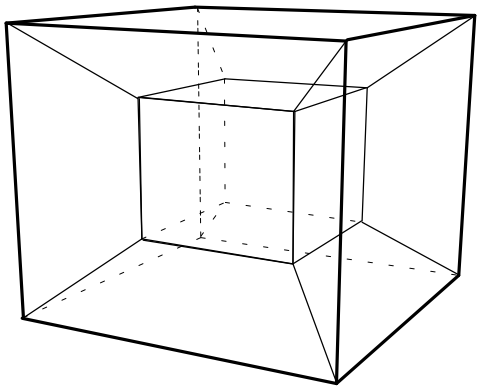
Hyperebenen in Hyperkuben
Eine Klassifizierung und Quantifizierung

Diplomarbeit



Zweidimensionale Darstellung der dreidimensionalen
Projektion eines vierdimensionalen Hyperwürfels

Ausgeführt am Institut für Grundlagen der
Informationsverarbeitung der Technischen Universität Graz

Vorgelegt von
Oswin Aichholzer
Oktober 1992

That is, of course, a perfectly valid procedure; but it does not give any insight into how on earth a person would ever have dreamed up the formula in the first place, except by some lucky guess.

DONALD E. KNUTH
In: *The Art of Computer Programming, Vol.1*

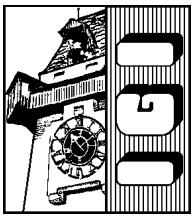
Für meine Eltern, Margaritha und Franz

Kurzfassung:

In dieser Arbeit werden affine Hyperebenen in höherdimensionalen Räumen behandelt, die den n-dimensionalen Hyperkubus schneiden. Dabei konzentriert sich die Untersuchung auf folgende Fragestellung: Wieviele und welche Arten von Hyperebenen gibt es, die durch Eckpunkte des Hyperkubus eindeutig festgelegt sind?

Neben der Betrachtung, wieviele solcher Hyperebenen existieren, wird eine Klassifizierung nach verschiedenen Kriterien wie Symmetrie, Parallelität zu den Koordinatenachsen, Anzahl der geschnittenen Eckpunkte etc. untersucht. Die Arbeit enthält sowohl eine vollständige enumerative Berechnung aller relevanten Werte bis einschließlich der achten Dimension, als auch die theoretische Herleitung allgemein gültiger Sätze über solche Hyperebenen.

Die Beiträge dieser Arbeit fallen in das Gebiet der geometrischen Kombinatorik und finden sowohl in der Codierungs- und Lerntheorie als auch in der linearen Optimierung sowie im VLSI-Design Anwendung.



Diese Arbeit wurde am Institut für Grundlagen der Informationsverarbeitung der Technischen Universität Graz unter der Leitung von Univ.-Prof. DI. Dr. Franz Aurenhammer durchgeführt.

Ich danke den Mitarbeitern des Instituts für die freundliche Unterstützung.

Mein besonderer Dank gilt auch Herrn Univ.-Doz. DI. Dr. Franz Rendl vom Institut für Mathematik B für seine wertvollen Anregungen zu Kapitel 2.4.1.

Inhaltsverzeichnis:

1. Einleitung 6
    1.1. Vorwort - Höhere Dimensionen 6
    1.2. Problemstellung, Motivation 7
        1.2.1. Klassen von Hyperebenen 7
        1.2.2. Motivation 7
        1.2.3. Verwandte Probleme und Resultate 9
    1.3. Schwerpunkte der Diplomarbeit 10
2. Grundlegende Theorie 13
    2.1. Hyperebenen und Hyperwürfel 13
    2.2. Symmetrie und k-Parallelität 17
    2.3. Anzahl der Eckpunkte 20
    2.4. Der Orthogonalvektor einer Hyperebene 23
        2.4.1. 0-1-Matrizen 24
3. Programmbeschreibung 27
    3.1. Allgemeines 27
    3.2. Die MULTI-FOR-Schleife 29
    3.3. Generator 32
    3.4. Hyperebenen Test 36
        3.4.1. Der Graycode 37
    3.5. Hyperebenen Klassifizierung 39
    3.6. Auswertung 40
    3.7. Programmtechnisches und Zeitverhalten 42
    3.8. Bedienung des Programms 43
4. Auswertung und Interpretation der Daten sowie theoretische Sätze 45
    4.1. Quantitative Auswertung 45
    4.2. Länge der Orthogonalvektoren 46
        4.2.1. Abschätzung der Anzahl von Hyperebenen mittels Orthogonalvektoren 49
        4.2.2. Maximaler Skalarwert b der Vektordarstellung einer Hyperebene 50
    4.3. Anzahl der Ecken auf Hyperebenen 52
    4.4. Symmetrische Hyperebenen 54
    4.5. Zueinander parallele Hyperebenen 59
    4.6. Schräge Hyperebenen und Spuren im Q_4 61
    4.7. Nichteindeutigkeit der Typen 65

5. Quelltexte des Programms 68
HYPER.PAS 68
TOOLS1.PAS 79
TOOLS2.PAS 84
MULTI_FO.PAS 87
6. Programmergebnisse: 90
Dimension 2 90
Dimension 3 91
Dimension 4 92
Dimension 5 93
Dimension 6 94
Dimension 7 98
Dimension 8 103
7. Tabellen 109
8. Zusammenfassung 114
9. Literaturverzeichnis: 115
Lebenslauf 117

1. Einleitung

1.1. Vorwort - Höhere Dimensionen

> Alice lachte. »Ich brauche es gar nicht zu versuchen«, sagte sie; »etwas Unmögliches kann man nicht glauben.«
> »Du wirst darin eben noch nicht die rechte Übung haben«, sagte die Königin. »In deinem Alter habe ich täglich eine halbe Stunde darauf verwendet. Zu Zeiten habe ich vor dem Frühstück bereits bis zu sechs unmögliche Dinge geglaubt. ...«
>
> LEWIS CARROLL
> In: *Alice hinter den Spiegeln*

Schließt man den uns vertrauten dreidimensionalen Raum aus, so finden naturgemäß besonders die zweite und vierte Dimension Interesse, sowohl in der populärwissenschaftlichen Literatur ([DEW1], [DEW2], [SARN], [GARD]) als auch der Kunst[1]. Obwohl der zweidimensionale Raum ebenfalls vertraut und leicht überschaubar scheint, gibt es auch hier durchaus überraschende Zusammenhänge (vergleiche [DEW1]). Während dabei der Zugang aber noch problemlos gelingt, wird in der vierten Dimension meist die Unterstützung von Computern benötigt ([DEW2], [SARN]), oder es werden nur sehr rudimentäre Aspekte betrachtet.

Kann man für ernsthafte Untersuchungen in der vierten Dimension noch auf zahlreiche Analogien zurückgreifen, so ist bei Betrachtung höherdimensionaler Räume Vorsicht vor "anschaulicher" Intuition geboten (vergleiche dazu auch [VOLK]). So beträgt zum Beispiel das Volumen des n-dimensionalen Einheitswürfels konstant 1 und der Abstand zweier diagonaler Eckpunkte $\sqrt{n}$, - dieser nimmt also mit größer werdendem n beliebig zu. Völlig anders ist die Situation bezüglich der n-dimensionalen Einheitskugel (r=1). Unabhängig von der Dimension können zwei Punkte auf ihrer Oberfläche (n-Sphäre) maximal den Abstand 2 voneinander entfernt sein. Die Fläche des Einheitskreises beträgt π, das Volumen der Einheitskugel 4,2 und das (Hyper)Volumen der vierdimensionalen Einheitskugel 4,9. Im fünfdimensionalen Fall ist es noch größer, nämlich 5,263, sinkt aber für n=6 auf 5,1 und nimmt von da an ständig ab. Gegen die Erwartung einer beliebigen Zunahme des Volumens besitzt dieses also ein Maximum (5,277) mit (gebrochener) endlicher Dimension 5,256 [GARD], [WELL].

Ein weniger bekanntes Beispiel für ein solches, nicht monotones Verhalten in höheren Dimensionen wird ebenfalls in [GARD] gezeigt: Das Flächenverhältnis eines Kreises zu einem umschriebenen Quadrat ($\pi/4$) ist größer als das Verhältnis eines einbeschriebenen Quadrats zum Kreis ($2/\pi$). Ebenso paßt eine Kugel "besser" in einen Würfel als umgekehrt, obwohl hier der Unterschied zwischen beiden Verhältnissen bereits kleiner ist. Bis

---

[1]So verwendet der Surrealist Salvador Dali in seinem Gemälde "Kreuzigung (Corpus Hypercubus)" (1953/54) einen in den dreidimensionalen Raum aufgeklappten Hyperwürfel als Kreuz. Heute befindet sich das Gemälde im Metropolitan Museum of Art in New York und gilt als eines der bedeutendsten religiösen Werke Dalis.

zur achten Dimension nimmt diese Relation fortlaufend ab und kehrt sich ab dem neundimensionalen Raum um: Das Verhältnis von "9-Kugel" zu "9-Würfel" ist kleiner als das Verhältnis von "9-Würfel" zu "9-Kugel".

Daß höhere Dimensionen nicht notwendigerweise auch schwerer zu behandeln sind, zeigt das bekannte Problem der dichtesten Kugelpackung sowie das Kuß-Zahl-Problem[2] (siehe [SLOA], [STEW]). Für letzteres sind exakte Lösungen für die Dimensionen eins, zwei, drei, acht und 24 schon länger bekannt, für die Dimension vier ist aber erst kürzlich eine Lösung gefunden worden [HSIA]. Für die restlichen Dimensionen sind lediglich obere und untere Schranken bekannt.

Für eine gute Einführung in geometrische Zusammenhänge siehe z.B. [EDEL] und [PRSH]. Eine ausführliche Behandlung von regulären Polytopen ist im bekannten Standardwerk von Coxeter zu finden [COX].

1.2. Problemstellung, Motivation

1.2.1. Klassen von Hyperebenen

Die Untersuchungen in dieser Arbeit konzentrieren sich auf folgende Fragestellung: Wieviele und welche Arten von Hyperebenen gibt es, die durch Eckpunkte des Hyperkubus eindeutig festgelegt sind? Betrachtet wird also jene Klasse von Hyperebenen, die sich durch Eckpunkte des Einheitswürfels aufspannen läßt. Dabei ist der Einheits-Hyperwürfel als das kartesische Produkt des Einheitsintervalls definiert ($[0,1]^d$), und seine Eckpunkte bilden die Menge $\{0,1\}^d$. Der Einheits-Hyperwürfel wird im Folgendem oft kurz als Q_d bezeichnet. Ein (d-1) dimensionaler Unterraum des $\mathbb{R}^d$ wird dann als Hyperebene bezeichnet.

Daneben gibt es weitere Möglichkeiten, das Verhältnis zwischen Hyperebenen und Einheitswürfel zu betrachten. So können speziell jene Hyperebenen untersucht werden, die den Hyperkubus lediglich in Eckpunkten schneiden und seine Kanten nicht kreuzen, womit der Schnittkörper die konvexe Hülle der geschnittenen Eckpunkte bildet. Oder man interessiert sich lediglich für die durch Hyperebenen geschnittenen Kanten des Würfels. Weiters kann natürlich auch die spezielle Struktur des durch den Schnitt gebildeten Polytops von Interesse sein. Beispiele zur Motivation solcher Betrachtungsweisen werden in den nächsten Abschnitten kurz gezeigt.

1.2.2. Motivation

Die folgenden Beispiele verdeutlichen Anwendungsmöglichkeiten von Hyperebenen, die durch Eckpunkte des Hyperkubus eindeutig festgelegt sind und stellen somit den Ausgangspunkt der hier durchgeführten Untersuchungen dar.

---

[2]Wieviele (Hyper-) Kugeln können so um eine zentrale Kugel herumgelegt werden, daß sie alle diese berühren?

A. String - Suche

Gegeben sind n 0-1-Strings der Länge d sowie ein zu suchender Anfrage-String. Gesucht wird jener String aus der Menge der vorhandenen n Strings, der zum Anfrage-String den geringsten Hammingabstand (Anzahl differierender Bits) aufweist, dem Anfrage-String gleichsam am ähnlichsten ist.

Würde man davon ausgehen, daß der Anfrage-String immer in der Menge enthalten ist, so könnte dieser durch binäre Suche ([MEHL],[SEDG]) höchst effizient gefunden werden. Ist der String jedoch nicht in der Menge enthalten, so kann dies bereits bei der ersten Fallunterscheidung der binären Suche zur Verzweigung in den falschen Teilbaum führen, da der optimale String sich im ungünstigsten Fall bereits an der ersten Stelle vom Anfrage-String unterscheidet. Da die Suche nun aber im falschen Teilbaum durchgeführt wird und dieser nicht mehr verlassen werden kann, kann der String mit dem minimalen Hammingabstand nicht mehr erreicht werden. Daher scheiden die binäre Suche und ähnliche effiziente Suchverfahren für dieses Problem aus.

Betrachtet man die gegebenen Strings als Teilmenge der Eckpunkte eines d-dimensionalen Hyperwürfels, so bilden diese Ecken aufgrund der Struktur des Hyperwürfels eine konvexe Hülle. Die String-Suche entspricht nun der Suche jener Ecke einer konvexen Hülle, die den minimalen Hammingabstand zu der, dem Anfrage-String entsprechenden Ecke hat. Aufgrund der Konvexität genügt es dabei, sich auf die Suche entlang der Kanten der konvexen Hülle in Richtung des abnehmenden Hammingabstandes zu beschränken, da kein "lokales Optimum" auftreten kann. Die benötigte Laufzeit dafür beträgt $O(d*\rho)$, wobei ρ die Valenz (Anzahl ausgehender Kanten) einer Ecke ist. Die Analyse der erwarteten Anzahl von Kanten führt unter anderem auf die Bestimmung von Hyperebenen durch die Eckpunkte des Hyperkubes, also genau auf die in Kapitel 1.2.1. besprochenen Fälle [CLSH].

