -Spann des Wurzelgitters von  $E_8$  ( $=: U$ ) und  $\mathbb{H}_{\mathbb{F}}$  angeben, unter dem die 240 Wurzeln von  $E_8$  auf die 120 Einheitsikosiane und deren  $\tau$ -Vielfachen abgebildet werden. Hierbei verwenden wir das Standardskalarprodukt des  $\mathbb{R}^8$   $(- | -)$  auf  $U$  und  $2(- \cdot -)_{\tau}$  auf  $\mathbb{H}_{\mathbb{F}}$ :

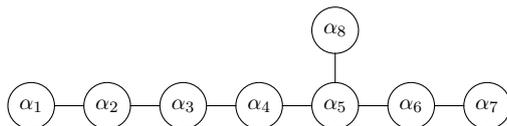
**Satz 4.2.3.** *Es sei  $R$  das Wurzelsystem von  $E_8$  und  $\mathcal{I}$  die Gruppe der Ikosiane wie in 4.2.1 definiert. Dann gibt es einen isometrischen Isomorphismus  $\pi_{\parallel}$  mit:*

$$\pi_{\parallel}: R \rightarrow \mathcal{I} \cup \tau\mathcal{I} \text{ und } (\alpha | \beta) = 2(\pi_{\parallel}(\alpha) \cdot \pi_{\parallel}(\beta))_{\tau}.$$

*Beweis.* Es sei

$$S = \{\alpha_1, \dots, \alpha_8\}$$

eine Menge von einfachen Wurzeln von  $E_8$  mit Dynkin-Diagramm



Dann ist  $\pi_{\parallel}$  gegeben durch

$$\begin{aligned}\alpha_1 &\mapsto \frac{1}{2}(-\sigma, -\tau, 0, -1) =: a_1 \\ \alpha_2 &\mapsto \frac{1}{2}(0, -\sigma, -\tau, 1) =: a_2 \\ \alpha_3 &\mapsto \frac{1}{2}(0, 1, -\sigma, -\tau) =: a_3 \\ \alpha_8 &\mapsto \frac{1}{2}(0, -1, -\sigma, \tau) =: a_4 \\ \alpha_4 &\mapsto \tau a_4 = \frac{1}{2}(0, -\tau, 1, \tau^2) \\ \alpha_5 &\mapsto \tau a_3 = \frac{1}{2}(0, \tau, 1, -\tau^2) \\ \alpha_6 &\mapsto \tau a_2 = \frac{1}{2}(0, 1, -\tau^2, \tau) \\ \alpha_7 &\mapsto \tau a_1 = \frac{1}{2}(1, -\tau^2, 0, -\tau)\end{aligned}$$

Die Gleichheit  $(\alpha \mid \beta) = 2(\pi_{\parallel}(\alpha) \cdot \pi_{\parallel}(\beta))_{\tau}$  lässt sich leicht mit Maple nachprüfen.  $\square$

Trotzdem sollten wir die Gleichheit anhand von einem Beispiel nachvollziehen:

**Beispiel 4.2.4.**

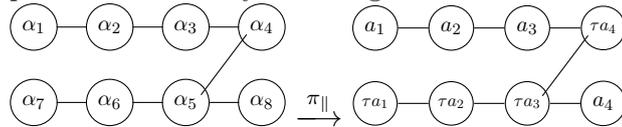
$$(\alpha_1 \mid \alpha_2) = -1$$

und

$$\begin{aligned}2(\pi_{\parallel}(\alpha_1) \cdot \pi_{\parallel}(\alpha_2))_{\tau} &= 2 \cdot \frac{1}{4}((-\sigma, -\tau, 0, -1) \cdot (0, -\sigma, -\tau, 1))_{\tau} \\ &= \frac{1}{2} \left( \underbrace{\sigma\tau}_{-1} - 1 \right)_{\tau} = -1.\end{aligned}$$

Genauso kann man einen isometrischen Isomorphismus  $\pi_{\perp}$  definieren, indem man in der obigen Betrachtung  $\tau$  durch  $\sigma$  ersetzt.

$\pi_{\parallel}$  bildet also vier einfache Wurzeln von  $E_8$  auf bestimmte Ikosiane und die anderen vier auf  $\tau$ -Vielfache derselben ab. Veranschaulichen wir, was hier passiert an dem Dynkin-Diagramm



Wir benötigen noch eine  $\mathbb{Q}$ -lineare Abbildung  $T$  in  $U$ , die diese Multiplikation mit  $\tau$  schon in  $U$  simuliert. Es soll also gelten:

$$T : U \rightarrow U \text{ mit } \pi_{\parallel}(Tx) = \tau\pi_{\parallel}(x) \text{ für alle } x \in U.$$

Wenn  $\pi_{\parallel}(\alpha_i) = \tau\pi_{\parallel}(\alpha_j)$  gilt, dann ist

$$\pi_{\parallel}(T\alpha_i) = \tau\pi_{\parallel}(\alpha_i) = \tau^2\pi_{\parallel}(\alpha_j) = \pi_{\parallel}(\alpha_j) + \tau\pi_{\parallel}(\alpha_j) = \pi_{\parallel}(\alpha_j) + \pi_{\parallel}(\alpha_i).$$

Wir setzen also:

$$\begin{array}{ll} T\alpha_1 = \alpha_7 & T\alpha_7 = \alpha_1 + \alpha_7 \\ T\alpha_2 = \alpha_6 & T\alpha_6 = \alpha_2 + \alpha_6 \\ T\alpha_3 = \alpha_5 & T\alpha_5 = \alpha_3 + \alpha_5 \\ T\alpha_8 = \alpha_4 & T\alpha_4 = \alpha_8 + \alpha_4. \end{array}$$

Außerdem gilt die folgende Bemerkung:

**Bemerkung 4.2.5.** Wenn für  $x \in U$  gilt, dass  $\text{QN}(\pi_{\parallel}(x)) \in \mathbb{Q}$ , dann ist

$$(x | Tx) = 2 \cdot (\pi_{\parallel}(x) \cdot \tau\pi_{\parallel}(x))_{\tau} = 2 \cdot (\tau \text{QN}(\pi_{\parallel}(x)))_{\tau} = 0$$

und

**Bemerkung 4.2.6.** Für  $x \in R$  mit  $\pi_{\parallel}(x) \in \mathcal{I}$  ist  $\pi_{\parallel}(Tx) \in \tau\mathcal{I}$  und es gilt:

$$(Tx | Tx) = 2 (\tau\pi_{\parallel}(x) \cdot \tau\pi_{\parallel}(x))_{\tau} = 2 \left( \tau^2 \underbrace{(x \cdot x)}_1 \right)_{\tau} \stackrel{\tau^2=1+\tau}{=} 2$$

### 4.3 Die Coxeter Gruppen $H_4$ und $H_3$

Nun werden wir die Coxeter Gruppe  $H_4$  einführen.

Die Weyl-Gruppe von  $E_8$  ist

$$W(E_8) = \langle s_1, \dots, s_8 \mid (s_i s_j)^{m_{ij}} = 1 \rangle.$$

Hierbei ist  $M = (m_{ij})$  durch die Matrix

$$M = (m_{ij}) = \begin{pmatrix} 1 & 3 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 1 & 3 & 2 & 2 & 2 & 2 \\ 2 & 2 & 3 & 1 & 3 & 2 & 2 & 2 \\ 2 & 2 & 2 & 3 & 1 & 3 & 2 & 3 \\ 2 & 2 & 2 & 2 & 3 & 1 & 3 & 2 \\ 2 & 2 & 2 & 2 & 2 & 3 & 1 & 2 \\ 2 & 2 & 2 & 2 & 3 & 2 & 2 & 1 \end{pmatrix} \quad (4.1)$$

gegeben.

**Definiton 4.3.1.** Die Coxeter Gruppe  $H_4$  ist abstrakt über die Repräsentation

$$H_4 = \langle S_1, S_2, S_3, S_4 \mid (S_i S_j)^{m_{ij}} = 1 \rangle$$

gegeben. Hier ist nun

$$M = (m_{ij}) = \begin{pmatrix} 1 & 3 & 2 & 2 \\ 3 & 1 & 3 & 2 \\ 2 & 3 & 1 & 5 \\ 2 & 2 & 5 & 1 \end{pmatrix}.$$

**Bemerkung 4.3.2.** Es sei  $x \in \mathbb{H}$  mit  $\text{QN}(x) = 1$ . Dann ist die Spiegelung an der Hyperebene orthogonal zu  $x$  (im  $\mathbb{R}^4 \simeq \mathbb{H}$ ):

$$S_x : v \mapsto v - \frac{2v \cdot x}{x \cdot x} x = v - \frac{v\bar{x} + x\bar{v}}{x\bar{x}} x = v - v\bar{x}x - x\bar{v}x = -x\bar{v}x.$$

Sei nun  $x \in R$  eine Wurzel, die wenn sie in  $\mathbb{H}_{\mathbb{F}}$  betrachtet wird in  $\mathcal{I}$  liegt. Dann liegt die mit  $\text{T}x \in R$  identifizierte Wurzel  $\tau x$  in  $\tau\mathcal{I}$  und es gilt nach 4.2.5  $(\text{T}x \mid x) = 0$ .