B. Rundreise - Problem

Gegeben ist ein Graph mit n Knoten, für welchen ein geschlossener Pfad, welcher jeden Knoten genau einmal besucht (Hamiltonsche Tour), gesucht ist.

Durch Numerierung aller möglichen Kanten des Graphen kann man eine Bijektion zu einem Hyperwürfel der Dimension $\binom{n}{2}$ herstellen: Jeder Graph entspricht einer Ecke mit einer 1 an der i-ten Koordinate genau dann, wenn die i-te Kante im Graphen existiert. Jede Hamiltonsche Tour entspricht dann einer speziellen Ecke des Hyperwürfels. Die Menge aller solcher Touren entspricht dann der Eckmenge, deren konvexe Hülle das sogenannte Rundreise-Polytop ist. Die Suche nach der kürzesten Rundreise entspricht somit einem linearen Programm, dessen Ungleichungen wiederum Hyperebenen durch Eckpunkte des Hyperwürfels sind. Zur Lösung dieser, in der Praxis oft sehr großen, linearen Programme werden Heuristiken wie z.B. das Schnittebenen-Verfahren verwendet; die Kenntnis gewisser Hyperebenen durch Ecken bzw. deren Facetten im Rundreise-Polytop ist dabei entscheidend [GLS].

C. Weitere Anwendungen

Über die oben erklärten Probleme hinaus wäre eine direkte Anwendung von Hyperebenen in der Codierungstheorie vorstellbar: Zum Beispiel wählt man alle Ecken in einer geeigneten, fixen Hyperebene als gültige Codewörter. Damit wäre eine Unterscheidung zwischen gültigen und ungültigen Codes sehr einfach durch einen Hyperebenentest durchführbar. Geeignete Hyperebenen mit einer möglichst gleichmäßigen Verteilung der Hammingdistanz zwischen den Eckpunkten würden zu einer Erhöhung der Übertragungssicherheit führen.

Weitere Anwendungen sind auch in der linearen Optimierung denkbar, ähnlich dem oben erwähnten Rundreise-Problem. In der Tat führen etliche Optimierungsaufgaben auf Graphen auf "Teilpolytope" des Hyperkubus, so etwa das Zuordnungspolytop [GLS].

1.2.3. Verwandte Probleme und Resultate

A. Spezialfälle von Hyperebenen

Eines der wenigen Bücher, in denen der Zusammenhang zwischen Hyperebenen und Hyperkuben behandelt wird, ist das Standardwerk von Coxeter über Polytope [COX]. Darin untersucht er speziell jene Hyperebenen, die man erhält wenn man entlang eines Vektors von einem Außenpunkt des Hyperwürfels in sein Zentrum vordringt und dabei alle zu diesem Vektor normalen Hyperebenen betrachtet. Als Ausgangspunkt für den Vektor dienen dabei Eckpunkt, Mittelpunkt der Kante sowie Flächenmittelpunkt des Hyperkubes. Die dabei auftretenden Schnitte der Hyperebenen weisen entsprechende Symmetrien auf und werden mit zunehmender Dimension immer komplexer.

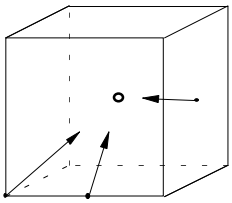


Abbildung 1: Betrachtungslinien für Hyperebenen

Darüber hinausgehend sind scheinbar noch keine allgemeinen Ergebnisse bekannt, da Hyperebenen im Zusammenhang mit Hyperkuben noch kaum untersucht wurden. Dies dürfte zum Teil auch daran liegen, daß für einen enumerativen Zugang in diesem Bereich speziell für höhere Dimensionen ein Computer notwendig ist. Daher wurden solche Untersuchungen früher nicht durchgeführt.

B. "Abdecken" des Hyperwürfels durch Hyperebenen

Eine weitere Frage ist, wieviele Hyperebenen notwendig sind, um alle Eckpunkte eines n-dimensionalen Hyperwürfels abzudecken. Während die Antwort darauf noch trivial ist, da zwei zueinander parallele Hyperebenen mit je 2^{n-1} Ecken genügen, ist sie es für die folgende Frage nicht mehr: Wieviele Hyperebenen sind notwendig, um alle Ecken *bis auf eine* (diese darf nicht abgedeckt werden, ist also explizit ausgenommen) abzudecken? Eine offensichtliche obere Schranke dafür bilden n Hyperebenen welche sich alle in einer Ecke treffen und jeweils genau die Hälfte des Hyperwürfels abdecken. Die diagonal gegenüberliegende Ecke ist dann jene, welche nicht abgedeckt wird. Daß diese einfache Lösung aber auch optimal ist, d.h. daß es keine Anordnung mit weniger Hyperebenen geben kann, welche die Bedingungen erfüllt, konnte erst vor kurzem gezeigt werden [ALFÜ].

C. Lerntheorie

Die Lerntheorie beschäftigt sich mit Hyperebenen, die ausschließlich durch Kanten, nicht aber durch Eckpunkte des Hyperwürfels dringen, und so die Menge der Eckpunkte in zwei Klassen teilen:

Betrachtet man die Menge aller Objekte mit n verschiedenen Ja-Nein-Eigenschaften[3], so kann man diese als Eckpunkte des n-dimensionalen Einheitswürfels auffassen. Jedes Objekt entspricht dann einer Ecke $x=(x_1,..,x_n)$ und jede Eigenschaft entspricht einer Koordinatenachse und kann entweder vorhanden sein ($x_i=1$) oder auch nicht ($x_i=0$). Will man nun ein bestimmtes Objekt durch einfache (lineare) Ja-Nein-Fragen ermitteln, so können diese Fragen als Hyperebenen aufgefaßt werden. Diese Fragen (Hyperebenen) teilen alle Objekte (Eckpunkte) in zwei Klassen, je nachdem ob die Frage mit ja (Ecke liegt über der Ebene) oder nein (Ecke liegt unterhalb) zu beantworten ist. Da es dabei egal ist, wo genau die Hyperebene die Kanten schneidet, kann man sich auf jeweils einen Vertreter für alle (unendlich vielen) dazu gleichwertigen Hyperebenen beschränken.

Folgende interessante kombinatorische Fragestellungen ergeben sich: Wieviele verschiedene sinnvolle Hyperebenen (≡ nicht triviale Fragen) gibt es? Was ist die maximale Anzahl von Kanten, die eine Hyperebene gleichzeitig schneiden kann? Wieviele Hyperebenen (≡ Fragen) werden mindestens benötigt, um alle Kanten des Hyperwürfels zu schneiden (≡ das Objekt eindeutig zu ermitteln) [NEIL],[BAK] ?

1.3. Schwerpunkte der Diplomarbeit

Wie schon vorher erwähnt werden in dieser Arbeit Hyperebenen behandelt, die durch den Hyperwürfel eindeutig "aufgespannt" werden können. Diese Betrachtung schließt natürlich Hyperebenen, die nur Kanten des Hyperwürfels schneiden, aus. Es ist jedoch durchaus der Spezialfall möglich, daß keine Kanten geschnitten werden, die Hyperebene also nur Eckpunkte des Hyperwürfels berührt[4]. Neben der Frage, wieviele solche Hyperebenen möglich sind, wird eine Klassifizierung nach verschiedenen Kriterien wie Symmetrie, Parallelität zu den Koordinatenachsen, Anzahl der geschnittenen Eckpunkte und Ähnlichem vorgenommen.

Um einen möglichst guten Einblick in die Zusammenhänge der dort auftretenden Strukturen zu erhalten, werden für höhere Dimensionen (ab vier) konkrete Zahlenwerte und Beispiele für alle auftretenden Klassen von Hyperebenen berechnet. Aus diesen, mittels Computer zu ermittelnden Werten, werden dann allgemein gültige Zusammenhänge abgeleitet und theoretisch bewiesen. Sowohl quantitative (Wieviele Hyperebenen gibt es?) als auch qualitative (Welche Arten gibt es, und in welchen Zusammenhang stehen diese?) Verhältnisse werden dabei beschrieben.

---

[3]Das sind Eigenschaften, die nur zwei möglichen Realisationen (vorhanden oder nicht) besitzen.

[4]Tatsächlich gibt es für die Dimensionen eins, zwei und drei ausschließlich solche Hyperebenen. Erst ab dem vierdimensionalen Raum schneiden derart festgelegte Hyperebenen auch Kanten des Hyperwürfels. Für eine genaue Betrachtung dieser Zusammenhänge siehe Kapitel 4.6.

Die Arbeit gliedert sich in folgende drei Schwerpunkte:

A. Enumeration und Klassifizierung von Hyperebenen

Um unter gegebenen Zeit- und Speicherbeschränkungen auch Berechnungen für höhere Dimensionen durchführen zu können wurde ein durch zahlreiche theoretische Überlegungen optimiertes Programm entwickelt. Ausgangspunkt der Berechnungen bilden alle in Frage kommenden Normalvektoren der Hyperebenen, die durch anschließende Untersuchungen klassifiziert werden. Vorteil dieser Vorgangsweise ist es, daß eine Speicherung der bereits untersuchten Hyperebenen selbst nicht notwendig ist. Da die Anzahl der Hyperebenen überexponential steigt, wird erst durch diese Methode eine Berechnung für Dimensionen größer als sechs möglich, da der Speicherbedarf andernfalls viel zu groß wäre.

Mittels der Hadamar-Ungleichung konnten Schranken für den maximalen Koordinatenwert von Orthogonalvektoren gefunden werden, wodurch der zu untersuchende Bereich eingegrenzt wird. Weiters konnte der Test, welche Eckpunkte auf einer Hyperebene liegen, durch Verwendung des Gray-Codes sehr effizient gestaltet werden. Um den Rechenaufwand möglichst gering zu halten, wurden über Symmetriebetrachtungen die Normalvektoren in Gruppen zusammengefaßt, für die jeweils nur ein Repräsentant untersucht werden muß. Durch diese Überlegungen wurde ein Vorstoß von der sechsten Dimension (ca. 250.000 Hyperebenen) bis in die achte Dimension (ca. 86 Milliarden Hyperebenen) möglich.

Neben der Betrachtung, wieviele solcher Hyperebenen existieren, wird bei der Klassifizierung nach verschiedenen Kriterien wie Symmetrie, Parallelität zu den Koordinatenachsen, Anzahl der geschnittenen Eckpunkte, Länge des Normalvektors etc. unterschieden.

B. Interpretation der Daten

Im Folgenden werden kurz die wichtigsten empirischen Schlüsse, die aus dem gewonnenen Datenmaterial gezogen werden können, wiedergegeben:

Die Anzahl schräger Hyperebenen dominiert die Anzahl k-paralleler Hyperebenen[5] für $k \geq 1$. Mit zunehmender Dimension verlieren daher bei der quantitativen Betrachtung k-parallele Hyperebenen an Bedeutung. In Q_8 beträgt z.B. ihr Anteil bereits weniger als 0,7 %.

Die Anzahl möglicher Hyperebenen wächst tatsächlich überexponentiel ($\approx 2^{d^2}$). Vergleicht man damit die triviale obere Schranke[6] von $\binom{2^d}{d}$, so zeigt sich, daß diese Schranke nicht so überhöht ist, wie ihre Herleitung vermuten hätte lassen.

---

[5]Das sind zu k der kartesischen Koordinatenachsen parallele Hyperebenen. 0-parallele Hyperebenen werden auch als schräg bezeichnet. (Vergleiche dazu Definition 9 auf Seite 17.)

Bei den schrägen Hyperebenen überwiegen jene mit einer möglichst geringen Anzahl von Eckpunkten aus Q_d. So liegen bis zur achten Dimension auf über 30 % der schrägen Hyperebenen aus Q_d nur genau d Eckpunkte. Dies erklärt auch, warum die vorher erwähnte triviale Schranke so gut ist, denn sie wurde ja unter Verwendung solcher Hyperebenen hergeleitet.

Betrachtet man die verschiedenen Arten von Hyperebenen, so finden sich in jeder Dimension neue Typen, welche sich nicht aus Typen tieferer Dimensionen aufbauen lassen - dies erschwert entsprechend die Analyse der Hyperebenen.

C. Theoretische Sätze

Das gewonnene Datenmaterial und die entsprechende Interpretation der Daten führten zur Entwicklung zahlreicher theoretischer Sätze. So wurde neben einer Schranke für die maximalen Koeffizienten des Normalvektors auch eine Schranke über das maximale Skalarprodukt des Vektors mit einer Ecke des Hyperkubes gezeigt. Weiters wurden mehrere Sätze über die mögliche Anzahl von Eckpunkten auf Hyperebenen bewiesen.

Wichtige Ergebnisse wurden über den Zusammenhang zwischen verschiedenen Arten von Hyperebenen gezeigt. So konnten k-parallele Hyperebenen auf schräge Hyperebenen sowie schräge, unsymmetrische Hyperebenen auf schräge, symmetrische (≡ gleich viele Ecken auf beiden Seiten) Hyperebenen reduziert werden. Das bedeutet für weitere Untersuchungen, daß nur mehr schräge, symmetrische Hyperebenen betrachtet werden müssen, da sich alle anderen Arten von Hyperebenen auf diese sehr spezielle Klasse zurückführen lassen!

Obere und untere Schranken wurden für die Anzahl von zueinander parallelen Hyperebenen abgeleitet. Ein weiterer Satz ermöglicht die einfache Überprüfung, wann eine Hyperebene "hüllentreu" ist, das heißt wann ihre Spur im Hyperkubus tatsächlich die konvexe Hülle der auf ihr liegenden Eckpunkte ist.

---

[6]Diese obere Schranke erhält man durch folgende Überlegung: Um eine Hyperebene eindeutig festzulegen, benötigt man zumindest d der 2^d vorhandenen Eckpunkte. Man hat aber maximal $\binom{2^d}{d}$ Möglichkeiten diese d Eckpunkte auszuwählen, womit unmittelbar die obere Schranke folgt.

2. Grundlegende Theorie

Dieser Abschnitt liefert die grundlegenden Definitionen zur Handhabung von Hyperebenen und Einheitswürfel in höheren Dimensionen. Neben einer Erläuterung der wichtigsten dabei vorkommenden Strukturen, die eine genaue Klassifizierung der Hyperebenen nach verschiedenen Kriterien ermöglichen, werden bereits einige Sätze zur Vereinfachung der Zusammenhänge gezeigt.

Besonderer Wert wird dabei auf Betrachtungen gelegt, die eine optimierte Programmgestaltung für die anschließende numerische Untersuchung ermöglichen. Eine Diskussion über die positiven Auswirkung dieser Überlegungen auf die Laufzeit des Programms ist im nächsten Abschnitt zu finden.

2.1. Hyperebenen und Hyperwürfel

Eine erste, einfache Definition von Hyperebenen liefert [MESC]:

Definition 1: Ein (durch eine lineare Gleichung gegebener) (d-1) dimensionaler Unterraum des $\mathbb{R}^d$ (d = 4,5,...) heißt eine *Hyperebene*.

Eine etwas andere, aber gleichwertige Definition erhält man über die allgemeinere Beschreibung von k-Flächen, wie sie z.B. in [EDEL] zu finden ist:

Definition 2: Für $0 \leq k \leq d$ ist eine *k-Fläche* definiert als die affine Hülle von k+1 linear unabhängigen Punkten. Gewöhnlich wird dabei eine 0-Fläche als *Punkt*, eine 1-Fläche als *Linie* und eine 2-Fläche als *Ebene* bezeichnet. Eine (d-1)-Fläche wird dann als *Hyperebene* bezeichnet.

Bemerkung: Natürlich gibt es in $\mathbb{R}^d$ nur eine d-Fläche, nämlich $\mathbb{R}^d$ selbst. Der Vollständigkeit halber wird die leere Menge oft auch als (-1)-Fläche bezeichnet.

Je nach Anwendungsgebiet gibt es zahlreiche Möglichkeiten, Hyperebenen zu beschreiben. Oft werden dabei jedoch zusätzliche Bedingungen gefordert, wie z.B. nicht vertikale Hyperebenen (das sind Hyperebenen, die parallel zur d-Koordinatenachse laufen) [EDEL]. Im folgenden soll eine hier brauchbare Beschreibungsweise angegeben werden:

Definition 3: Eine Hyperebene $H \subseteq \mathbb{R}^d$, $0 \notin H$, wird durch die Angabe eines Vektors $a \in \mathbb{R}^d$ eindeutig beschrieben durch: $H := \{ x \in \mathbb{R}^d \mid \langle a, x-a \rangle = 0 \}$. Siehe dazu Abbildung 2.
Dabei bezeichnet $\langle a,b \rangle := a_1b_1+a_2b_2+...+a_db_d$, $a,b \in \mathbb{R}^d$ das gewöhnliche Skalarprodukt (inneres Produkt) zweier Vektoren.

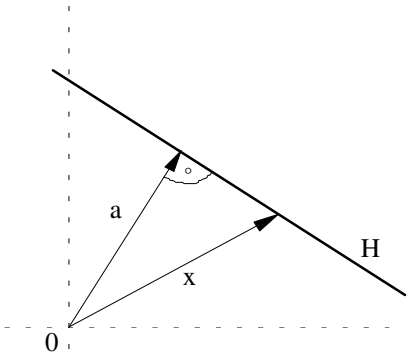


Abbildung 2: Beschreibung einer Hyperebene durch einen Vektor a.

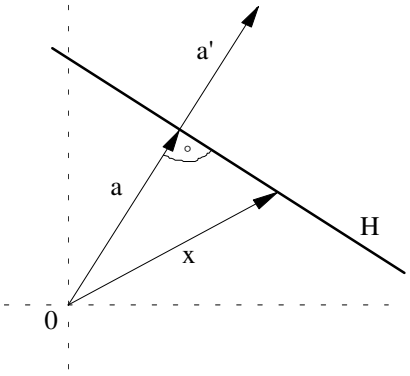


Abbildung 3: Allgemeinere Beschreibung einer Hyperebene.

Diese Definition führt allerdings bei Hyperebenen, die durch den Nullpunkt gehen, zu Problemen, da hier der Vektor a verschwindet. Sämtliche solche Hyperebenen würden dann durch den Nullvektor dargestellt werden, und es wäre somit keine Eindeutigkeit mehr gegeben. Um dies zu vermeiden, läßt man die Länge von a beliebig (aber > 0) oder legt sie je nach Bedarf geeignet fest.

Da aber nun durch den Vektor a nur die Orientierung und nicht die genaue Lage der Hyperebene festgelegt wird, benötigt man einen weiteren skalaren Wert b. Dieser wird wie folgt hergeleitet (vergleiche dazu Abbildung 3):

Sei $a' := k*a$, $k \in \mathbb{R}$, so gilt für alle $x \in H$ nach Definition 3:

$$\langle a',x\rangle = \langle k*a,x\rangle = k*\langle a,x\rangle = k*(\langle a,x-a\rangle + \langle a,a\rangle) = k*\langle a,a\rangle.$$

Setzt man also $b := k*\langle a,a\rangle$ so erhält man folgende, verträgliche Darstellung einer Hyperebene:

Definition 4: Eine Hyperebene $H \subseteq \mathbb{R}^d$ wird durch die Angabe eines Paares (a,b), $a \in \mathbb{R}^d$, $b \in \mathbb{R}$, eindeutig beschrieben durch: $H := \{x \in \mathbb{R}^d \mid \langle a,x\rangle = b\}$

Mit $HE_d := \{H \mid H \text{ wie oben definiert}\}$ wird die Menge aller Hyperebenen in $\mathbb{R}^d$ bezeichnet.

Man beachte, daß nun keine Einschränkungen der Hyperebene mehr notwendig sind. Da die Hyperebene orthogonal auf den Vektor a steht, wird dieser auch *Orthogonalvektor* genannt.

Diese Beschreibung der Hyperebene wird häufig verwendet und bietet beweistechnische Vorteile. So lassen sich darauf zum Beispiel problemlos Methoden der linearen Algebra anwenden oder es läßt sich die Zusammenfassung mehrerer Hyperebenen als multilineares Polynom darstellen ([ABCO],[ALFÜ]).

Definition 5: Der *d-dimensionalen Einheits-Hyperwürfel* (im weiterem kurz *EHW*) ist das kartesische Produkt des Einheitsintervalls: $Q_d := [0,1]^d$. Mit $\{0,1\}^d$ wird dann die Menge der Eckpunkte des EHW*s* bezeichnet.

Völlig gleichberechtigt könnte man für diese Definition auch die Werte {-1,+1} verwenden, wie dies z.B. in [ALFÜ] getan wird. Die hier verwendete Definition hat aber den Vorteil, daß man die Eckpunkte direkt als d-stellige Binärzahlen auffassen kann, was beweistechnisch oft günstig ist.

Im englischen Sprachraum wird der Einheits-Hyperwürfel wegen seines dimensionsunabhängigen (Hyper-) Volumes von 1 oft auch als *'measure polytope'* bezeichnet. Weiters stellt er eine der drei möglichen regelmäßigen Polytope in höheren Dimensionen dar: Neben dem measure polytope (allgemeiner Würfel) sind dies noch regular simplex (allgemeiner Tetraeder) und cross polytope (allgemeiner Oktaeder). Interessant ist,

daß von den fünf existierenden platonischen Körpern nur diese drei eine Fortsetzung in höheren Dimensionen finden. Eine ausführliche Behandlung von regelmäßigen Polytopen ist in [COX] zu finden.

Trivialerweise ist die Anzahl der Ecken eines Hyperwürfels gleich 2^d. Allgemein gilt für die Anzahl k-dimensionaler Subkuben (Ecken sind dabei der nulldimensionale Fall) die in einem d-dimensionalen Hyperwürfel auftreten, folgende Rekursion:

$$W_k^d = 2*W_k^{d-1} + W_{k-1}^{d-1},\ W_d^d=1,\ d \geq k \geq 0$$

Dies sieht man, wenn man die Bildung des d-dimensionalen EHWs aus dem d-1 dimensionalen EHW betrachtet. Dabei werden die bereits vorhandenen k-dimensionalen Subkuben von Q_{d-1} verdoppelt und seine k-1 dimensionalen Subkuben werden um die neue, zu ihnen orthogonale Dimension erweitert, also ebenfalls zu k-dimensionalen Subkuben im Q_d (vergleiche auch [COX]).

Eine andere Überlegung liefert eine direkte Formel für diese Anzahl: Für W_k^d müssen jeweils jene d-k Koordinaten ausgewählt werden, welche der k-dimensionale Subkubus in Q_d nicht aufspannt (die also "fest" sind), wofür es $\binom{d}{d-k}$ Möglichkeiten gibt. Jede dieser Koordinaten kann nun 0 oder 1 sein, also gibt es dafür 2^{d-k} Möglichkeiten. Zusammen liefert dies $W_k^d = \binom{d}{d-k} * 2^{d-k} = \binom{d}{k} 2^{d-k}$.

Dieses Ergebnis ist auch mit der vorhergehenden Rekursion verträglich:

$$2*W_k^{d-1} + W_{k-1}^{d-1} = 2*\binom{d-1}{k} 2^{(d-1)-k} + \binom{d-1}{k-1} 2^{(d-1)-(k-1)} = \left(\frac{(d-1)!}{k!*(d-1-k)!} + \frac{(d-1)!}{(k-1)!*(d-k)!}\right) 2^{d-k} =$$

$$= \left(\frac{d-k}{d} * \frac{d!}{k!*(d-k)!} + \frac{k}{d} * \frac{d!}{k!*(d-k)!}\right) 2^{d-k} = \frac{d!}{k!*(d-k)!} * 2^{d-k} = \binom{d}{k} 2^{d-k} = W_k^d$$

Damit wurde folgendes Lemma gezeigt:

Lemma 1: Die Anzahl k-dimensionaler Subkuben die in einem d-dimensionalen Hyperwürfel auftreten ist

$$W_k^d = \binom{d}{k} 2^{d-k},\ d \geq k \geq 0$$

Einfache "Kenndaten" des d-dimensionalen EHWs sind in folgender Tabelle zusammengefaßt:

Dimension d	Ecken im Q_d	Kanten im Q_d	Flächen (Quadrate) im Q_d	Begrenzende Hyperebenen im Q_d (Facetten)
d	$W_0^d = 2^d$	$W_1^d = d*2^{d-1}$	$W_2^d = 2^{d-3}*d*(d-1)$	$W_{d-1}^d = 2*d$
1	2	1	-	2
2	4	4	1	4
3	8	12	6	6
4	16	32	24	8
5	32	80	80	10
6	64	192	240	12
7	128	448	672	18
8	256	1024	1792	16
9	512	2304	4608	18
10	1024	5120	11520	20

Tabelle 1: "Kenndaten" von Hyperwürfeln

Für die weiteren Untersuchungen interessieren uns nurmehr jene Punkte auf einer Hyperebene, die aus Q_d stammen. Umgekehrt sind speziell jene Hyperebenen interessant, die als affine Hülle der Ecken aus Q_d gebildet werden können. Aus Definition 2 ergibt sich unmittelbar, daß zumindest d Punkte aus Q_d notwendig sind, um eine Hyperebene zu beschreiben.

Definition 6: Als *Hyperebenen im Q_d* werden jene Hyperebenen bezeichnet, die sich als affine Hülle von Ecken aus Q_d bilden lassen. Die Menge diese Hyperebenen wird kurz mit $HE(Q_d)$ bezeichnet. $|HE(Q_d)|$ ist somit die Anzahl aller möglichen Hyperebenen im Q_d.

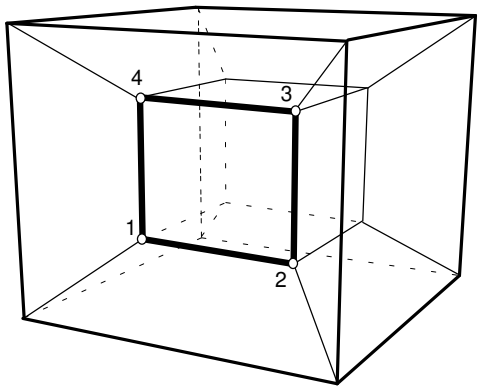


Abbildung 4: d Ecken aus Q_d definieren nicht immer eine Hyperebene (Hier Q_4 mit den Ecken 1 bis 4, die lediglich ein Q_2 bilden.)

Um die Probleme beim Zählen von Hyperebenen im Q_d zu verdeutlichen seien folgende Beispiele erwähnt:

- Hyperebenen im Q_d können mehr als d Ecken aus Q_d beinhalten. Siehe dazu z.B. Abbildung 5.
- Durch d Ecken aus Q_d wird nicht zwingend eine Hyperebene festgelegt. Dieser Fall tritt erstmals im Q_4 auf, wo durch 4 Eckpunkte lediglich eine Ebene (Q_2) gebildet wird. Siehe dazu Abbildung 4.

Definition 7: Eine Hyperebene $H \in HE(Q_d)$ heißt *(j,p,k)-Ebene*, wenn sie $p \geq d$ Ecken von Q_d enthält und die verbleibenden Ecken in Teilmengen zu j und k Ecken teilt, wobei $j \leq k$ gilt.

Diese Bezeichnung ist insofern redundant, als natürlich $j + p + k = 2^d$ gilt.