**Lemma 4.3.3.** *Es sei  $x \in \mathcal{I}$  und  $s_x$  und  $s_{\text{T}x}$  die Spiegelungen in  $U \simeq \mathbb{H}_{\mathbb{F}}$  bezüglich  $(- \mid -)$ . Dann gilt mit unserer Identifikation von  $U$  und  $\mathbb{H}_{\mathbb{F}}$  über  $\pi_{\parallel}$*

$$s_{\text{T}x} s_x = S_x$$

*Beweis.* Es ist

$$\begin{aligned} s_{\text{T}x} s_x v &= s_{\text{T}x} (v - (v \mid x) x) = v - (v \mid x) x - \frac{2(v - (v \mid x) x \mid \text{T}x)}{(\text{T}x \mid \text{T}x)} \text{T}x \\ &= v - (v \mid x) x - (v \mid \text{T}x) \text{T}x = v - 2(v \cdot x)_{\tau} x - 2(v \cdot \tau x)_{\tau} \tau x. \end{aligned}$$

Setze nun  $v \cdot x = p + \tau q$  mit  $p, q \in \mathbb{Q}$ , dann ist  $(v \cdot x)_{\tau} = p$  und  $(v \cdot \tau x)_{\tau} = q$  und damit

$$s_{\text{T}x} s_x v = v - 2(p + \tau q) x = v - 2(v \cdot x) x = S_x v.$$

□

Setze nun

$$S_1 := s_1 s_7 \quad S_2 := s_2 s_6 \quad S_3 := s_3 s_5 \quad S_4 := s_4 s_8.$$

**Satz 4.3.4.**  $S_1, \dots, S_4$  erfüllen die Coxeter Gleichungen für  $H_4$ .

*Beweis.* Alle Gleichungen bis auf eine  $((S_3S_4)^5 = 1)$  folgen direkt aus den Coxeter Gleichungen für  $s_1, \dots, s_8$ . So ist zum Beispiel

$$S_1S_1 = s_1s_7s_1s_7 = (s_1s_7)^2 = 1.$$

Da mit 4.1  $s_1, s_7, s_3, s_5$  alle paarweise kommutieren gilt auch

$$(S_1S_3)^2 = (s_1s_7s_3s_5)^2 = 1.$$

Genauso gilt auch, dass sowohl  $s_1$ , als auch  $s_2$  mit  $s_6$  und  $s_7$  kommutiert, daher ist:

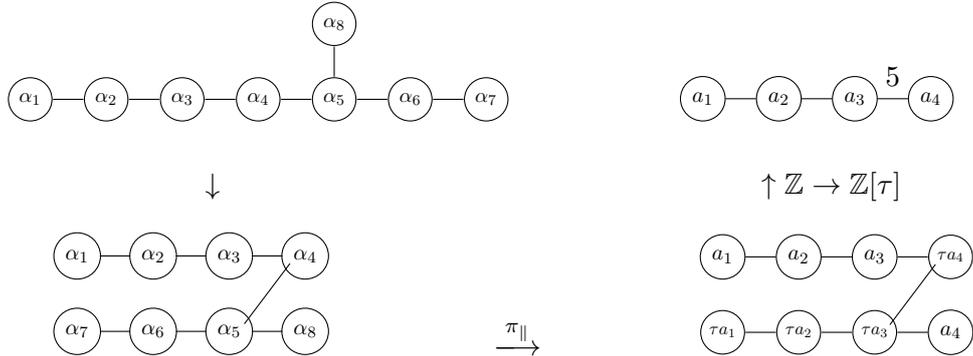
$$(S_1S_2)^3 = (s_1s_7s_2s_6)^3 = (s_1s_2)^3 (s_7s_6)^3 = 1.$$

Die einzige Gleichung, die man so nicht erhält, ist

$$(S_3S_4)^5 = (s_3s_5s_4s_8)^5 = 1.$$

Aber da  $s_3s_5s_4s_8$  ein Coxeter Element der Weyl Gruppe von der Lie Gruppe vom Typ  $A_4$  ist, hat dieses die Ordnung 5 und erfüllt daher die Gleichung.  $\square$

Veranschaulichen wir wieder (wie in [18], Abschnitt II B) mit den Coxeter-Dynkin-Diagrammen was passiert ist:



Diesen Vorgang nennt man Faltung.

Nun können wir die Verbindung zwischen den Ikosianen und den Wurzelsystemen vom Typ  $H_4$  herstellen. Für jedes  $x \in \mathcal{I}$  ist nämlich  $S_x : \mathcal{I} \rightarrow \mathcal{I}$ . Daher ist  $\mathcal{I}$  ein nicht-kristallographisches Wurzelsystem vom Typ  $H_4$ . Somit ist  $\{a_1, a_2, a_3, a_4\}$  eine Basis von  $\mathcal{I}$  sodass jedes Element von  $\mathcal{I}$  eindeutig als nicht negative oder nicht positive Linearkombination von  $a_1, a_2, a_3$  und  $a_4$  dargestellt werden kann.

**Definiton 4.3.5.** Die Untergruppe von  $H_4$ , die von  $S_2, S_3$  und  $S_4$  erzeugt wird ist  $H_3$ :

$$H_3 \simeq \langle S_2, S_3, S_4 \rangle.$$

**Lemma 4.3.6.** Die Gruppe  $\langle S_2, S_3, S_4 \rangle$  stabilisiert die reinen Quaternionen  $\mathbb{H}_0$ .

*Beweis.* Ist  $x \in \mathcal{I}$  ein reiner Quaternion, gilt also  $\bar{x} = -x$  und, da alle Elemente aus  $\mathcal{I}$  Einheitsquaternionen sind, also  $x\bar{x} = 1$  und somit  $x^{-1} = \bar{x}$ , so gilt mit 4.3.2

$$S_x : v \rightarrow -x\bar{v}x = x\bar{v}(-x) = x\bar{v}\bar{x} = x\bar{v}x^{-1}$$

Ist  $v$  nun ebenfalls ein reiner Quaternion, so ist  $S_x(v)$  auch ein reiner Quaternion und die Behauptung ist bewiesen.  $\square$

## 4.4 Eine dreidimensionale Projektion von $E_8$ nach D. Richter

Nun haben wir schon eine Isomorphie zwischen den Wurzeln von  $E_8$  und den Ikosianen beziehungsweise  $H_4$  aufgezeigt, haben den Sprung vom Achtdimensionalen ins Vierdimensionale also schon geschafft. Nun gibt es verschiedene Möglichkeiten von hier in das leichter darstellbare Dreidimensionale zu kommen. In diesem Abschnitt möchte ich eine hiervon aufzeigen, bei der das Bild durch einige Drehungen und Spiegelungen wieder in sich selbst abgebildet werden kann. Hierbei werde ich nach [8] vorgehen.

Wir sind bei Ikosianen der Form

$$a + ib + jc + kd$$

angelangt, wo  $a, b, c, d \in \mathbb{F}$ . Betrachten wir nun die Projektion

$$\begin{aligned} \pi_2 : \mathbb{R}^4 &\rightarrow \mathbb{R}^3 \\ (a, b, c, d) &\mapsto (b, c, d). \end{aligned}$$

Mit Hilfe eines Maple Programms kann man diese Projektion durchführen. Im Anhang befindet sich ein entsprechendes Programm. Die so erhaltenen Punkte werden wieder, wie schon im zweidimensionalen Fall, verbunden falls die zugehörigen Wurzeln minimalen Abstand haben.

### 4.4.1 Das Bild

Die Punkte liegen auf dem Rand von sieben konzentrischen Kugeln um den Ursprung und ein Punkt liegt auf dem Ursprung selbst. Wie viele Punkte es mit welchem Abstand zum Ursprung gibt, zeigt die folgende Tabelle. Das  $()^P$  an den Punkten bedeutet, dass alle geraden Permutationen genommen werden müssen.

Abstand zum Ursprung	Punkte	Anzahl
0	(0, 0, 0)	1
$1/2\sqrt{2 + \sigma}$	$1/2 (0, \pm\sigma, \pm 1)^P$	12
$1/2\sqrt{3}$	$1/2 (\pm\tau, \pm\sigma, 0)^P$	12
	$1/2 (\pm 1, \pm 1, \pm 1)$	8
$1/2\sqrt{2 + \tau}$	$1/2 (\pm 1, \pm\tau, 0)^P$	12
1	$(\pm 1, 0, 0)^P$	6
	$1/2 (\pm 1, \pm\sigma, \pm\tau)^P$	24
$1/2\sqrt{3}\tau$	$1/2 (\pm\tau, \pm\tau, \pm\tau)$	8
	$1/2 (\pm\tau^2, \pm 1, 0)^P$	12
$1/2\sqrt{3 + 4\tau}$	$1/2 (\pm\tau, \pm\tau^2, 0)^P$	12
$\tau$	$(\pm\tau, 0, 0)^P$	6
	$1/2 (\pm\tau^2, \pm\tau, \pm 1)^P$	24
		137

Diese werden 2828 Kanten verbunden.