Eine triviale obere Schranke für die maximale Anzahl von Hyperebenen im Q_d erhält man durch folgende Überlegung: Um eine Hyperebene eindeutig festzulegen, benötigt man zumindest d der 2^d vorhandenen Eckpunkte. Man hat aber maximal $\binom{2^d}{d} (\approx 2^{d^2})$ Möglichkeiten diese d Eckpunkte auszuwählen und somit eine einfache obere Schranke gefunden. Diese Schranke erscheint ob ihres extremen Wachstums sehr grob. Wie aber die numerischen Untersuchungen zeigen werden, besitzt die tatsächliche Anzahl von Hyperebenen ein sehr ähnliches Verhalten.

2.2. Symmetrie und k-Parallelität

Die folgenden Definitionen stellen eine erste Möglichkeit der Klassifizierung von Hyperebenen dar. Die dabei verwendeten Begriffe der Symmetrie und Parallelität entsprechen den herkömmlichen Bedeutungen in niederen Dimensionen und werden hier auf höhere Dimensionen verallgemeinert.

Definition 8: Eine Hyperebene $H \in HE(Q_d)$ heißt *symmetrisch*, wenn sie vom Typ (j,p,j) ist.

Definition 9: Eine Hyperebene $H \in HE(Q_d)$ ist *k-parallel*, wenn sie genau zu k Kanten des Hyperwürfels (≡ Achsen des Koordinatensystems) parallel ist ($0 \le k \le d-1$). 0-parallele Hyperebenen werden als *schräg* bezeichnet.

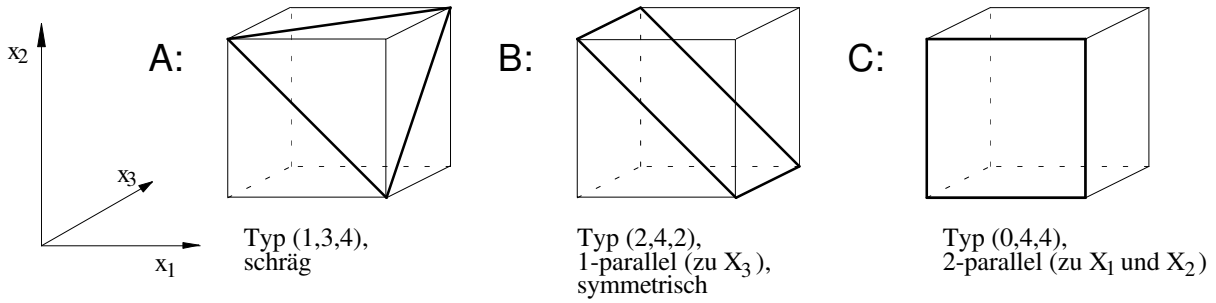


Abbildung 5: Beispiele für verschiedene Ebenen-Typen, die durch Ecken des Q_3 (gewöhnlicher Würfel) gebildet werden.

Definition 10: Sei $H \in HE(Q_d)$ eine Hyperebene durch $p \ge d$ Ecken von Q_d. Q_t sei ein Sub-Hyperwürfel von Q_d mit $0 \le t \le d$. Dann heißt $H \cap Q_t$ die *Spur* von H im Q_t.

Die Spur von H im Q_t enthält also genau jene Eckpunkte von Q_t, die auf H liegen. Damit folgt nun unmittelbar folgendes

Lemma 2: $H \in HE(Q_d)$ ist genau dann k-parallel im Q_d, wenn die Spur von H im von den restlichen d-k Achsen aufgespannten Q_{d-k} schräg ist und k minimal ist (d.h. k ist der kleinste Wert für den dies gilt). Dieser Q_{d-k} ist dann zu den k Achsen orthogonal.

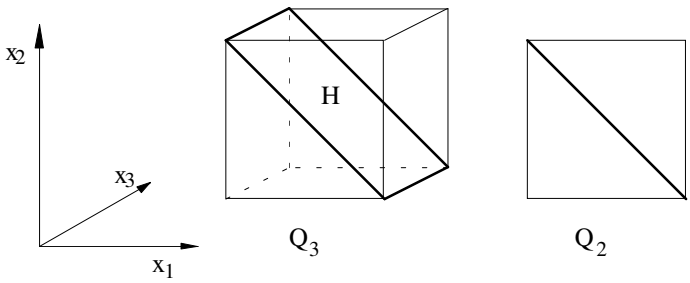


Abbildung 6: Schräge Spur einer 1-parallelen Hyperebene aus Q_3 im Q_2.

In Abbildung 6 ist H 1-parallel (zu X_3). Die restlichen zwei Achsen X_1 und X_2 spannen einen Q_2 (Quadrat) auf. (Q_2 ist daher orthogonal zur X_3-Achse.) Die Spur von H im Q_2 ist ein Liniensegment, das im Q_2 schräg liegt. (Die Spur von H im Q_3 ist ein Rechteck.)

Definition 11: Für $d > 0$ und $0 \le k < d$, sei $e_k(d)$ die Anzahl von k-parallelen Hyperebenen im Q_d.

Mit dieser Definition läßt sich nun ein erster, wichtiger Satz angeben:

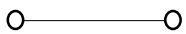
Satz 1: Für $d > 0$ und $0 \le k < d$ gilt: $e_k(d) = \binom{d}{k} * e_0(d-k)$

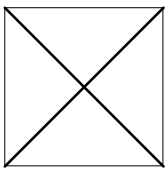
Beweis: Im Q_d gibt es $\binom{d}{k}$ Möglichkeiten, orthogonal zu k Achsen zu sein (wähle k der d vorhandenen Achsen dafür aus). Jede dieser Möglichkeiten legt einen Q_{d-k} fest, und damit laut Lemma 2 $e_0(d-k)$ schräge Spuren hierin. Jede solche Spur entspricht dann aber genau einer verschiedenen k-parallelen Hyperebene im Q_d. ❑

Dieser Satz ist deshalb besonders schön, weil es somit genügt sich lediglich auf die Untersuchung von schrägen Hyperebenen zu konzentrieren. Die gesamte Anzahl von Hyperebenen im Q_d ergibt sich dann mit

$$\left| HE(Q_d) \right| = e_0(d) + \sum_{i=1}^{d-1} \binom{d}{i} * e_0(d-i)$$

Die folgenden Beispiele geben die schrägen Hyperebenen im Q_1, Q_2 und Q_3 an:

$e_0(1) = 2$:  2 Eckpunkte

$e_0(2) = 2$:  2 Diagonalen

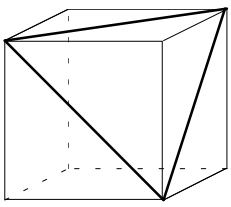
$e_0(3) = 8$:  Pro Ecke eine Ebene vom Typ (1,3,4)

Abbildung 7: Schräge Hyperebenen im Q_1, Q_2 und Q_3.

Für den Q_3 ergibt sich somit:

$$|HE(Q_3)| = e_0(3) + \sum_{i=1}^{2} \binom{3}{i} * e_0(3-i) = e_0(3) + 3*e_0(2) + 3*e_0(1) = 8 + 6 + 6 = 20.$$

Vergleicht man dies mit Abbildung 5, so sieht man sofort den direkten Zusammenhang: 8 schräge Ebenen (Bild A), 6 1-parallele Ebenen (Bild B) und 6 2-parallele Ebenen (Bild C) können unmittelbar durch abzählen festgestellt werden und stimmen mit den oben ermittelten Werten überein.

Für den Q_4 ergeben sich die Zahlen:

$$e_1(4) = \binom{4}{1} * e_0(3) = 4 * 8 = 32$$
$$e_2(4) = \binom{4}{2} * e_0(2) = 6 * 2 = 12$$
$$e_3(4) = \binom{4}{3} * e_0(1) = 4 * 2 = 8$$

Allgemein gilt:

$$e_{d-1}(d) = \binom{d}{d-1} * e_0(1) = d * 2 = 2d$$

(Anzahl der Facetten (= begrenzenden Hyperebenen) von Q_d.)

$$e_{d-2}(d) = \binom{d}{d-2} * e_0(2) = d * (d-1)$$
$$e_{d-3}(d) = \binom{d}{d-3} * e_0(3) = \frac{4}{3} * d * (d-1) * (d-2)$$

Diese Folge läßt sich natürlich für $e_{d-k}(d)$ aus der Kenntnis von $e_0(k)$ fortsetzen. Wie man aus späteren Berechnungen jedoch sehen kann, steigt der dabei auftretende Vorfaktor stark an und gehorcht keiner einfachen Darstellung ($e_{d-4} = \frac{11}{3} * ...$; $e_{d-5} = \frac{338}{15} * ...$; $e_{d-6} = \frac{14668}{45} * ...$ u.s.w.).

Das verbleibende Hauptproblem ist es also, $e_0(d)$ für $d \geq 4$ zu bestimmen.

Eine etwas andere Sichtweise für parallele Hyperebenen sowie ein alternativer Beweis für Satz 1 ergibt sich aus der Betrachtung des Orthogonalvektors von Hyperebenen.

Lemma 3: Sei $H \in HE(Q_d)$ und (a,b) der zugehörige Orthogonalvektor und Skalarwert nach Definition 4. Dann gilt: H ist k-parallel $\Leftrightarrow$ genau k Koordinatenwerte von a sind 0.

Beweis: Es genügt zu zeigen, daß sogar folgende, stärkere Aussage gilt: genau dann, wenn H parallel zur Achse X_i ist, gilt $a_i = 0$. Daraus folgt dann unmittelbar das Lemma.

"$\Rightarrow$" Sei H parallel zu X_i. Dann gibt es für jede Ecke p die auf H liegt eine Ecke p' mit $p'_j := p_j$, $j \neq i$ und $p'_i := 1-p_i$ die ebenfalls auf H liegt. Es gilt somit <a,p> = <a,p'> = b und daher $0 = b - b = <a,p'> - <a,p> = a_i*(1-p_i) - a_i*p_i = a_i$.

"$\Leftarrow$" Sei $a_i=0$. Analog zum ersten Schritt gilt dann <a,p> = b $\Leftrightarrow$ <a,p'> = b. Dies bedeutet aber wiederum daß H parallel zu X_i sein muß. ❑

Damit ergibt sich nun ein alternativer

Beweis 2 für Satz 1: Sei $H \in HE(Q_d)$ k-parallel und a der zugehörige Orthogonalvektor. Dann sind k der d Koordinatenwerte von a gleich Null, wofür es $\binom{d}{k}$ Möglichkeiten gibt. Die restlichen Werte von a bilden einen Vektor der Dimension d-k der keine 0 enthält und somit eine schräge Hyperebene im Q_{d-k} repräsentieren muß. Dafür gibt es aber per Definition $e_0(d-k)$ Möglichkeiten, womit der Satz folgt. ❑

Um die Anzahl von möglichen Orthogonalvektoren mit denselben Koordinatenwerten, aber verschiedener Anordnung der Elemente, berechnen zu können, dient folgendes

Lemma 4: Sei $a = (a_1,a_2,..,a_d)$ der Orthogonalvektor von $H \in HE(Q_d)$. Weiters bezeichnen $n_1,..,n_k$, $k \le d$, $1 \le n_i \le d$, $n_1 + .. + n_k = d$, die Häufigkeit der verschiedenen Koordinatenwerte. Dann gibt es $\frac{d!}{n_1! * n_2! * ... * n_k!}$ Hyperebenen in $HE(Q_d)$, deren Orthogonalvektoren alle dieselben Koordinatenwerte (in verschiedener Anordnung) besitzen.

Beweis: Die Beschreibung der Werte d und n_1 bis n_k entspricht genau der Definition des Multinomialkoeffizienten (siehe z.B. [JACO]; oft wird dafür auch die Bezeichnung Polynomialkoeffizient verwendet [BRON]), der daher sofort die gesuchte Anzahl liefert. Aufgrund der Symmetrie des EHWs (die Koordinatenachsen können beliebig permutiert werden) bildet natürlich auch jeder dieser Orthogonalvektoren tatsächlich eine Hyperebene im Q_d, sofern zumindest ein spezieller Orthogonalvektor dies tut (ist nach Voraussetzung erfüllt). ❑

2.3. Anzahl der Eckpunkte

Da der EHW vollkommen spiegel- und drehsymmetrisch ist, ist natürlich jede seiner Ecken gleichberechtigt und besitzt daher die gleichen Typen und Anzahl von Hyperebenen. Es ist daher naheliegend sich auf die Berechnung der Hyperebenen durch einen Punkt zu beschränken und daraus die Gesamtzahl aller Hyperebenen im EHW zu ermitteln.

Lemma 5 Gibt es in einer beliebigen Ecke des Q_d m Hyperebenen vom Typ (j,p,k), so gibt es im Q_d genau $\frac{m * 2^d}{p}$ Hyperebenen dieses Typs.[7]

Beweis: Es gibt im Q_d 2^d Ecken und, wegen der Symmetrie des EHWs, durch jede dieser Ecken m solcher Hyperebenen. Dabei würde aber jede Hyperebene p-mal gezählt werden, weshalb sich also genau $\frac{m * 2^d}{p}$ Hyperebenen ergeben. ❑

[7]Da es sich dabei um eine Anzahl handelt, muß das Ergebnis ganzzahlig sein. Für eine ungerade Anzahl p von Eckpunkten auf der Hyperebene bedeutet dies dann aber, daß m ein vielfaches von p sein muß!

Werden nun nur Hyperebenen durch eine bestimmte Ecke betrachtet, so ist es vorteilhaft den Nullpunkt (Ecke (0,..,0) dafür zu verwenden: Für Hyperebenen durch den Nullpunkt ist der Skalarwert b stets 0 und es genügt daher die Orthogonalvektoren zu betrachten. Weiters folgt damit, daß für schräge Hyperebenen durch den Nullpunkt zumindest ein Koordinatenwert des Orthogonalvektors > 0 und analog ein Koordinatenwert < 0 sein muß, da ansonsten keine Ecke aus Q_d (ungleich dem Nullpunkt) auf der Hyperebene liegen könnte.