#### 4.4.2 Symmetrie

Die Symmetriegruppe des erhaltenen Modells ist natürlich eine Untergruppe von  $O(3)$ , da es nur endlich viele Punkte gibt. Wir benötigen den folgenden Satz aus [16]:

**Satz 4.4.1.** *Es sei  $X \subset \mathbb{R}^3$  eine endliche Teilmenge, deren Symmetriegruppe  $S(X) \subset O(3)$  die zentrale Symmetrie  $J : x \mapsto -x$ , enthält. Dann ist  $S(X)$  isomorph zum direkten Produkt  $H \times \langle J \rangle$ , wobei  $H \subset SO(3)$  eine Untergruppe der Drehsymmetrien von  $X$  ist.*

*Beweis.* Zunächst zeigen wir die Isomorphie zwischen  $O(3)$  und  $SO(3) \times \{J\}$ . Diese ist durch die Abbildung

$$\Theta : O(3) \rightarrow SO(3) \times \langle J \rangle, \text{ mit } \Theta(A) = (\det(A) \cdot A, \det(A))$$

gegeben. Es ist also  $\Theta(A, 1)$  für  $A \in SO(3)$  und  $\Theta(A) = (-A, -1)$  für  $A \in O(3) \setminus SO(3)$ . Diese Abbildung ist ein Homomorphismus:

$$\begin{aligned} \Theta(AB) &= (\det(AB) AB, \det(AB)) = (\det(A) A \det(B) B, \det(A) \det(B)) \\ &= \Theta(A) \Theta(B). \end{aligned}$$

Offensichtlich ist diese Abbildung sowohl injektiv als auch surjektiv. Bemerkte, dass  $J$  unter  $\Theta$  auf  $(I, -1)$  abgebildet wird. Somit wird  $S(X)$  auf eine Untergruppe von  $SO(3) \times \langle J \rangle$  abgebildet. Diese Untergruppe ist  $H \times \langle J \rangle$ .  $\square$

Nun zu den endlichen Untergruppen von  $SO(3)$  (nach [17], 5.7 Klassifikations-Theorem):

**Theorem 4.4.2.** Jede endliche Untergruppe von  $SO(3)$  ist konjugiert zu einer der folgenden Untergruppe:

1. der zyklischen Gruppe  $C_n$  mit Ordnung  $n$
2. der Diedergruppe  $D_n$  mit Ordnung  $2n$
3. der Tetraedergruppe  $A_4$  mit Ordnung 12
4. der Würfelgruppe  $S_4$  mit Ordnung 24
5. der Ikosaedergruppe  $A_5$  mit Ordnung 60.

Da der Beweis sehr umfangreich ist, verweise ich hierfür auf [17], Kapitel 5.9.

Mit dem Programm im Anhang werden die Drehsymmetrien unseres Modells gezählt. Es sind 60. Die Diedergruppe und die zyklische Gruppe können es nicht sein, da diese Drehungen in der Ebene beschreiben. Daher muss die Drehsymmetriegruppe die  $A_5$  sein und somit ist die ganze Symmetriegruppe  $A_5 \times \langle J \rangle$ , welches die ganze Symmetriegruppe des Ikosaeders, also nach [1], Kapitel 2.13 die Gruppe vom Typ  $H_3$  ist. Diese hat Ordnung 120.

## 4.5 Eine dreidimensionales Projektion von $E_8$ , nach G. Moxness

Nun möchte ich noch eine andere Möglichkeit, ein dreidimensionales Bild von  $E_8$  zu zeichnen, aufzeigen. Nach [9] faltet die Matrix

$$H_{4fold} := \begin{pmatrix} \tau & 0 & 0 & 0 & \sigma^2 & 0 & 0 & 0 \\ 0 & -\sigma & 1 & 0 & 0 & \sigma & 1 & 0 \\ 0 & 1 & 0 & -\sigma & 0 & 1 & 0 & \sigma \\ 0 & 0 & -\sigma & 1 & 0 & 0 & \sigma & 1 \\ \sigma^2 & 0 & 0 & 0 & \tau & 0 & 0 & 0 \\ 0 & \sigma & 1 & 0 & 0 & -\sigma & 1 & 0 \\ 0 & 1 & 0 & \sigma & 0 & 1 & 0 & -\sigma \\ 0 & 0 & \sigma & 1 & 0 & 0 & -\sigma & 1 \end{pmatrix}$$

$E_8$  nach  $H_4 \cup \tau H_4$ . Man benötigt eigentlich nur die ersten vier oder letzten vier Zeilen, da wir die Matrix später aber invertieren wollen betrachten wir sie als  $(8 \times 8)$ -Matrix. Für die Projektion von  $E_8$  in den dreidimensionalen

Raum benötigen wir drei Basisvektoren. Dafür definieren wir zunächst

$$x := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2\tau \sin \frac{2\pi}{60} \\ 0 \\ 1 \\ 0 \end{pmatrix}, y := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2\tau \sin \frac{2\pi}{30} \\ 0 \\ 2 \sin \frac{2\pi}{15} \end{pmatrix}, z := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2\tau \sin \frac{2\pi}{60} \\ 0 \end{pmatrix}.$$

Hierbei sind  $x$  und  $y$  die Basisvektoren der van Oss Projektion des 600-Zellers in den  $\mathbb{R}^3$  (Siehe [24]). Fassen wir also kurz zusammen: Wir haben

$$E_8 \xrightarrow{H4_{fold}} H_4 \cup \tau H_4 \xrightarrow{pr_{x,y,z}} \mathbb{R}^3$$

und möchten drei Basisvektoren  $X$ ,  $Y$  und  $Z$  mit

$$E_8 \xrightarrow{pr_{X,Y,Z}} \mathbb{R}^3$$

Diese drei Vektoren berechnen wir also aus der Inversen von  $H4_{fold}$  und  $x$ ,  $y$  und  $z$ :

$$\begin{aligned} X &:= H4_{fold}^{-1} \cdot x \\ Y &:= H4_{fold}^{-1} \cdot y \\ Z &:= H4_{fold}^{-1} \cdot z. \end{aligned}$$

Ein Programm, das diese Projektion realisiert befindet sich im Anhang. Das Modell besteht dann aus 240 Punkten und 6720 Kanten. Die Punkte haben unterschiedliche Abstände zum Ursprung, daher erhält keine Drehung dieses Bild. Dennoch finden sich andere Symmetrien wieder: Betrachtet man das Bild vom richtigen Standpunkt aus ergeben sich erstaunliche Beobachtungen, die wir im Folgenden noch aufführen möchten. Bei jedem der folgenden Bilder ist angegeben, wie man die Winkel  $\theta$ ,  $\phi$  und  $\psi$  in Maple setzen muss, um diesen Blickwinkel zu erhalten. Man erhält aus zwei Betrachtungswinkeln einen Blick auf die van Oss Projektion des 600-Zellers. Eine hiervon ist natürlich, wenn man von oben auf das Modell schaut, also  $z = 0$  setzt:

Abbildung 4.1: E8, Blick senkrecht auf die  $xy$ -Ebene,  $\theta = \phi = \psi = 0$