Lemma 6 Sei $H \in HE(Q_d)$ und p die Anzahl von Ecken aus Q_d in H. Dann gilt: $d \le p \le 2^{d-1}$. Diese Schranken sind scharf.

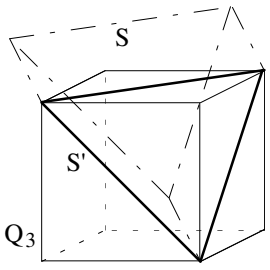
Beweis: Daß zumindest d Ecken notwendig sind wurde bereits früher festgestellt. Ein für beliebige Dimensionen gültiges Beispiel mit d Ecken liefert die Betrachtung des Orthogonalvektors a = (1,1,..,1) und dem Skalarwert b = 1. Die so definierten (schrägen) Hyperebenen werden durch die d linear unabhängigen Eckpunkte (1,0,..,0) , (0,1,0,..,0) , .. , (0,..,0,1) aufgespannt.

Da H maximal d-1 parallel sein kann, muß H zu einer Raumachse schräg sein. O.B.d.A. sei dies X_1. Falls nun aber die Ecke $(0,b_1,..,b_{d-1})$, $b_1,..,b_{d-1} \in \{0,1\}$, in H liegt, darf natürlich $(1,b_1,..,b_{d-1})$ nicht in H liegen und umgekehrt. D.h. es kann nur jeweils eine dieser beiden Ecken in H sein. Aus der Binärarithmetik folgt damit sofort, daß es maximal 2^{d-1} mögliche Ecken aus Q_d in H gibt. Daß auch die obere Schranke scharf ist, sieht man mit a = (1,0,..,0) und b = 0 sofort. Auf dieser (d-1 parallelen) Hyperebene liegen die 2^{d-1} Ecken $(0,b_1,..,b_{d-1})$, mit beliebigen Elementen $b_1,..,b_{d-1} \in \{0,1\}$. ❑

Lemma 7 Sei $H \in HE(Q_d)$, k-parallel. Weiters sei S' die Spur von H im orthogonalen Q_{d-k} und H' die affine Hülle von S' im Q_{d-k}. Dann gilt: Enthält H' p Ecken so enthält H $p*2^k$ Ecken.

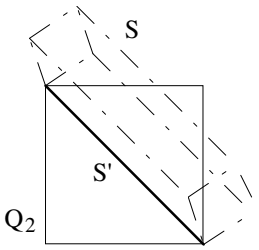
Beweis: Da H' eine schräge Hyperebene von Q_{d-k} ist, kann H' keinen Q_t, $t \ge 1$ enthalten (H' enthält also nur Ecken von Q_0). Enthält nun H' p Ecken, so enthält H p mal einen Q_k: Jeder Q_k enthält eine der p Ecken und wird von den k Achsen, zu denen H parallel ist, aufgespannt. ❑

Falls S' die konvexe Hülle der Ecken in H' ist, so ist $S = H \cap Q_d$ (die Spur von H im Q_d) ebenfalls die konvexe Hülle der Ecken in H, da $S = S' \times Q_k$. Die folgenden Beispiele sollen die unterschiedlichen Arten für Q_4 verdeutlichen.



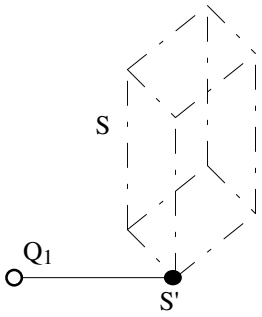
Beispiel 1: H im Q_4 ist 1-parallel:

H' ist eine Ebene, deren Spur S' im orthogonalen Q_3 ein Dreieck bildet. Dabei ist p = 3, und H enthält drei EHW Q_1 (Linien). Also ist $S = S' \times Q_1$ ein dreiseitiges Prisma mit 6 Ecken aus Q_4.



Beispiel 2: H im Q_4 ist 2-parallel.

H' ist eine Gerade, deren Spur S' im orthogonalen Q_2 ein Segment bildet. Dabei ist p = 2, und H enthält zwei EHW Q_2 (Quadrate). Also ist $S = S' \times Q_2$ ein Quader mit 8 Ecken aus Q_4.



Beispiel 3: H im Q_4 ist 3-parallel:

H' ist ein Punkt, dessen Spur S' im orthogonalen Q_1 ein Ecke bildet. Dabei ist p = 1, und H enthält einen EHW Q_3 (Würfel). Also ist $S = S' \times Q_3$ ein Würfel (Facette von Q_4!) mit 8 Ecken aus Q_4.

Abbildung 8: Unterschiedliche Spuren im Q_4.

Bei allen obigen Beispielen ist die Spur S' von H' tatsächlich die konvexe Hülle der Ecken des orthogonalen EHWs (Q_3, Q_2 bzw. Q_1). Damit ist auch die Spur S von H die konvexe Hülle von Ecken aus Q_4. Aus den Beispielen ergibt sich auch, daß dies für alle schrägen Spuren aus Q_1, Q_2 und Q_3 ebenfalls gilt. Leider gilt dies für schräge Hyperebenen ab Q_4 nicht immer: S ist nicht immer die konvexe Hülle der Ecken von Q_d, die in H liegen! Vielmehr ist S ein konvexes Polytop, dessen Ecken entweder Ecken von Q_d oder Schnitte von H mit Kanten von Q_d sind. Als Konsequenz kann man die Form von Spuren nicht alleine von den Ecken in den entsprechenden Hyperebenen ablesen, was die Analyse von $HE(Q_d)$ beträchtlich erschwert.

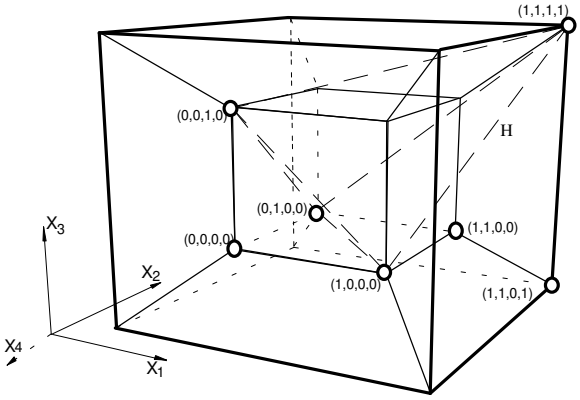


Abbildung 9: Schräge Hyperebene H im Q_4, dessen Spur **nicht** die konvexe Hülle von Ecken aus Q_4 ist.

Ein Beispiel für eine solche Hyperebene ist in Abbildung 9 zu sehen. Die durch $a = (1,1,1,-2)$[8] und b = 1 gebildete Hyperebene H enthält genau die vier Ecken (1,0,0,0), (0,1,0,0), (0,0,1,0) und (1,1,1,1) aus Q_4 und ist vom Typ (4,4,8). Die Ecke (1,1,0,0) liegt dabei über H, die Ecke (1,1,0,1) darunter[9]. Da jedoch beide Ecken durch eine Kante e verbunden sind (diese liegt parallel zu X_4), schneidet die Hyperebene H diese Kante (genau bei (1,1,0,1/2)!). Damit ist aber $e \cap H$ eine Ecke von S aber nicht von Q_4 und somit ist S nicht die konvexe Hülle von Ecken aus Q_4.

---

[8]Sowohl Orthogonalvektoren als auch Ecken des EHWs werden der Einfachheit halber in transponierter Vektorschreibweise dargestellt.

[9]Eine Ecke x wird dabei als über H liegend bezeichnet, wenn <a,x> > b und analog unter H liegend wenn <a,x> < b gilt. Ebenso wäre natürlich eine Vertauschung dieser Begriffe oder die Bezeichnungen links und rechts möglich, wie dies auch in der Literatur (oft uneinheitlich) gehandhabt wird.

Es gibt aber auch für $d \geq 4$ schräge Spuren, die durchaus die konvexe Hülle von Ecken aus Q_d sind. Beispiele dazu, sowie eine vollständige Auflistung der schrägen Spuren im Q_4 werden im Kapitel 4.6. ab Seite 61 dargestellt. Dort wird auch ein allgemeiner Satz über den Zusammenhang zwischen Orthogonalvektoren und Spuren von Hyperebenen gezeigt.

2.4. Der Orthogonalvektor einer Hyperebene

In diesem Abschnitt wird der mit Definition 4 eingeführte Orthogonalvektor einer Hyperebene näher betrachtet. Dabei wird o.B.d.A. nur der bereits erwähnte Fall von durch den Nullpunkt gehenden Hyperebenen behandelt, da aufgrund der Symmetrie des EHWs alle anderen Fälle analog dazu gezeigt werden können.

Sei also $H \in HE(Q_d)$ eine solche, durch den Nullpunkt gehende Hyperebene. Dann gibt es neben dem Nullpunkt zumindest d-1 linear unabhängige Ecken aus Q_d, durch die H festgelegt wird[10]. Bezeichnet man mit $v_1, v_2, .., v_{d-1}$, $v_i \in \{0,1\}^d$ diese Ecken, so wird durch

$$V * a = b \text{ wobei } V = \begin{pmatrix} v_{1_1} & v_{1_2} & \cdots & v_{1_d} \\ v_{2_1} & v_{2_2} & \cdots & v_{2_d} \\ \vdots & \vdots & \ddots & \vdots \\ v_{d-1_1} & v_{d-1_2} & \cdots & v_{d-1_d} \end{pmatrix}, b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

ein lineares, homogenes Gleichungssystem der Ordnung d für den Orthogonalvektor von H beschrieben (die Zeilen von V sind dabei die Ecken aus Q_d). Da der Rang von V gleich d-1 ist, ergibt sich der Orthogonalvektor a damit als einparametrige Lösung, d.h. seine Richtung ist eindeutig festgelegt, während seine Länge beliebig (aber ungleich Null) gewählt werden kann. O.B.d.A. sei nun V' die aus V ohne deren letzter Spalte gebildete (d-1)x(d-1)-Matrix, wobei rang(V') = rang(V) = d-1. (Ansonsten existiert eine andere Spalte von V durch deren Streichung der Rang unverändert bleibt. Durch Indexvertauschung kann dieser Fall analog betrachtet werden.) Legt man nun den Parameter $a_d = -1$ fest, was wegen der Ranginvarianz erlaubt ist (siehe Klammerbemerkung), so ergibt sich das lineare Gleichungssystem

$$V' * a' = b' \text{ wobei } V' = \begin{pmatrix} v_{1_1} & v_{1_2} & \cdots & v_{1_{d-1}} \\ v_{2_1} & v_{2_2} & \cdots & v_{2_{d-1}} \\ \vdots & \vdots & \ddots & \vdots \\ v_{d-1_1} & v_{d-1_2} & \cdots & v_{d-1_{d-1}} \end{pmatrix}, b' = \begin{pmatrix} v_{1_d} \\ v_{2_d} \\ \vdots \\ v_{d-1_d} \end{pmatrix}$$

der Ordnung (d-1). Dieses ist nun, wie aus der linearen Algebra bekannt, wegen rang(V') = d-1 = rang(V'|b') = rang (V) eindeutig lösbar. Man beachte auch, daß $V' \in \{0,1\}^{(d-1)x(d-1)}$ und $b' \in \{0,1\}^{d-1}$. Nun bezeichnet man mit V'^k jene Matrix, in der die k-te Spalte von V' durch b' ersetzt wurde und mit D = det(V') und $D_k = det(V'^k)$. Löst man dieses Gleichungssystem nach der Regel von Cramer (siehe z.B. [FISCH],[BRON]), dann gilt: $a'_1 = \frac{D_1}{D}$, $a'_2 = \frac{D_2}{D}$, .. , $a'_{d-1} = \frac{D_{d-1}}{D}$. Um nun den Orthogonalvektor a zu erhalten setzt man den (früher mit -1

[10] Liegen mehr als d Ecken auf der Hyperebene, so können stets d Ecken (inkl. Nullpunkt) entsprechend ausgewählt werden; vergleiche dazu auch Definition 2.

gewählten) Parameter a_d auf -D. Damit gilt für den Orthogonalvektor a = $(D_1,D_2,..,D_{d-1},-D)$. Da sowohl die Elemente von V' als auch von b' ganzzahlig sind, gilt dies auch für D und D_k, k=1,..,d-1, und damit für die Elemente von a. Um nun für jede Hyperebene einen eindeutigen Orthogonalvektor zu erhalten, wird der so berechnete Vektor a noch durch den größten gemeinsamen Teiler seiner Elemente dividiert. Weiters wird a gegebenenfalls mit -1 multipliziert, sodaß von allen Elementen von a, die ungleich Null sind, jenes mit dem niedersten Index positiv wird. Damit hat man den kürzesten, ganzzahligen Orthogonalvektor für diese Ebene eindeutig ermittelt. Unter Berücksichtigung von $V' \in \{0,1\}^{(d-1)x(d-1)}$ und $b' \in \{0,1\}^{d-1}$, kann man nun das Ergebnis im folgenden Satz zusammenfassen:

SATZ 2: Sei $H \in HE(Q_d)$. Dann gibt es für H einen eindeutig bestimmten kürzesten Orthogonalvektor $a = (a_1,a_2,..,a_d)$ mit a_i ganzzahlig, i=1,..,d. Bezeichnet man mit A_d die Menge der 0-1-Matrizen der Ordnung d so gilt weiters für die Elemente von a die Abschätzung:

$$|a_i| \le \max_{M \in A_{d-1}} |\det M| \quad i = 1,..,d$$

Daß, wie eingangs erwähnt, dieser Satz auch für Hyperebenen gilt, die nicht durch den Nullpunkt gehen, wird durch eine Transformation solcher Hyperebenen in den Ursprung gezeigt: Man betrachte eine beliebige Ecke $x \in H$. Für alle Koordinatenrichtungen, an denen x = 1 gilt, wird die Hyperebene am Mittelpunkt des Würfels gespiegelt. D.h. für alle Ecken auf H wird in dieser Koordinate eine 0 zur 1 und eine 1 zur 0. Weiters ändert der Orthogonalvektor für diesen Koordinatenwert sein Vorzeichen[11]. Wird dieser Vorgang für alle Einsen aus x wiederholt, so wird x schließlich zur Nullecke und man hat nun eine zur ursprünglichen Hyperebene äquivalente Hyperebene durch den Nullpunkt. Da der zugehörige Orthogonalvektor sich dabei lediglich für einige Koordinatenwerte im Vorzeichen unterscheidet gilt somit obiger Satz für beliebige Hyperebenen.

Um also die möglichen Orthogonalvektoren und damit Hyperebenen im Einheits-Hyperwürfel einzuschränken, werden im folgenden Kapitel Determinanten von 0-1-Matrizen näher untersucht.

2.4.1. 0-1-Matrizen

Einer der wichtigsten und bekanntesten Sätze zur Abschätzung von Determinanten ist die Hadamar-Ungleichung (siehe z.B. [KNU2] S414):

SATZ 3: Hadamar-Ungleichung: Für eine beliebige nxn-Matrix B mit den Elementen b_{ij} gilt:

$$|\det B| \le \prod_{i=1}^{n} \left(\sum_{j=1}^{n} b_{ij}^2 \right)^{1/2}$$

[11]Der Skalarwert b ändert sich dabei um den Wert des Orthogonalvektors an dieser Koordinatenstelle. Dies ist jedoch hier für die Betrachtung der maximalen Länge eines Orthogonalvektors ohne Bedeutung.

Damit können nun Determinanten von 0-1-Matrizen näher betrachtet werden:

Definition 12: Mit A_d wird die Menge der *0-1-Matrizen* der Ordnung d bezeichnet, also $A \in \{0,1\}^{dxd}$.

SATZ 4 Sei $A \in A_d$. Dann gilt $|\det(A)| \le 2 * \left(\frac{d+1}{4}\right)^{\frac{d+1}{2}}$

Diese Abschätzung ist zumindest für $d = 2^k-1$, k=1,2,.. scharf.

Beweis: Sei $B = 2*A$, dann gilt $|\det(B)| = 2^d * |\det(A)|$. Weiters sei $C = \left(\begin{array}{c|ccc} 1 & 1 & \cdots & 1 \\ \hline 1 & & & \\ \vdots & & B & \\ 1 & & & \end{array}\right)$ eine

(d+1)x(d+1)-Matrix. Durch Subtraktion der 1. Zeile von den restlichen erhält man :

$$C' = \left(\begin{array}{c|ccc} 1 & 1 & \cdots & 1 \\ \hline -1 & & & \\ \vdots & & \pm 1 & \\ -1 & & & \end{array}\right)$$

Man beachte daß $C' \in \{-1,+1\}^{(d+1)x(d+1)}$ und $|\det(C')| = 2^d * |\det(A)|$ gilt. Nach der Hadamar-

Ungleichung gilt aber $|\det(C')| \le \prod_{i=1}^{d+1} \left(\sum_{j=1}^{d+1} c'^2_{ij}\right)^{1/2} = (d+1)^{\frac{d+1}{2}}$, also gilt damit

$$|\det(A)| = \frac{1}{2^d} * |\det(C')| \le \left(\frac{1}{2}\right)^{\frac{1}{2}} * (d+1)^{\frac{d+1}{2}} = 2 * \left(\frac{1}{4}\right)^{\frac{d+1}{2}} * (d+1)^{\frac{d+1}{2}} = 2 * \left(\frac{d+1}{4}\right)^{\frac{d+1}{2}}$$

Daß diese Schranke für $d=2^k-1$ scharf ist, folgt aus den anschließenden Betrachtungen über Hadamar-Matrizen! Da für C' auch eine Hadamar-Matrix verwendet werden kann und diese für zumindest die Ordnungen 2^k existiert, folgt mit Lemma 8 die Schärfe der Schranke. ❑

Definition 13: Eine quadratische Matrix H der Ordnung n mit Elementen $h_{ij} = \pm 1$, welche die Matrizengleichung $HH^T = H^TH = nI_n$ erfüllt, wird *Hadamar-Matrix* genannt. Dabei ist I_n die Einheitsmatrix der Ordnung n.

Beispiele für Hadamar-Matrizen sind:

$$n = 1: \begin{pmatrix} 1 \end{pmatrix} \quad n = 2: \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad n = 4: \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Mit Hilfe des Kronecker-Produkts läßt sich aus der Existenz von Hadamar-Matrizen der Ordnung m und n sofort die Existenz von Hadamar-Matrizen der Ordnung $m * n$ zeigen (siehe [HALL]). Mit obigen Beispielen existieren also speziell für $n = 2^k$, k = 1,2,.. solche Matrizen. Es läßt sich zeigen, daß für n > 3 die Ordnung einer Hadarmar-Matrix von der Form $4*k$, $k \in \mathbb{N}$ sein muß [HALL], und eine noch offene Vermutung geht

sogar von der Existenz für alle Ordnungen der Form 4∗k aus. Inzwischen konnte gezeigt werden, daß diese Vermutung zumindest für n < 268 richtig ist.

Lemma 8: Sei H eine Hadamar-Matrix der Ordnung n. Dann gilt: $|\det(H)| = n^{n/2}$

Beweis: Wie aus der linearen Algebra bekannt, gilt $\det(A) = \det(A^T)$ sowie $\det(A*B) = \det(A) * \det(B)$ (Determinantenmultiplikationssatz, siehe z.B. [FISCH] S 139 ff). Daher folgt:

$$|\det(H)| = \left(\det(H)*\det(H)\right)^{1/2} = \left(\det(H)*\det(H^T)\right)^{1/2} = \left(\det(H*H^T)\right)^{1/2} =$$

$$= \left(\det(n*I_n)\right)^{1/2} = \left(n^n\right)^{1/2} = n^{n/2}. \quad \square$$

Damit folgt aber, wie oben erwähnt, daß die Hadamar-Ungleichung für solche Matrizen scharf ist und somit die Abschätzung für die Determinanten von 0-1-Matrizen der Ordnung 2^k-1 ebenfalls scharf ist (sowie vermutlich auch für die Ordnungen 4∗k-1). Wird also für Satz 2 eine 0-1-Matrix der Ordnung d-1 mittels Satz 4 abgeschätzt, so folgt damit unmittelbar

SATZ 5: Sei $H \in HE(Q_d)$. Dann gilt für den kürzesten, ganzzahligen Orthogonalvektor a von H:

$$|a_i| \le 2*\left\lfloor \left(\frac{d-1}{4}\right)^{1/2} \right\rfloor, \; i = 1,..,d$$

Die folgende Tabelle gibt einen Überblick über die durch diesen Satz festgelegten maximalen Koordinatenwerte[12] von Orthogonalvektoren für die Dimensionen 1 bis 10.

Dimension	1	2	3	4	5	6	7	8	9	10
$2*\left\lfloor \left(\frac{d-1}{4}\right)^{1/2} \right\rfloor$	1,00	1,00	1,30	2,00	3,49	6,75	14,18	32,00	76,89	195,31
tatsächliche Grenze	1	1	1	2	3	5	9	**18**	≥ 42 ≤ 76	≥ 96 ≤ 195

Tabelle 2: Maximaler Betrag des Koordinatenwerts von ganzzahligen Orthogonalvektoren aus Q_d.

Wie sich später herausstellen wird, liegen die tatsächlich auftretenden Werte (hier in Zeile drei eingetragen) ab d = 6 unter den berechneten Grenzen. Besonders für d = 8 war dies überraschend, da hier ja die Abschätzung der Determinante scharf ist. Näheres dazu siehe Kapitel 4.2. ab Seite 46.

[12]Wird im weiteren Verlauf von maximalen Koordinatenwerten geschrieben, so ist immer deren Absolutbetrag zu verstehen.

3. Programmbeschreibung

3.1. Allgemeines

Ziel ist es, ein Programm zu entwickeln, daß für beliebige Dimensionen d alle Hyperebenen, die durch Ecken aus Q_d gebildet werden können, vollständig zählt und auflistet. Um aus den Resultaten möglichst einen Einblick in weitere Zusammenhänge zu erhalten, sollten soviele Zusatzinformationen wie möglich berechnet werden. Beispiele für solche weiteren Informationen sind der Typ der Hyperebene, der Orthogonalvektor, die Anzahl äquivalenter Typen oder die Anzahl der Hyperebenen mit derselben Eckanzahl aus Q_d.

Eine erste Realisierungsmöglichkeit liefert folgende, einfache Idee: Untersuche für alle möglichen $\binom{2^d}{d}$ d-Tupel aus Q_d, ob diese Ecken eine Hyperebene aufspannen. Falls dies der Fall ist, speichere den Orthogonalvektor als Kennung für diese Hyperebene bzw. überprüfe ob ein solcher Orthogonalvektor bereits behandelt wurde und erhöhe dann einen entsprechenden Zähler.

Etwas effizienter kann diese Berechnung durch die Tatsache gemacht werden, daß alle Ecken gleichberechtigt sind, d.h. daß es genügt, nur Hyperebenen, die durch den Nullpunkt laufen, zu zählen. Es sind also nur jene d-Tupel zu betrachten, in denen die Ecke (0,..,0) enthalten ist. Für jeden Typ einer Hyperebene ist dann deren Anzahl mit dem Faktor $\frac{2^d \text{ (=Alle Ecken aus } Q_d)}{\text{Anzahl Ecken auf der Hyperebene}}$ zu multiplizieren, um die Gesamtanzahl von Hyperebenen im Q_d zu erhalten.

Der Vorteil dieses Programms ist, daß es eine einfache Grundidee besitzt, und somit die Gefahr einer fehlerhaften Implementation gering ist. Dies ist für erste Versuche insofern nicht unwesentlich, als ja für die Ergebnisse von Dimensionen größer als vier keinerlei Vergleichswerte vorhanden sind. Der Nachteil des Programms ist klar: Voraussichtlich geringe Effizienz, da weitergehende theoretische Überlegungen nicht genutzt werden, sowie hoher Speicherbedarf, weil ja für alle Arten von Hyperebenen der Orthogonalvektor gespeichert werden muß. Dennoch ergaben die Berechnungen mit diesem Programm einige wertvolle Hinweise für die Entwicklung einer verbesserten Version. Berechnet werden konnten die (korrekten!) Werte bis zur Dimension 6, wofür ca. drei Stunden Rechenzeit (80386/33MHz mit 80387) notwendig waren. Eine Berechnung der Werte für die Dimension 7 hätte auf diesem Rechner mehrere hundert Jahre benötigt und wäre zudem mit einem zu hohen Speicherbedarf zum Speichern der Orthogonalvektoren verbunden gewesen (in der Größenordnung von 20 bis 25 MB für d=7 und ca. 10 GB für d=8 an Arbeitsspeicher). Somit schien eine Berechnung für höhere Dimensionen auch auf größeren Rechenanlagen mit diesem Programm recht aussichtslos.

Für eine neue Version lieferte das Programm folgende Erkenntnisse:

- Der Orthogonalvektor ist in seiner Größe beschränkt. Dies führte zur Entwicklung der theoretischen Schranken für den Betrag des maximalen Koordinatenwertes (siehe Kapitel 2.4.1.).

- Das Speichern und Vergleichen von Orthogonalvektoren oder ähnlichen Kennungen ist zu vermeiden, da es einerseits den Großteil der Rechenzeit benötigt (suchen bereits vorhandener Kennungen) und andererseits für höhere Dimensionen aufgrund des hohen Speicherbedarfs nicht realisierbar ist.
- Der Test, welche Ecken auf einer durch den Orthogonalvektor gegebenen Hyperebene liegen, ist mit dem gewöhnlichen Skalarprodukt sehr zeitraubend.

Obige Beobachtungen führten zur Entwicklung der zweiten, aktuellen Programmversion. Dabei wird zusätzlich von der Tatsache gebrauch gemacht, daß nur schräge Hyperebenen gezählt werden müssen, da k-parallele Hyperebenen sich daraus berechnen lassen (Vergleiche Satz 1). Die Grundidee des Programms ist folgende:

Erzeuge alle möglichen Orthogonalvektoren für schräge Hyperebenen aus Q_d, die durch den Nullpunkt gehen. Überprüfe, ob die, auf der dadurch definierten Hyperebene liegenden Ecken, diese Hyperebene als affine Hülle aufspannen. Falls dies der Fall ist, wurde eine neue Hyperebene gefunden und diese ist mit dem entsprechenden Multiplikationsfaktor aufzusummieren.

Weiters genügt es nach Lemma 4 (Seite 20), für alle Orthogonalvektoren mit denselben Koordinatenwerten nur eine einzige Permutation daraus stellvertretend zu untersuchen und das Ergebnis mit dem Multinomialkoeffizienten zu multiplizieren. Besonders diese Tatsache trägt viel zur höheren Effizienz des Programms bei!

Vorteile dieser Idee sind:

- Da jede in Frage kommende Hyperebene nur genau einmal untersucht wird, ist der Speicherbedarf gering! Im wesentlichen ist er abhängig von d, aber unabhängig von der tatsächlichen Anzahl von Hyperebenen, da außer des Aufsummierens der Anzahl keine weiteren Informationen über bereits behandelte Hyperebenen gespeichert werden müssen.
- Höhere Effizienz durch Ausnützung theoretischer Überlegungen.
- Leichtere Portierbarkeit des Programms, da durch den geringen Speicherbedarf komplizierte Speicherverwaltungsmethoden (meist sehr systemabhängig) wegfallen.