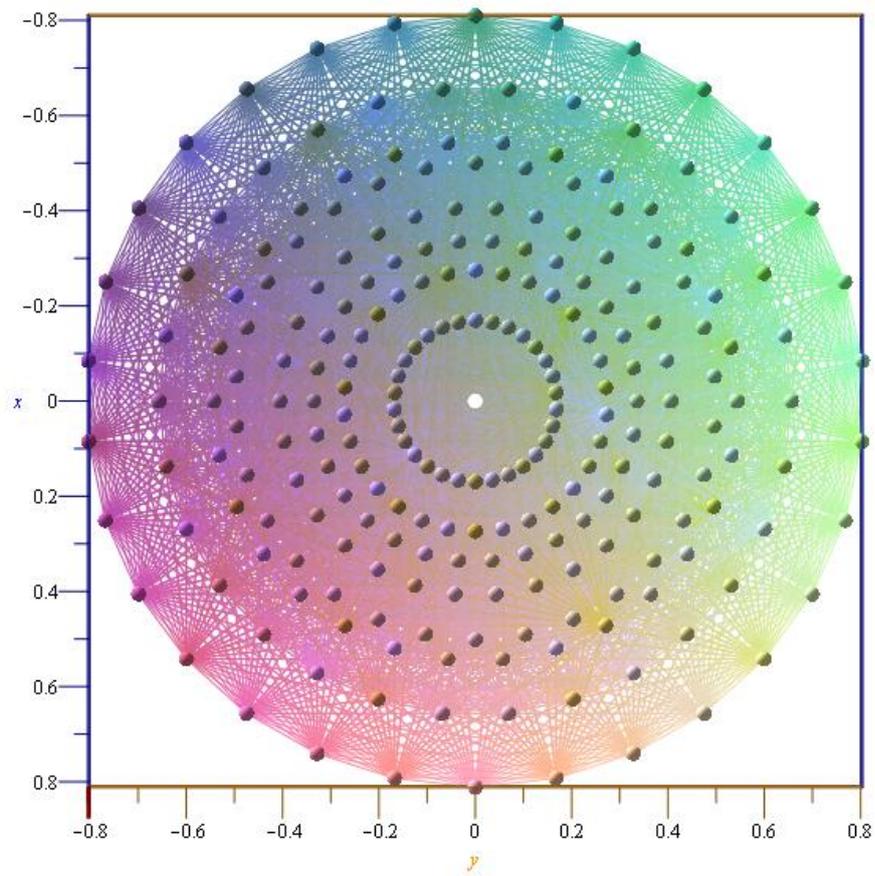
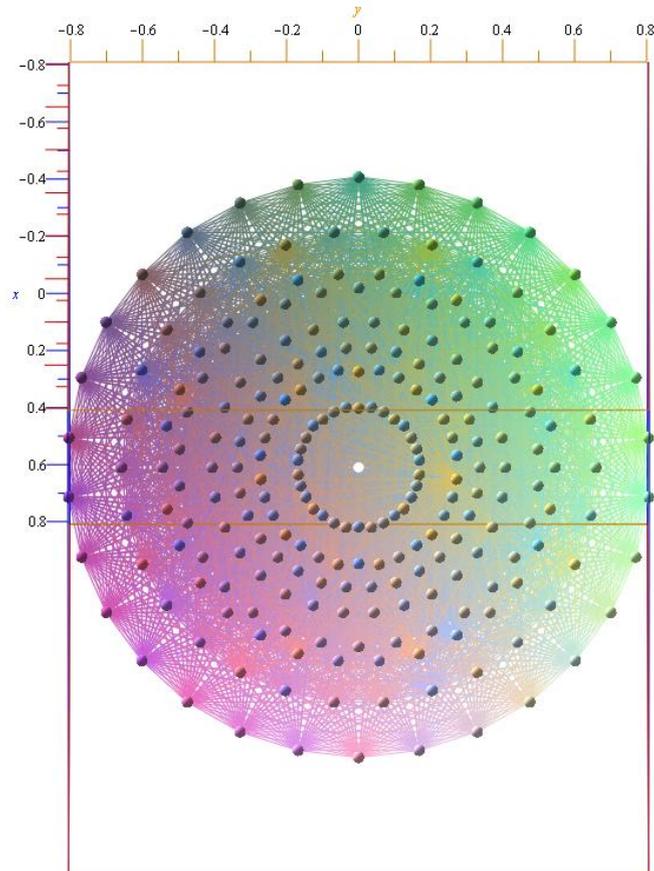
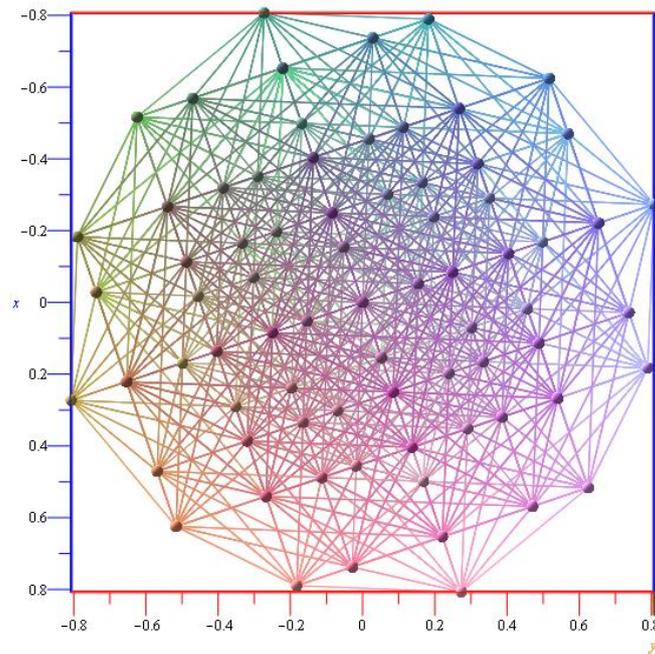


Abbildung 4.2: E8,  $\theta = 0, \phi = -37, \psi = 0$



Schaut man senkrecht auf die  $x$ - $z$ -Ebene, so sieht man das Bild, das eine orthonormale Projektion von  $H_4 \cup \tau H_4$  in den zweidimensionalen Raum hervorbringt:

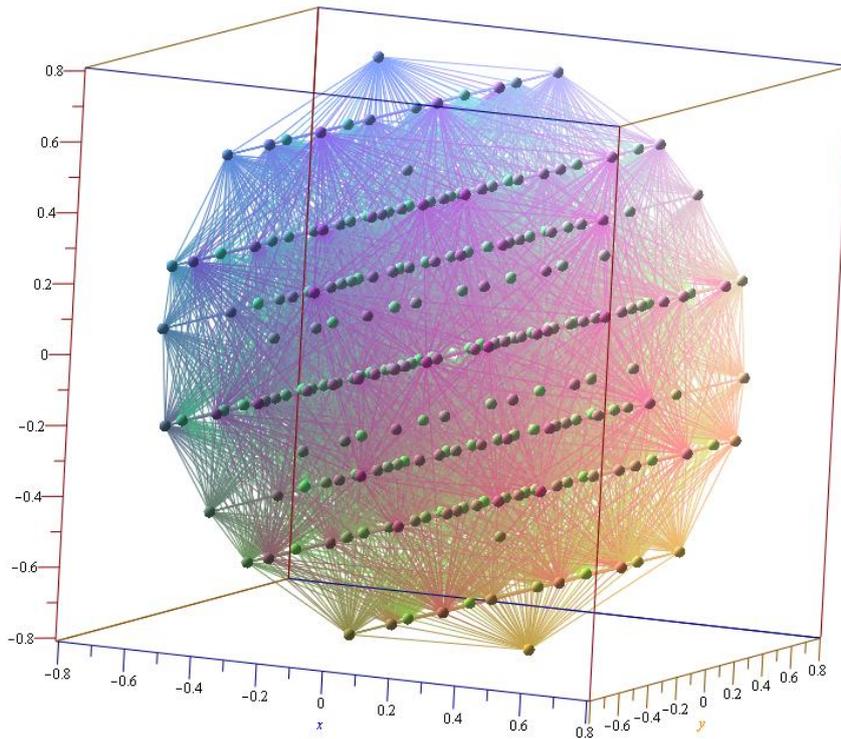
Abbildung 4.3: E8, Blick auf die  $xz$ -Ebene,  $\theta = \phi = 0, \psi = 90$



Außerdem habe ich noch einen Betrachtungswinkel gefunden, von dem aus alle Ecken auf 13 Geraden liegen. Eine Basis für den zweidimensionalen Raum, in den projiziert wird kann mit dem Programm im Anhang berechnet werden. Diese sind:

$$\left\langle \left( \begin{array}{c} -0,320425910085493770642525991446 \\ 0 \\ 0,947273580411637502498655156571 \end{array} \right), \left( \begin{array}{c} 0,492998564369845025811812853468 \\ 0,853898561547087212433920261235 \\ 0,166762292251836963140634345441 \end{array} \right) \right\rangle$$