Die Programmstruktur läßt sich in vier Punkte unterteilen:

- **Generator:** Erzeugt alle (notwendigen) Orthogonalvektoren für schräge Hyperebenen die durch den Nullpunkt gehen.
- **Hyperebenen Test:** Überprüft, ob für den angegebenen Orthogonalvektor tatsächlich eine Hyperebene im Q_d erzeugt wird und berechnet gegebenenfalls. die Anzahl der darauf liegenden Ecken.
- **Hyperebenen Klassifizierung:** Die gefundenen Hyperebenen werden entsprechend summiert und mit bestimmten Merkmalen eingetragen.
- **Hyperebenen Auswertung:** Für die gefundenen Hyperebenen wird deren tatsächliche Anzahl berechnet, sie werden nach Typ sortiert und zur weiteren Auswertung ausgegeben.

Diese vier Punkte werden in den folgenden Abschnitten ausführlich dargestellt.

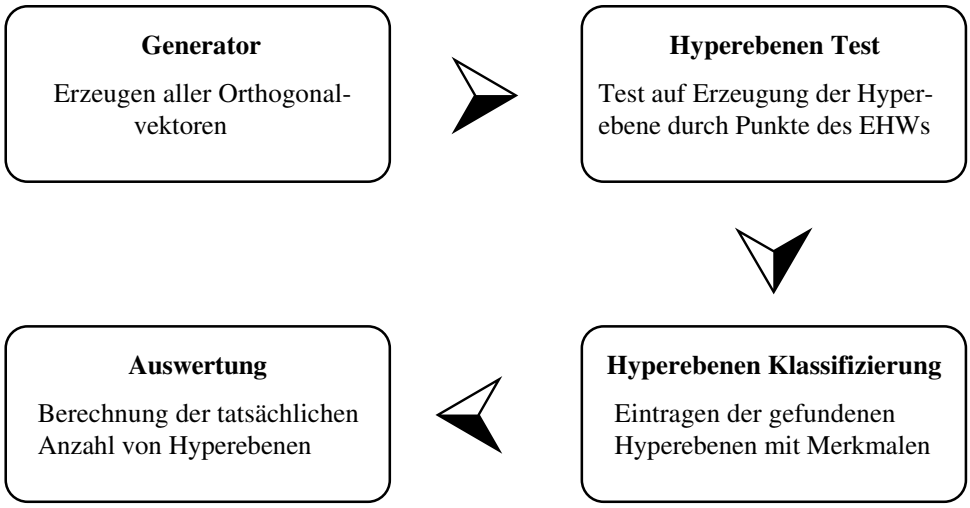


Abbildung 10: Grundstruktur des Programms

3.2. Die MULTI-FOR-Schleife

Im generierenden Teil des Programms werden alle Orthogonalvektoren innerhalb des angegebenen Bereiches (maximaler Betrag des Koordinatenwertes) erzeugt. Abhängig von der zu untersuchenden Dimension d ist es daher notwendig, alle d Koordinatenwerte des Vektors mittels ineinandergeschachtelter Schleifen so zu durchlaufen, daß alle möglichen Vektoren erzeugt werden. Dazu wäre eine Sprachkonstruktion ähnlich der folgenden Form notwendig (dabei entspricht die Anzahl von Schleifen (`anz`) gleich der Dimension d):

```
FOR i1 := anfang TO ende DO
     FOR i2 := anfang TO ende DO
          ....
               FOR ianz := anfang TO ende DO
                    { Schleifenkörper }
```

Die benötigte Anzahl von Durchläufen des Schleifenkörpers wäre dabei $(ende-anfang+1)^{anz}$.

Wie bereits früher erwähnt, ist es ausreichend, stellvertretend für alle Orthogonalvektoren mit denselben Vektorelementen nur einen speziellen Vektor zu betrachten und das Ergebnis dann mit der Anzahl solcher Vektoren zu multiplizieren (siehe dazu Lemma 4). Um anstatt jeder möglichen Permutation nur jeweils einen Vertreter zu erzeugen, genügt es, innerhalb der Vektorelemente Monotonie zu fordern. Für einen Orthogonalvektor a würde daher folgendes gelten: $a_1 \le a_2 \le .. \le a_d$. Der Vorteil dabei ist, daß wesentlich weniger Vektoren erzeugt und untersucht werden müssen (eine genaue Analyse dazu folgt weiter unten). Die dafür notwendige Sprachkonstruktion wird hier als *MULTI-FOR-Schleife* bezeichnet und wäre etwa von folgender Form:

```
FOR i1 := anfang TO ende DO
    FOR i2 := i1 TO ende DO
        ....
            FOR ianz := ianz-1 TO ende DO
                { Schleifenkörper }
```

Da aber die Dimension d einen Eingabeparameter des Programms darstellt, ist die Anzahl von ineinandergeschachtelten FOR-Schleifen ebenfalls variabel. Insbesondere ist zum Zeitpunkt der Programmübersetzung die Anzahl solcher Schleifen nicht festgelegt. Daher bieten die gängigen höheren Programmiersprachen keine direkte Unterstützung von Konstrukten der oben dargestellten Form.

Aus diesem Grund wurde ein eigener Programmteil entworfen, der eine variable Anzahl von ineinandergeschachtelten FOR-Schleifen entsprechend der obigen Konstruktion simuliert. Details zur Programmierung sowie zu den Aufrufmodalitäten sind im Listing von MULTI_FO.PAS zu finden.

Im folgenden wird die Anzahl der Durchläufe des Schleifenkörpers analysiert. Da alle Orthogonalvektoren von schrägen Hyperebenen untersucht werden sollen kann, o.B.d.A. der Schleifenanfang auf eins gesetzt werden. Die Anzahl von Durchläufen für die erste Konstruktion ist dann trivialerweise $(ende)^{anz}$. Zur Analyse der MULTI-FOR-Schleife wird eine noch allgemeinere Form untersucht, die für den Startwert jeder Schleife eine Differenz zum aktuellen Wert der übergeordneten Schleife zuläßt. Daß für den Startwert der ersten Schleife 1 festgelegt ist, bedeutet keine Einschränkung. Die Anzahl von Durchläufen einer MULTI-FOR-Schleife mit einem Anfangswert $anfang \neq 1$ und einem Endwert $ende$ entspricht nämlich genau der Anzahl einer MULTI-FOR-Schleife mit Anfangswert 1 und Endwert $(ende-anfang+1)$, da nur die Differenz $(ende-anfang)$ von Bedeutung ist.

Definition 14: Mit F(anz,ende,diff), anz ≥ 1, ende,diff ≥ 0, bezeichnet man die Anzahl von Durchläufen des Schleifenkörpers einer allgemeinen MULTI-FOR-Schleife der Form:

```
FOR i1 := 1 TO ende DO
    FOR i2 := i1 + diff TO ende DO
        ....
            FOR ianz := ianz-1 + diff TO ende DO
                { Schleifenkörper }
```

SATZ 6 $$F(anz,ende,diff) = \left(\frac{ende-(anz-1)*(diff-1)}{anz}\right) = \frac{\prod_{i=1}^{anz-1}(ende-(anz-1)*(diff-1)-i)}{anz!}$$

Beweis: Folgende Bedingungen müssen für F erfüllt sein:

i. F(1,ende,diff) = ende

ii. Für anz ≥ 2: $F(anz,ende,diff) = \sum_{i=1+diff}^{ende} F(anz-1,i-diff,diff)$ (Innerste Schleife explizit, Rest rekursiv) $\left(= \sum_{i=1}^{ende} F(anz-1,ende-(i+diff)+1,diff) \text{ (Äußerste Schleife explizit, Rest rekursiv)} \right)$

zu i.: $F(1,ende,diff) = \binom{ende-(1-1)*(diff-1)}{1} = \binom{ende}{1} = ende$ ✓

zu ii.: Es gilt $\sum_{k=0}^{n}\binom{k}{m} = \binom{n+1}{m+1}$ (s. z.B. [KNU2] S.54) Damit folgt $\sum_{i=1+diff}^{ende} F(anz-1,i-diff,diff) =$

$$\sum_{i=1+diff}^{ende}\binom{i-diff-(anz-2)*(diff-1)}{anz-1} \sum_{i=1+diff}^{ende}\binom{i-diff+anz-diff*(anz-2)-2}{anz-1}$$

$$\sum_{j=anz-diff*(anz-2)-1}^{anz+ende-diff*(anz-1)-2}\binom{j}{anz-1} \sum_{j=0}^{anz+ende-diff*(anz-1)-2}\binom{j}{anz-1} \underbrace{\sum_{j=0}^{anz-diff*(anz-2)-2}\underbrace{\binom{j}{anz-1}}_{0}}_{0}$$

(Der zweite Ausdruck ist Null, da $j \le anz-diff*(anz-2)-2 < anz-1$ für $anz \ge 2$ und $diff \ge 0$)

s.o. $= \binom{anz+ende-diff*(anz-1)-1}{anz} = \binom{ende-(anz-1)*(diff-1)}{anz} = F(anz,ende,diff)$ ❑

Einen schöneren, weil konstruktiven Beweis liefert die direkte Betrachtung aller möglichen Vektoren als Zahlen zur Basis "ende":

Beweis 2: Jede m-stellige Zahl (zur Basis b) $x = x_1\ x_2\ ..\ x_m$, $x_i \in \{0..b-1\}$ für die $x_i < x_j$ für $i < j$ gilt, wird durch eine MULTI-FOR-Schleife mit anf=0, ende=b-1, diff=1 und anz=m genau einmal erzeugt. D.h. man kann m verschiedene Zahlen (Ziffern von x) aus dem Bereich 0..b-1 auswählen, wobei die Reihenfolge keine Bedeutung hat, da die Zuordnung der Zahlen zur Stelle automatisch der Größe nach bestimmt ist (siehe Abbildung 11/A). Dies entspricht aber genau dem Binomialkoeffizienten $\binom{b}{m}$. Damit gilt für $F(anz,ende,1) = \binom{ende}{anz}$.

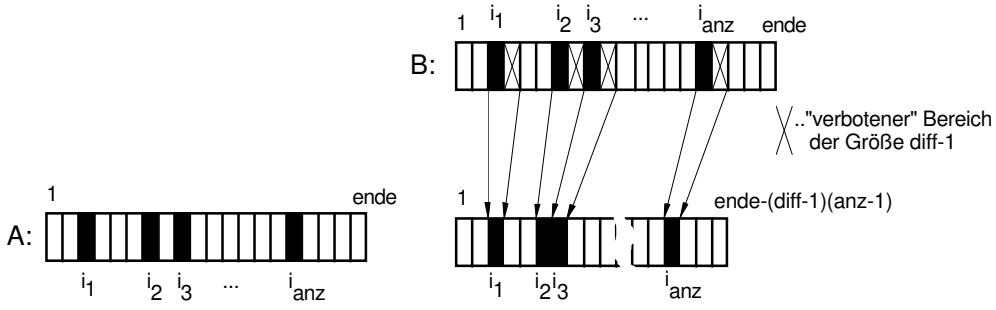


Abbildung 11: Auswahl von Elementen ohne (A) und mit Differenz (B).

Für den Fall diff ≠ 1 muß jede nachfolgende Schleife um den Wert (diff-1) später beginnen als für diff = 1. Dies ist aber (bzgl. der Durchlaufanzahl) äquivalent einer Verkleinerung des Schleifenendes um (diff-1). Für anz-1 Schleifen (für die erste Schleife ist diese Anpassung natürlich nicht notwendig!) ergibt sich daher der verkleinerte Auswahlbereich 0 bis ende-(anz-1)*(diff-1) (siehe Abbildung 11/B). Nach dieser Bereichsverschiebung gelten alle

vorherigen Überlegungen natürlich auch hier; d.h. es werden wiederum anz Elemente aus ende-(anz-1)*(diff-1) vorhandenen Elementen ausgewählt. Also gilt:

$$F(anz,ende,diff) = \binom{ende-(anz-1) * (diff-1)}{anz}$$

Man beachte, daß dies auch für diff=0 gilt! (Hier wird der Auswahlbereich sogar um anz-1 vergrößert, da ja dieselben Werte mehrfach verwendet werden dürfen). ❑

3.3. Generator

Die Aufgabe dieses Programmteiles ist es, alle möglichen Orthogonalvektoren schräger Hyperebenen im Q_d für die weitere Untersuchung zu erzeugen. Um die Auswertung solcher Vektoren möglichst effektiv zu gestalten, sollen dabei vom Generator nur jeweils die für eine Untersuchung unbedingt notwendigen Vektoren erzeugt werden. D.h., es genügt, stellvertretend für alle Orthogonalvektoren mit denselben Vektorelementen nur einen speziellen Vektor dieser Art zu betrachten. Das Ergebnis wird dann mit der Anzahl solcher Vektoren multipliziert (siehe dazu auch die Bemerkungen bei der MULTI-FOR-Schleife). Weiters soll der Vektor minimal im Sinne von Lemma 4 sein. Dazu wird der größte gemeinsame Teiler (ggT) der Vektorelemente mit dem Euklidschen Algorithmus berechnet (siehe z.B. [SEDG]), und Vektoren mit einem ggT größer als eins werden ausgeschieden. Nach Kapitel 2.4. genügt es, nur Orthogonalvektoren für Hyperebenen die durch den Nullpunkt gehen zu erzeugen, d.h. es werden nur Hyperebenen die, durch (a,0) definiert sind, betrachtet. Daher muß zumindest ein Koordinatenwert des Orthogonalvektors negativ sein (ansonsten könnten keine weiteren Punkte auf H liegen).