Abbildung 4.4: E8,  $\theta = -46$ ,  $\phi = 81$ ,  $\psi = 20$



## Anhang A

# Programm zur Projektion der Wurzeln eines Wurzelsystems in seine Coxeter Ebene

Für alle Programme wird die Maple Pakete “The coxeter and weyl Packages“ von John Stembridge benötigt. Diese findet man auf [www.math.lsa.umich.edu/jrs/maple.html#coxeter](http://www.math.lsa.umich.edu/jrs/maple.html#coxeter). Mit dem folgenden Maple Programm kann man sich die Projektion eines Wurzelsystems in seine Coxeter Ebene ausgeben lassen. Möchte man ein anderes Wurzelsystem projektieren, so muss man R und eis entsprechend ändern. Wenn ein Wurzelsystem in die Coxeter Ebene eines anderen Wurzelsystems projiziert werden, so muss in R dasjenige stehen, dessen Coxeter Ebene berechnet werden soll und in der Definition von allRoots muss statt R dann das Wurzelsystem stehen, dass projiziert werden soll. Im Programm wird dann zunächst das Coxeter Element berechnet. Dazu werden alle einfachen Spiegelungen berechnet und dann alle multipliziert. Danach werden die Eigenwerte und zugehörigen Eigenvektoren des Coxeter Elements berechnet und es wird der Eigenvektor zum Eigenwert  $e^{\frac{2\pi i}{h}}$  herausgesucht. Der Real- und Imaginärteil dieses Vektors wird  $\lambda$  bzw.  $\mu$  genannt und eine Orthonormalbasis hieraus berechnet. Nun werden die Wurzeln projiziert und die Ergebnisse sowohl in der Liste projRoots, als auch in einer Liste equivClassprR abgespeichert. In der letzteren werden die Punkte zusammen mit den ursprünglichen Wurzeln, die auf diesen Punkt projiziert werden abgespeichert. Dies wird benötigt um später die Linien leichter berechnen zu können. Schließlich werden noch die Listen distlist (hierin stehen Listen, deren erste Einträge der Abstand eines Punktes vom Ursprung ist und alle weiteren Einträge enthalten alle Punkte, die diesen Abstand haben), dist (enthält nur die Abstände, zum sortieren gedacht) und distlistsort (gleiche Liste wie distlist, nur sortiert) angelegt. Schließlich werden noch die Linien, die gezeichnet werden müssen ermittelt.

```

with(ListTools):
with(plots):
with(coxeter):
with(weyl):
with(LinearAlgebra):

Digits:=30:
epsilon:=0.000000001:

#In die Coxeter-Ebene von R wird projiziert.
#Außerdem müssen in der Liste eis die Basisvektoren des  $R^n$  stehen
R:= E8:
eis:=[e1,e2,e3,e4,e5,e6,e7,e8]:

#R oben kann eine Unter algebra des R in der folgenden Zeile sein.
#So kann man sich die Projektion in die Coxeter Ebene einer Unter algebra ausgeben lassen.
allRoots:=[op(pos_roots(R)), -op(pos_roots(R))]:

S:=base(R):
veis:=Vector(eis):
n:=numelems(eis):
h:=cox_number(R):

#cox wird die Matrix des Coxeter-Elements
cox:=Matrix(n, shape=identity):

#In diese Liste werden alle Spiegelungsmatrizen gespeichert
refls:=[]:

#eine Variablenliste, die es zu lösen gilt
listofVar:=[]:
A:=Matrix(n,x):
for i from 1 to n do
  for j from 1 to n do
    listofVar:=[op(listofVar),A(i,j)]:
  od:
od:

#Wir durchlaufen alle Basiselemente und Berechnen die Matrix für die Spiegelung an diesem Element
for i from 1 to numelems(S) do
  Ref:=Matrix(n):
  eqs:=[]:
  #durchläuft alle Basiselemente zur Berechnung der Matrix
  #(Matrix*Basiselement=Spiegelung(Basiselement))
  for j from 1 to numelems(S) do
    v:=Multiply(A,Vector(root_coords(S[j],eis))):
    ref:=root_coords(reflect(S[i],S[j]),eis):
    for k from 1 to n do
      eqs:=[op(eqs), v[k]=ref[k]]:
    od:
  od:
  sol:=solve(eqs, listofVar):
  for j from 1 to n do
    for k from 1 to n do
      if member(A(j,k)=A(j,k), sol[1]) then
        sol:=solve([op(sol[1]),A(j,k)=0],listofVar):
      fi:
    od:
  od:
  Ref:=subs(sol[1],A):

```

```

    refs:=[op(refs), Ref]:
od:

#Berechnung des Coxeter Elements:
for i from 1 to numelems(refs) do
    cox:=Multiply(cox,refs[i]):
od:

reew:=evalf(cos(2*Pi/h));
imew:=evalf(sin(2*Pi/h));

(ews,evs):=Eigenvectors(cox):

found:=false:
for i from 1 to numelems(ews) do

    if abs(evalf(Re(evalf(ews[i]))-reew))<epsilon and abs(evalf(Im(evalf(ews[i]))-imew))<epsilon then
        found:=true:
        break:
    fi:
od:

if found then
    ev:=Column(evs, i):
else print(„ERROR“):
fi:

lambda:=Vector(Re(evalf(ev))):
mue:=Vector(Im(evalf(ev))):

mue:=mue-DotProduct(mue,lambda)/(DotProduct(lambda,lambda))*lambda:
lambda:=lambda/norm(lambda,2):
mue:=mue/norm(mue,2):
B:=[DotProduct(lambda,veis,conjugate=false), DotProduct(mue,veis,conjugate=false)]:

projRoots:=[]:
#an erster Stelle jeder Liste steht der projizierte Vektor, danach alle Wurzeln, die auf diesen projiziert
#werden
equivClassprR:=[]:

for i in allRoots do
    p:=[iproduct(i,B[1]), iproduct(i,B[2])]:

    equivClassFound:=false:
    for j from 1 to nops(equivClassprR) while equivClassFound=false do
        if abs(evalf(p[1]-equivClassprR[j,1,1]))<epsilon and abs(evalf(p[2]-equivClassprR[j,1,2]))<epsilon
            then
                equivClassprR[j]:=[op(equivClassprR[j]), i]:
                equivClassFound:=true:
            fi:
        od:
        if equivClassFound=false then
            equivClassprR:=[op(equivClassprR), [p , i]]:
            projRoots:=[op(projRoots), p]:
        fi:
    od:

punkte:=pointplot(projRoots):

distlist:=[]:

```

```

for i in projRoots do
  normi:=norm(Vector(i),2):
  found:=false:
  for j from 1 to nops(distlist) do
    if abs(evalf(normi-distlist[j][1]))<epsilon then
      distlist[j]:=[op(distlist[j]), i]:
      found:=true:
      break:
    fi:
  od:
  if found=false then
    distlist:=[op(distlist), [normi, i]]
  fi:
od:

```

```

dist:=[]:
for i from 1 to nops(distlist) do
  dist:=[op(dist), distlist[i][1]]:
od:
dist:=sort(evalf(dist)):
distlistsort:=[]:
for i from nops(dist) by -1 to 1 do
  for j from 1 to nops(distlist) do
    if abs(evalf(distlist[j][1]-dist[i]))<epsilon then
      distlistsort:=[op(distlistsort), distlist[j]]:
      break:
    fi:
  od:
od:

```

```

colors:=["DarkRed", "DarkGreen", "DarkBlue", "Red", "Green", "Blue", "Violet", "LightGreen", "Black"]:

```

```

lines:=[]:
tl:=[]:

```

```

for i from 1 to nops(distlistsort) do
  for j from 2 to nops(distlistsort[i]) do
    for k from i to nops(distlistsort) do
      for m from 2 to nops(distlistsort[k]) do
        for n from 1 to nops(equivClassprR) do
          if abs(evalf(equivClassprR[n][1][1]-distlistsort[i][j][1]))<epsilon
            and abs(evalf(equivClassprR[n][1][2]-distlistsort[i][j][2]))<epsilon then
            break:
          fi:
        od:
        for p from 1 to nops(equivClassprR) do
          if abs(evalf(equivClassprR[p][1][1]-distlistsort[k][m][1]))<epsilon
            and abs(evalf(equivClassprR[p][1][2]-distlistsort[k][m][2]))<epsilon then
            break:
          fi:
        od:
        for q from 2 to nops(equivClassprR[n]) do
          for r from 2 to nops(equivClassprR[p]) do
            if member(equivClassprR[n][q]-equivClassprR[p][r], allRoots) then
              linetoprint:=true:
              lineprinted:=false:
              for z from 1 to nops(tl) do
                if abs(evalf(distlistsort[i][j][1]-tl[z][1][1]))<epsilon

```

```

and abs(evalf(distlistsort[i][j][2]-tl[z][1][2]))<epsilon
and abs(evalf(distlistsort[k][m][1]-tl[z][2][1]))<epsilon
and abs(evalf(distlistsort[k][m][2]-tl[z][2][2]))<epsilon then
  linetoprint:=false:
  break:
fi:
od:
if linetoprint then
  lines:=[op(lines), plottools[line](distlistsort[i][j], distlistsort[k][m],
  color=colors[i],thickness=0), plottools[line](-distlistsort[i][j], -distlistsort[k][m],
  color=colors[i], thickness=0)]:
  tl:=[op(tl), [distlistsort[i][j], distlistsort[k][m]], [distlistsort[k][m], distlistsort[i][j]],
  [-distlistsort[i][j], -distlistsort[k][m]], [-distlistsort[k][m], -distlistsort[i][j]]]:
  lineprinted:=true:
  break:
fi:
fi:
od:
if lineprinted=true then
  break:
fi:
od:
od:
od:
od:
od:
display(punkte, lines, thickness=0);

```

## Anhang B

# Programm zum dreidimensionalen Modell von $E_8$ nach D.Richter

Es wird wieder das Paket Coxeter/Weyl von J.Stembridge benötigt.

```

with(coxeter):
with(weyl):
with(plots):
with(ListTools):
with(plottools):
with(geom3d):
with(LinearAlgebra):
Digits:=30:
epsilon:=0.000000001:
sigma:=1/2*(1-sqrt(5)):
tau:=1/2*(1+sqrt(5)):

sigma:=evalf(sigma):
tau:=evalf(tau):

S:=[e7-e6,e6-e5,e5-e4,e4-e3,e3-e2,e2-e1,1/2*(e1-e2-e3-e4-e5-e6-e7+e8),e1+e2]:
map1:=1/2*Matrix([[ -sigma,0,0,0,0,1,0],
                 [-tau,-sigma,1,-tau,tau,1,-tau^2,-1],
                 [0,-tau,-sigma,1,1,-tau^2,0,-sigma],
                 [-1,1,-tau,tau^2,-tau^2,tau,-tau,tau]]):

R:=[op(pos_roots(S)),op(-pos_roots(S))]:

#Die Liste, in der später die Bildpunkte stehen
proje8:=[]:

#Zählt, wie viele verschiedene Bildpunkte es gibt

#Berechnen der Bildpunkte
for i from 1 to nops(R) do
  p:=Multiply(map1,Vector(root_coords(R[i],S))):
  p:=[p[2],p[3],p[4]]:
  proje8:=[op(proje8),evalf(p)]:

od:

#Hier werden später alle Linien drin stehen
lines:=[]:
#Dieser Counter zählt nur die Linien, die auch gezeichnet werden
counterl:=0:
#Dieser Counter zählt wie viele Wurzeln verbunden werden müssten, wenn jede Wurzel auf einen anderen
#Punkt projiziert wird
cou:=0:
#Hier stehen zur Überprüfung, ob eine Linie schon gezeichnet wurde die Anfang- und Endpunkte aller Linien
drin
lin:=[]:
for i from 1 to nops(R)-1 do
  for j from i+1 to nops(R) do
    linge:=false:
    #Muss eine Linie gezeichnet werden?
    if member(R[i]-R[j], R) then
      linge:=true:
      cou:=cou+1:
    fi:
    if linge=true then
      found:=false:

```

```

#Wurde diese Linie schon einmal gezeichnet oder projektieren beide Wurzeln auf den gleichen
#Punkt?
for k from 1 to nops(lin) do
  if abs(evalf(proje8[i][1]-lin[k][1][1]))<epsilon and abs(evalf(proje8[j][1]-lin[k][2][1]))<epsilon
    and abs(evalf(proje8[i][2]-lin[k][1][2]))<epsilon
    and abs(evalf(proje8[j][2]-lin[k][2][2]))<epsilon
    and abs(evalf(proje8[i][3]-lin[k][1][3]))<epsilon
    and abs(evalf(proje8[j][3]-lin[k][2][3]))<epsilon then
      found:=true:
    elif abs(evalf(proje8[j][1]-lin[k][1][1]))<epsilon and abs(evalf(proje8[i][1]-lin[k][2][1]))<epsilon
      and abs(evalf(proje8[j][2]-lin[k][1][2]))<epsilon
      and abs(evalf(proje8[i][2]-lin[k][2][2]))<epsilon
      and abs(evalf(proje8[j][3]-lin[k][1][3]))<epsilon
      and abs(evalf(proje8[i][3]-lin[k][2][3]))<epsilon then
        found:=true:
    elif abs(evalf(proje8[i][1]-proje8[j][1]))<epsilon and abs(evalf(proje8[i][2]-proje8[j][2]))<epsilon
      and abs(evalf(proje8[i][3]-proje8[j][3]))<epsilon then
        found:=true:
  fi:
od:
#wenn nicht, so wird die Linie gezählt und gezeichnet
if found=false then
  lines:=[op(lines), plottools[line](proje8[i], proje8[j], thickness=0)]:
  counter:=counter+1:
  lin:= [op(lin),[proje8[i],proje8[j]]]:
fi:
fi:
od:
print("Es gibt ", counterl, "von", cou, "Linien");
#distlist soll eine Liste von Listen sein. In jedem Eintrag steht an erster Stelle wie weit die Punkte die an allen
#weiteren Stellen stehen vom Ursprung entfernt sind.
distlist:=[]:

for i in proje8 do
  #Abstand vom aktuellen Bildpunkt berechnen
  normi:=norm(Vector(i),2):

  #Diese Variable soll angeben, ob es für diesen Abstand schon eine Liste gibt
  found:=false:

  #Überprüfen, ob es schon weitere Punkte mit dem gleichen Abstand gibt, wenn ja aktuellen Punkt in die
  #entsprechende Liste dazu schreiben
  for j from 1 to nops(distlist) do
    if evalf(abs(normi-distlist[j][1]))<epsilon then
      distlist[j]:=[op(distlist[j]), i]:
      found:=true:
      break:
    fi:
  od:
  #Wenn es noch keine Liste für diesen Abstand gab, wird eine erstellt und an distlist angehängt
  if found=false then
    distlist:=[op(distlist), [normi, i]]:
  fi:
od:
#dist ist eine Liste in der alle Abstände, die vorkommen stehen. diese wird sortiert.
dist:=[]:
for i from 1 to nops(distlist) do
  dist:=[op(dist), distlist[i][1]]:
od:

```

```
dist:=sort(evalf(dist)):
```

```
#distlistsort ist distlist, bei dem die Listen absteigend nach dem Abstand zum Ursprung sortiert werden.
```

```
distlistsort:=[]:
```

```
for i from nops(dist) by -1 to 1 do
```

```
  for j from 1 to nops(distlist) do
```

```
    if abs(evalf(distlist[j][1]-dist[i]))<epsilon then
```

```
      distlistsort:=[op(distlistsort), distlist[j]]:
```

```
      break:
```

```
    fi:
```

```
  od:
```

```
od:
```

```
#zeichnen der Punkte und Linien
```

```
ursprung:=pointplot3d([0,0,0], color=red):
```

```
points:=pointplot3d(proje8):
```

```
display(lines, ursprung);
```

```
#ball ist distlistsort, nur dass in den einzelnen "Unterlisten" die Abstände nicht vermerkt sind, sondern nur die  
#Koordinaten der Punkte mit gleichem Abstand.
```

```
#Außerdem kommt zuerst die Liste mit Koordinaten, wo die Punkte den größten Abstand zu 0 haben und
```

```
#Punkte, die doppelt getroffen werden stehen
```

```
#hier nur noch ein mal drin.
```

```
ball:=[]:
```

```
for i from 1 to nops(dist) do
```

```
  ball:=[op(ball),[]]:
```

```
  for j from 2 to nops(distlistsort[9-i]) do
```

```
    found:=false:
```

```
    for k from 1 to nops(ball[i]) do
```

```
      if abs(evalf(distlistsort[9-i][j][1]-ball[i][k][1]))<epsilon
```

```
      and abs(evalf(distlistsort[9-i][j][2]-ball[i][k][2]))<epsilon
```

```
      and abs(evalf(distlistsort[9-i][j][3]-ball[i][k][3]))<epsilon then
```

```
        found:=true:
```

```
        break:
```

```
      fi:
```

```
    od:
```

```
    if found=false then ball[i]:=[op(ball[i]), distlistsort[9-i][j]]: fi:
```

```
  od:
```

```
od:
```

```
#Symmetrie
```

```
with(LinearAlgebra):
```

```
Digits:=30:
```

```
epsilon:=0.0000001:
```

```
ba:=ball[8]:
```

```
N:=nops(ba):
```

```
X:=proje8:
```

```
#Prozedur zum Prüfen ob 3 Vektoren linear unabhängig sind
```

```
linearunabh:= proc(v1,v2,v3)
```

```
  local M,d:
```

```
  M:=Matrix([v1,v2,v3]):
```

```
  d:=Determinant(M):
```

```
  if evalf(abs(d))<epsilon then
```

```
    RETURN(false):
```

```
  fi:
```

```
  RETURN(true):
```

```
end proc:
```

```
#Prozedur zum Prüfen, ob eine 3x3 Matrix R eine Drehmatrix ist (Determinante 1 und orthogonal)
```

```
istorth:=proc(R)
```

```
local d, i, j, Einh:
```

```
  d:=Determinant(R):
```

```
  if evalf(abs(d-1))>epsilon then
```

```
    RETURN(false):
```

```
  fi:
```

```
  Einh:=Multiply(R,R^+):
```

```
  for i from 1 to 3 do
```

```
    for j from 1 to 3 do
```

```
      if i=j then
```

```
        if evalf(abs(Einh[i][j]-1))>epsilon then
```

```
          RETURN(false):
```

```
        fi:
```

```
      else
```

```
        if evalf(abs(Einh[i][j]))>epsilon then
```

```
          RETURN(false):
```

```
        fi:
```

```
      fi:
```

```
    od:
```

```
  od:
```

```
  RETURN(true):
```

```
end proc:
```

```
#Liste der Variablen für die Berechnung der Drehmatrix
```

```
A:=Matrix(3, x):
```

```
listofVar:=[]:
```

```
for i from 1 to 3 do
```

```
  for j from 1 to 3 do
```

```
    listofVar:=[op(listofVar),A(i,j)]:
```

```
  od:
```

```
od:
```

```
#Zähler der Drehmatrizen
```

```
counter:=0:
```

```
#Basis von  $\mathbb{R}^3$ 
```

```
v1:=ba[1]:
```

```
v2:=ba[2]:
```

```

v3:=ba[3]:
if linearunabh(v1,v2,v3)=false then print("ERROR"); fi:

for d from 1 to N do
  for e from 1 to N do
    for f from 1 to N do
      #Wir suchen eine Matrix, die die 3 Basisvektoren auf 3 andere linear unabhängige Vektoren
      #dreht
      vi:=ba[d]:
      vj:=ba[e]:
      vk:=ba[f]:
      if linearunabh(vi,vj,vk)=true then
        vd:=Multiply(A,Vector(v1)):
        ve:=Multiply(A,Vector(v2)):
        vf:=Multiply(A,Vector(v3)):
        eqs:=[vd[1]=vi[1], vd[2]=vi[2], vd[3]=vi[3], ve[1]=vj[1], ve[2]=vj[2], ve[3]=vj[3], vf[1]=vk[1],
            vf[2]=vk[2], vf[3]=vk[3]]:
        sol:=solve(eqs, listofVar):

        R:=subs(sol[1],A):

        if istorth(R)=true then
          for i from 1 to nops(X) do
            z:=Multiply(R,Vector(X[i])):
            found:=false:
            Xstabil:=true:
            for j from 1 to nops(X) do
              if abs(evalf(z[1]-X[j][1]))<epsilon and abs(evalf(z[2]-X[j][2]))<epsilon
              and abs(evalf(z[3]-X[j][3]))<epsilon then
                found:=true:
                fi:
              od:
              if found=false then Xstabil:=false: fi:
            od:
            if Xstabil=true then
              counter:=counter+1;
              fi:
            fi:
          od:
        od:
      od:
    od:
  print("Es gibt", counter, "Drehmatrizen, die das Bild in sich selbst abbilden.");
  #Prüfen, ob zentrale Symmetrie eine Symmetrie des Bildes ist:
  zS:=Matrix([[ -1,0,0], [0,-1,0], [0,0,-1]]):
  for i from 1 to nops(X) do
    z:=Multiply(zS,Vector(X[i])):
    found:=false:
    Xstabil:=true:
    for j from 1 to nops(X) do
      if abs(evalf(z[1]-X[j][1]))<epsilon and abs(evalf(z[2]-X[j][2]))<epsilon
      and abs(evalf(z[3]-X[j][3]))<epsilon then
        found:=true:
        fi:
      od:
      if found=false then Xstabil:=false: fi:
    od:
  if Xstabil=true then print("Die zentrale Symmetrie ist eine Symmetrie des Bildes");fi:

```

Anhang C

Programm zum  
dreidimensionalen Modell von  
 $E_8$  nach G.Moxness

```

with(ListTools):
with(plots):
#with(plottools):
with(coxeter):
with(weyl):
with(LinearAlgebra):
with(geom3d):
Digits:=30:

Phi:=(1+sqrt(5))*1/2:
phi:=1/2*(sqrt(5)-1):
epsilon:=0.000000000001:
H4fold:=Matrix([[Phi, 0, 0, 0, phi^2, 0, 0, 0],
                [0, phi, 1, 0, 0, -phi, 1, 0],
                [0, 1, 0, phi, 0, 1, 0, -phi],
                [0, 0, phi, 1, 0, 0, -phi, 1],
                [phi^2, 0, 0, 0, Phi, 0, 0, 0],
                [0, -phi, 1, 0, 0, phi, 1, 0],
                [0, 1, 0, -phi, 0, 1, 0, phi],
                [0, 0, -phi, 1, 0, 0, phi, 1]]):
H4foldinv:=MatrixInverse(H4fold):

x:=[0, -Phi*2*sin(2*Pi/60), 0, 1, 0,0,0,0]:
y:=[Phi*2*sin(2*Pi/30), 0, 2*sin(2*Pi/15), 0, 0,0,0,0]:
z:=[0, 1, 0, Phi*2*sin(2*Pi/60), 0,0,0,0]:

X:=Multiply(H4foldinv, Vector[column](x)):
Y:=Multiply(H4foldinv, Vector[column](y)):
Z:=Multiply(H4foldinv, Vector[column](z)):

R:=E8:

Se8:=[e1-e2, e2-e3, e3-e4, e4-e5, -1/2*e1-1/2*e2-1/2*e3-1/2*e4-1/2*e5-1/2*e6-1/2*e7-1/2*e8, e6+e7, e5-e6,
e6-e7]:

posRoots:=pos_roots(Se8):
allRoots:=[op(posRoots), (-1)*op(posRoots)]:
nar:=nops(allRoots):
npr:=nops(posRoots):

#Dies soll die Liste werden, in der die Koordinaten aller projizierten Wurzeln stehen (also alle Punkte, die
#später geplottet werden)
projRoots3d:=[]:

for i in allRoots do
  #Die X-Komponente der projizierten Wurzel von i ist <i|X>, Y und Z genauso.
  p3dx:=iprod(i, BilinearForm(X, Vector([e1,e2,e3,e4,e5,e6,e7,e8]), conjugate=false)):
  p3dy:=iprod(i, BilinearForm(Y, Vector([e1,e2,e3,e4,e5,e6,e7,e8]), conjugate=false)):
  p3dz:=iprod(i, BilinearForm(Z, Vector([e1,e2,e3,e4,e5,e6,e7,e8]), conjugate=false)):
  p3d:=[evalf(p3dx), evalf(p3dy), evalf(p3dz)]:
  projRoots3d:=[op(projRoots3d), p3d]:

od:
print("Es gibt ",nops(projRoots3d), " Punkte.");
points3d:=pointplot3d(projRoots3d):

#In dieser Liste sollen alle Linien stehen, die später gezogen werden
lines3d:=[]:

```

```

#in dieser Liste werden Punktepaare abgespeichert um zu überprüfen, ob es eine solche line schon einmal
#gegeben hat (benötigt für die Symmetriesuche)
lines3dlist:=[]:

counterl:=0:
#Diese Schleife durchläuft alle Paare von Wurzeln und überprüft ob deren Differenz wieder eine Wurzel ist.
#wenn ja muss eine Linie zwischen den entsprechenden Punkten, die sich durch die Projektion dieser
#Wurzeln ergeben, gezogen werden.
for i from 1 to nar-1 do
  for j from i+1 to nar do
    if member(allRoots[i]-allRoots[j], allRoots) then
      counterl:=counterl+1;
      lines3d:=[op(lines3d), plottools[line](projRoots3d[i], projRoots3d[j], thickness=0)]:
      lines3dlist:=[op(lines3dlist), [projRoots3d[i], projRoots3d[j]]]:
    fi:
  od:
od:
print("Es gibt ",counterl,"Linien.");
#distlist soll eine liste von listen sein. in jedem eintrag steht an erster Stelle wie weit die Punkte die an allen
#weiteren Stellen stehen vom Ursprung entfernt sind.
distlist:=[]:

for i in projRoots3d do
  #Abstand vom aktuellen Bildpunkt berechnen
  normi:=norm(Vector(i),2):
  #Diese Variable soll angeben, ob es für diesen Abstand schon eine Liste gibt
  found:=false:
  #überprüfen, ob es schon weitere Punkte mit dem gleichen Abstand gibt, wenn ja aktuellen Punkt in die
  #entsprechende Liste dazu schreiben
  for j from 1 to nops(distlist) while found=false do
    if evalf(abs(normi-distlist[j][1]))<epsilon then
      distlist[j]:=[op(distlist[j]), i]:
      found:=true:
    fi:
  od:

  #wenn es noch keine Liste für diesen Abstand gab, wird eine erstellt und an distlist angehängt
  if found=false then
    distlist:=[op(distlist), [normi, i]]
  fi:
od:
#dist ist eine Liste in der alle Abstände, die vorkommen stehen. diese wird sortiert.
dist:=[]:
for i from 1 to nops(distlist) do
  dist:=[op(dist), distlist[i][1]]:
od:
dist:=sort(dist):

#distlistsort ist distlist, bei dem die Listen absteigend nach dem Abstand zum Ursprung sortiert werden.
distlistsort:=[]:
for i from nops(dist) by -1 to 1 do
  for j from 1 to nops(distlist) do
    if distlist[j][1]=dist[i] then
      distlistsort:=[op(distlistsort), distlist[j]]:
      break:
    fi:
  od:
od:
#zeichnen der Punkte und Linien
display(points3d, lines3d);

```

```

#Symmetrie (Basisberechnung des Bildes mit den Geraden)
prpoints:={}:

#hier werden zur Überprüfung ob Ebene schon mal gefunden wurde der normale Vektor zur gefundenen
#Ebene gespeichert
normalvecs:=[]:
#hier werden alle Vektoren in einer Liste in der Liste gespeichert, die in den gefundenen Ebenen p liegen.
vecsinalp:=[]:

point(o, [0,0,0]):

#Durchläuft alle Paare von projizierten Wurzeln und berechnet via Gram-Schmidt Verfahren eine
#Orthogonobasis von der durch die Paare aufgespannten Ebene
for i from 1 to nops(projRoots3d)-1 do
  for j from i+1 to nops(projRoots3d) do
    B:=GramSchmidt([Vector(projRoots3d[i]), Vector(projRoots3d[j])]):
    if nops(B)=2 then
      point(p1, projRoots3d[i]):
      point(p2, projRoots3d[j]):
      if not AreCollinear(o, p1, p2) then
        #erzeugt Ebene aufgespannt von p1,p2, name der achsen soll x1,x2,x3 sein.
        plane(p, [o,p1,p2], [x1,x2,x3]):
        #überprüfung, ob diese Ebene schon mal getestet wurde
        found:=false:
        for k from 1 to nops(normalvecs) while found=false do
          if not evalb(solve([normalvecs[k][1][1]=c*NormalVector(p)[1],normalvecs[k][1][2]=c*NormalVector(p)[2], normalvecs[k][1][3]=c*NormalVector(p)[3]] , c)=[])) then
            found:=true:
          fi:
        od:

        if found=false then
          #soll die Anzahl der Vektoren zählen, die in p liegen.
          counter:=0:
          #hier sollen alle Vektoren gespeichert werden, die in p liegen.
          vecsinp:=[]:

          #läuft durch alle Punkte und schaut ob diese in der ebene liegen, wenn ja wird der
          #Zähler eins hoch gesetzt und die projizierte Wurzel an vecsinp angefügt
          for k from 1 to nops(projRoots3d) do
            p3inp:=subs([x1=projRoots3d[k][1], x2=projRoots3d[k][2], x3=projRoots3d[k][3]],
              Equation(p)):
            if evalb(p3inp) then
              counter:=counter+1:
              vecsinp:=[op(vecsinp), projRoots3d[k]]:
            fi:
          od:
          #wenn mehr als 30 vektoren in p liegen, wird der Normalenvektor und ein Vektor aus der
          #Ebene in normalvecs, außerdem werden die Vektoren, die in P liegen in vecsinalp
          #eingespeichert eingeschpeichert
          if counter>=30 then
            normalvecs:=[op(normalvecs), [NormalVector(p), projRoots3d[i]]]:
            vecsinalp:=[op(vecsinalp), vecsinp]:
          fi:
        fi:
      fi:
    fi:
  od:
od:

```

```

print("Es gibt ", nops(vecsinalp), "ebenen, die 30 oder mehr Vektoren aus dem 3d Bild enthalten");
plo:=[]:
points2d:=[]:
print(nops(vecsinalp[1]));
#Jetzt gehen wir alle Punkte durch und betrachten die Ebene, die orthogonal zu der eben gefundenen steht.
#In diese projektieren wir die Punkte aus dem 3d Bild
for i from 1 to nops(vecsinalp[1]) do #gab nur eine ebene p
  B:=[Vector(normalvecs[1][1])/norm(Vector(normalvecs[1][1]),2), Vector(vecsinalp[1][i])/
    norm(Vector(vecsinalp[1][i]), 2)]:
  prprRoots:=[]:
  equivclass2:=[]:
  dist:={}:
  for k in projRoots3d do
    pp:=[DotProduct(Vector(k), B[1])/DotProduct(B[1], B[1]),DotProduct(Vector(k), B[2])/DotProduct(B[2],
      B[2])]:

    #projektiere den Punkt in der Ebene auf den Normalenvektor der orthogonalen Ebene und berechne
    #seinen Abstand zur 0 (bei dem Bild liegen die Punkte auf Geraden senkrecht zur x Achse)
    gp:=DotProduct(pp[1]*B[1]+pp[2]*B[2], B[1])*B[1]:
    n:=norm(gp,2):
    prprRoots:=[op(prprRoots), pp]:
    equivclass2:=[op(equivclass2), [evalf(pp), Vector(k)]]:

    #suche, ob es diesen Abstand schon einmal gab, wenn nicht wird er dazu gefügt
    found:=false:
    for j from 1 to nops(dist) do
      if abs(dist[j]-evalf(n))<epsilon then
        found:=true:
        fi:
      od:
    if found=false then
      dist:={op(dist), n}:
    fi:
  od:
print("es gibt ", nops(dist), "verschiedene Geraden auf denen die projektierten Punkte im 2d liegen");

if nops(dist)<10 then
  print("Eine Basis ist ", B);
  #linien berechnen und zeichnen des Bildes
  lines2d:=[]:
  for i from 1 to nops(equivclass2) do
    for j from 1 to nops(lines3dlist) do
      if abs(evalf(equivclass2[i][2][1]-lines3dlist[j][1][1]))<epsilon
        and abs(evalf(equivclass2[i][2][2]-lines3dlist[j][1][2]))<epsilon
        and abs(evalf(equivclass2[i][2][3]-lines3dlist[j][1][3]))<epsilon then
        for k from 1 to nops(equivclass2) do
          if i<>k and abs(evalf(equivclass2[k][2][1]-lines3dlist[j][2][1]))<epsilon
            and abs(evalf(equivclass2[k][2][2]-lines3dlist[j][2][2]))<epsilon
            and abs(evalf(equivclass2[k][2][3]-lines3dlist[j][2][3]))<epsilon then
              lines2d:=[op(lines2d), plottools[line](equivclass2[i][1], equivclass2[k][1],
                thickness=0)]:
            fi:
          od:
        fi:
      od:
    od:
  plo:=[op(plo), [prprRoots, lines2d]]:
fi:
od:

```

```

for i from 1 to nops(plo) do
display(pointplot(plo[i][1]), plo[i][2]);
od;

#Überprüfung, ob eine Drehung um die z Achse das Bild fest lässt (weil ja wenn man auf die xy Ebene
#projektiert, das Van Oss Bild heraus kommt)
f:=false:
for i from 1 to 30 do
  Dreh:=Matrix([[cos(2*Pi/i), -sin(2*Pi/i),0],[sin(2*Pi/i),cos(2*Pi/i),0],[0,0,1]]):
  ttt:=1;
  for j from 1 to nops(projRoots3d) do
    if ttt=1 then
      v:=Multiply(Dreh,Vector(projRoots3d[j])):
      for k from 1 to nops(projRoots3d) do
        found:=false:
        if abs(evalf(v[1]-projRoots3d[k][1]))<epsilon and abs(evalf(v[2]-projRoots3d[k][2]))<epsilon
        and abs(evalf(v[3]-projRoots3d[k][3]))<epsilon then
          found:=true:
          fi:
        od:
        if found=false then ttt:=0: fi:
      fi:
    od:
  if ttt=1 then f:=true: print("Die Drehung um ", 2*Pi/i, "in der x-y Ebene erhält das Bild"); fi:
od:
if f=false then print("Keine Drehung um die z-Achse lässt das Bild fest."); fi:

```

# Literaturverzeichnis

- [1] James E. Humphreys: Reflection Groups and Coxeter Groups, Cambridge University Press, 1990
- [2] N. Bourbaki: Lie Groups and Lie Algebras, ch.4-6, Springer, 2002
- [3] J. Stembridge: <http://www.math.lsa.umich.edu/~jrs/coxplane.html>
- [4] H.S.M. Coxeter: Regular Polytopes London: Methuen 1948
- [5] H.S.M. Coxeter: Introduction to geometry, John Wiley and Sons, 1962
- [6] J.H. Conway und N.J.A Sloane: Sphere Packings, lattices and groups, Springer-Verlag 1988
- [7] R. V. Moody and J. Patera: Quasicrystals and icosians, Phys. A: Math. Gen. 26 (1993), 2829-2853.
- [8] <http://homepages.wmich.edu/~drichter/gossetzome.htm>
- [9] J Gregory Moxness, 'The 3D Visualization of  $E_8$  using an  $H_4$  Folding Matrix', <http://TheoryOfEverything.org/MyToE>
- [10] F.J. MacWilliams: The theory of Error-Correcting Codes, North-Holland Publishing Company, 1977
- [11] D. Welsh: Codes and Cryptography, Oxford University Press 1988
- [12] John H. Conway und Derek A. Smith: On Quaternions and Octonions: Their Geometry, Arithmetic and Symmetry, A K Peters 2003
- [13] John B. Baez: On Quaternions and Octonions: Their Geometry, Arithmetic, and Symmetry by John H. Conway and Derek A. Smith
- [14] YU. I. Manin: Cubic forms Algebra, Geometry, Arithmet, Elsevier Science Publishers B.V 1986
- [15] S.V. Matveev: Euler-Charakteristik. In: Michiel Hazewinkel (Hrsg.): Encyclopaedia of Mathematics. Springer-Verlag, Berlin 2002

- [16] <http://www-history.mcs.st-and.ac.uk/~john/MT4521/Lectures/L12.html>
- [17] G.Fischer: Lehrbuch der Algebra: Mit lebendigen Beispielen, ausführlichen Erläuterungen und zahlreichen Bildern, 2. Auflage. Vieweg+Teubner Verlag 2011
- [18] M. Artin: Algebra. Birkhäuser 1993
- [19] Dechant, Boehm, Twarock: Affine extensions of non-crystallographic Coxeter groups induced by projection, 2012, arXiv:1110.5228v2.
- [20] <http://arstechnica.com/science/2010/03/setting-the-record-straight-no-simple-theory-of-everything/>
- [21] A. G. Lisi: An Exceptionally Simple Theory of Everything. 2007 arXiv:0711.0770
- [22] <http://www.garibaldibros.com/linked-files/Borthwick-Garibaldi-E8.pdf>
- [23] J. Distler und S. Garibaldi: Ther is no “Theory of Everything“ inside  $E_8$ , arXiv:0905.2658
- [24] K. Jänich: Topologie, 8. Auflage, Springer Verlag 2005
- [25] S. L. van Oss: Das regelmässige Sechshundertzell und seine selbstdeckenden Bewegungen, J.Müller, 1988

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Desweiteren wurde diese Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt.

Mainz, 31. März 2015