Natürlich darf für einen Vektor a der zu ihm negative Vektor -a nicht ebenfalls gezählt werden, da er ja dieselbe Hyperebene liefert. Dies kann erreicht werden, indem man maximal die Hälfte der Koordinatenwerte von a negativ sein läßt. Dabei ist noch der Spezialfall einer geraden Anzahl von Koordinatenwerten, von denen genau die Hälfte negativ ist, zu beachten, denn hier würde wiederum für jeden Vektor a sein negatives Gegenstück -a erzeugt werden. Dies kann aber vermieden werden, indem man für die positiven Koordinatenwerte nur Werte größer oder gleich den negativen Koordinatenwerten erzeugt. (Dabei ist größer im Sinne des Zählens zu verstehen, also z.B. 5,2,8 < 5,2,9 < 5,3,0). Hier ist noch der Spezialfall des zu sich selbst (genaugenommen zu einer Permutation seiner selbst) negativen Vektors zu beachten: Da hier je zwei Permutationen gezählt werden die aber zusammen nur je eine Hyperebene erzeugen, ist der Multinomialkoeffizient von Lemma 4 noch durch zwei zu dividieren (Beispiel: -1, -2, 1, 2).

Diese etwas komplizierten Zusammenhänge sollen durch folgendes Beispiel veranschaulicht werden: Für Dimension 4 werden alle zu erzeugenden Vektoren mit maximalen Koordinatenwerten von 3 aufgelistet. Dabei bedeutet [½] daß der Multinomialkoeffizient für den entsprechenden Vektor zu halbieren ist (zu sich selbst

negativer Vektor) und ✂ daß der Vektor nicht minimal ist[13] und daher nicht weiter untersucht zu werden braucht.

Beispiel für Q_4 mit maximalem Koordinatenwert 3:

-1	1	1	1		-2	1	2	2		-3	2	2	2		-1	-2	2	3					
-1	1	1	2		-2	1	2	3		-3	2	2	3		-1	-2	3	3					
-1	1	1	3		-2	1	3	3		-3	2	3	3		-1	-3	1	3	[½]				
-1	1	2	2		-2	2	2	2	✂	-3	3	3	3	✂	-1	-3	2	2					
-1	1	2	3		-2	2	2	3		-1	-1	1	1	[½]	-1	-3	2	3					
-1	1	3	3		-2	2	3	3		-1	-1	1	2		-1	-3	3	3					
-1	2	2	2		-2	3	3	3		-1	-1	1	3		-2	-2	2	2	[½]	✂			
-1	2	2	3		-3	1	1	1		-1	-1	2	2		-2	-2	2	3					
-1	2	3	3		-3	1	1	2		-1	-1	2	3		-2	-2	3	3					
-1	3	3	3		-3	1	1	3		-1	-1	3	3		-2	-3	2	3	[½]				
-2	1	1	1		-3	1	2	2		-1	-2	1	2	[½]	-2	-3	3	3					
-2	1	1	2		-3	1	2	3		-1	-2	1	3		-3	-3	3	3	[½]	✂			
-2	1	1	3		-3	1	3	3		-1	-2	2	2										

Tabelle 3: Zu erzeugende Orthogonalvektoren für d=4 und maxK=3

Die an den Generator gestellten Bedingungen lassen sich also wie folgt zusammenfassen:

> Erzeuge alle Vektoren $a=(a_1,a_2,..,a_d)^T$, die folgende Bedingungen erfüllen (A sei dabei die Menge aller solcher Vektoren):
>
> - $a_i \in Z$ für i=1,..,d
> - $|a_i| \leq$ Maximaler Koordinatenwert für i=1,..,d
> - $a_i \neq 0$ für i=1,..,d (nur schräge HE testen)
> - $ggT(a_1,a_2,..,a_d) = 1$
> - $a,b \in A \Rightarrow a \neq b, a \neq -b$
> - $a,b \in A \Rightarrow \underset{(i_1,i_2,..,i_d)\in I}{\forall} (a_{i_1},a_{i_2},...,a_{i_d})^T \neq b, (a_{i_1},a_{i_2},...,a_{i_d})^T \neq -b$ (Für alle möglichen Permutationen nur einen Repräsentanten erzeugen!)

Für die Implementation des Programms bedeutet dies nun, daß für die negativen und positiven Koordinatenwerte jeweils eine eigene MULTI-FOR-Schleife verwendet wird. Die MULTI-FOR-Schleife für die positiven Koordinatenwerte befindet sich dabei im Inneren der MULTI-FOR-Schleife für die negativen Werte. Durch eine äußere Schleife (FOR anzneg ...), die von 1 bis $\left\lfloor \frac{\text{Dimension}}{2} \right\rfloor$ läuft, wird dabei die Anzahl negativer (anzneg) bzw. positiver (Dimension-anzneg) Koordinatenwerte festgelegt. Für die MULTI-FOR-Schleifen bedeuten nun die obigen Betrachtungen der Spezialfälle folgendes: Für eine geradzahlige Dimension mit genau der Hälfte negativer Koordinatenwerte starten die Laufvariablen der positiven MULTI-FOR-Schleife

---

[13] D.h. der größte gemeinsame Teiler seiner Koordinatenwerte ist größer als Eins, womit es einen kürzeren Repräsentanten gibt.

genau mit den aktuellen Werten der Laufvariablen der negativen MULTI-FOR-Schleife. Zusätzlich wird dabei nach jedem solchen "künstlichen" Beginn der positiven MULTI-FOR-Schleife eine Markierung gesetzt, die für den jeweils ersten Durchlauf (dabei sind ja die positiven und negativen Laufvariablen äquivalent) eine Halbierung des Multiplikationsfaktors bewirkt.
Zusammengefaßt ergibt sich damit folgendes Schema für den Generator (d ist Dimension von Q_d):

```
GENERATOR für Dimension d und maximalen Koordinatenwert maxK:

FOR anzneg := 1 TO [illegible] DO {anzneg ≡ Anzahl negativer Koordinatenwerte}
  {MULTI-FOR-Schleife für negative Koordinatenwerte}
  FOR neg_1 := 1 TO maxK DO
    FOR neg_2 := neg_1 TO maxK DO
      ....
        FOR neg_anzneg := neg_anzneg-1 TO maxK DO
          {Testen, ob ein Sonderfall vorliegt}
          IF anzneg = d/2 THEN {Sonderfall}
            pos_i := neg_i für i:=1,..,anzneg
            Markierung für ½ setzen
          {MULTI-FOR-Schleife für positive Koordinatenwerte
          Falls ein Sonderfall vorliegt, beginnen die FOR-Schleifen
          beim ersten Durchlauf mit den Werten neg_1 bis neg_anzneg}
          FOR pos_1 := 1 TO maxK DO
            FOR pos_2 := pos_1 TO maxK DO
              ....
                FOR pos_d-anzneg := pos_d-anzneg-1 TO maxK DO
                  {Testen, ob Vektor minimal}
                  IF ggT(neg_1,..,neg_anzneg,pos_1,..,pos_d-anzneg) = 1 THEN
                    {evt. Vektor weiter untersuchen}
                    Test für Vektor  (-neg_1,..,-neg_anzneg,pos_1,..,pos_d-anzneg)
                  ½-Markierung zurücksetzen
```

Mit Hilfe der Analyse der MULTI-FOR-Schleife kann nun die vom Generator zu erzeugende Anzahl von Vektoren berechnet werden (ohne Berücksichtigung der ggT-Überprüfung). Für die Dimension d und den maximalen Koordinatenwert maxK soll diese Anzahl mit G(d,maxK) bezeichnet werden.

Fall 1: Sei d ungerade. Dann folgt aus der Analyse der MULTI-FOR-Schleife unmittelbar

$$G(d,\mathrm{maxK}) = \sum_{i=1}^{[illegible]} [illegible](i,\mathrm{maxK},0)\cdot F(d-i,\mathrm{maxK},0) \quad (*) .$$

Fall 2: Sei d gerade. Beim letzten Summanden obiger Formel (für i = d/2) ist nun der erwähnte Spezialfall zu beachten. Dabei läuft die innere MULTI-FOR-Schleife erst ab dem aktuellen Wert der äußeren MULTI-FOR-Schleife. D.h. ist die äußere Schleife im k-ten Durchlauf, so läuft die innere Schleife von k bis F(d/2,maxK,0), macht also F(d/2,maxK,0)-(k-1) Durchläufe. Für den letzten Summanden ergibt sich daher folgende Anzahl von Durchläufen:

$$\sum_{k=1}^{F\left(\frac{d}{2},maxK,0\right)} \left\{F\left(\frac{d}{2},maxK,0\right)\binom{k}{k-1}\right\} = F\left(\frac{d}{2},maxK,0\right) * \left\{F\left(\frac{d}{2},maxK,0\right)+1\right\} - \sum_{k=1}^{F\left(\frac{d}{2},maxK,0\right)} k =$$

$$= F\left(\frac{d}{2},maxK,0\right) * \left\{F\left(\frac{d}{2},maxK,0\right)+1\right\} - \frac{F\left(\frac{d}{2},maxK,0\right) * \left\{F\left(\frac{d}{2},maxK,0\right)+1\right\}}{2} =$$

$$= \frac{F\left(\frac{d}{2},maxK,0\right) * \left\{F\left(\frac{d}{2},maxK,0\right)+1\right\}}{2} =$$

$$= F\left(\frac{d}{2},maxK,0\right) * F\left(\frac{d}{2},maxK,0\right) - \frac{F\left(\frac{d}{2},maxK,0\right) * \left\{F\left(\frac{d}{2},maxK,0\right)+1\right\}}{2}$$

Für d gerade muß also von (*) der Ausdruck $\frac{F\left(\frac{d}{2},maxK,0\right) * \left\{F\left(\frac{d}{2},maxK,0\right)+1\right\}}{2}$ abgezogen werden.

Gesamt ergibt sich damit für die vom Generator zu erzeugende Anzahl von Vektoren die Formel:

$$G(d,maxK) = \sum_{i=1}^{} \binom{F(d,maxK,0)}{i} * F(d-i,maxK,0) - \underbrace{\frac{F\left(\frac{d}{2},maxK,0\right) * \left\{F\left(\frac{d}{2},maxK,0\right)+1\right\}}{2}}_{\text{falls d gerade}}$$

Nach Satz 6 gilt: $F(d,maxK,0) = \binom{maxK-(i-1)*(0-1)}{i} = \binom{maxK+i-1}{i}$ und damit[14]

$$G(d,maxK) = \sum_{i=1}^{} \binom{maxK+i-1}{i} * \binom{maxK+d-i-1}{d-i} - \underbrace{\frac{\binom{maxK+\frac{d}{2}-1}{\frac{d}{2}} * \left(\binom{maxK+\frac{d}{2}-1}{\frac{d}{2}}+1\right)}{2}}_{\text{falls d gerade}}$$

Die folgende Tabelle gibt die errechnete Anzahl von Vektoren für die Dimensionen 1 bis 10 an. Als maximaler Koordinatenwert wird dabei die nach Satz 5 berechnete obere Schranke verwendet.

---

[14]Natürlich wäre hier eine geschlossene Darstellung wünschenswert. Einen Ansatz dazu liefert möglicherweise $\sum_{k\geq 0} \binom{r-tk}{k}\binom{s-t(n-k)}{n-k}\frac{r}{r-tk} = \binom{r+s-tn}{n}$, ([KNU1],S.58).

Dimension d	maximaler Koordinatenwert $maxK = \lfloor 2^{d/2} \rfloor$	Anzahl von zu erzeugenden Orthogonalvektoren
1	1	-
2	1	1
3	1	1
4	2	14
5	3	105
6	6	5754
7	14	2612288
8	32	5258065252
9	76	7.49314952 E+13
10	195	1.25490692 E+19

Tabelle 4: Anzahl zu erzeugender Orthogonalvektoren für Q_1 bis Q_{10}

3.4. Hyperebenen Test

Aufgabe dieses Teils ist es, zu überprüfen, wieviele Ecken aus Q_d auf der durch den übergebenen Orthogonalvektor definierten Hyperebene liegen. Weiters wird getestet, ob die affine Hülle dieser Ecken auch tatsächlich der Hyperebene entspricht, oder ob durch lineare Abhängigkeit einiger Ecken nur ein Raum mit geringerer Dimension aufgespannt wird.

Bei unbeschränkten Koordinatenwerten von Orthogonalvektoren wäre die Frage, ob auf den durch Orthogonalvektoren gebildeten Hyperebenen überhaupt Ecken aus Q_d zu finden sind, NP-vollständig. Dies kann durch Reduktion des bekannten, NP-vollständigen Problems *Partition*[15] gezeigt werden, indem man es auf die Frage, ob es solche Ecken aus Q_d gibt, zurückführt. Die Dimension des Vektors entspräche dabei der Mächtigkeit der Menge, und die Koordinatenwerte des Orthogonalvektors wären dabei die Gewichte der Mengenelemente. Da Partition jedoch durch dynamic programming in pseudo-polynomialer Zeit lösbar ist, gilt die NP-Vollständigkeit für begrenzte Koordinatenwerte nicht.

Um eine vollständige Liste aller auf, über und unter H liegenden Ecken zu erhalten, muß für jede Ecke $x \in Q_d$ das Skalarprodukt mit dem Orthogonalvektor a gebildet werden; - ist es Null, so liegt x auf H, ist es größer als Null so liegt x über H und ansonsten darunter. Da dies für jeden Orthogonalvektor und dabei wiederum für jede der 2^d-1 Ecken (Nullecke ausgenommen) berechnet werden muß, ergibt sich damit eine der häufigsten Operationen des Programms. Betrachtet man nun die übliche Skalarproduktbildung, so sind dabei für d-dimensionale Vektoren d Multiplikationen und d-1 Additionen notwendig. Selbst wenn man berücksichtigt, daß hier ein Vektor nur aus 0 und 1 besteht (die Ecken), so benötigt man immer noch d Fallunterscheidungen

---

[15]Partition: Gegeben: Endliche Menge A und ein Gewicht $s(a) \in Z^+$ für alle $a \in A$.
Frage: Gibt es eine Teilmenge $A' \subseteq A$ sodaß $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$? [GAJO],[MOSH]

und bis zu d-1 Additionen. Eine Analyse mit einem Profiler bestätigt diese Vermutung: Bis zu 60% der Laufzeit des Programms werden hier für die Skalarproduktbildung benötigt! Daher scheint eine Optimierung dieser Operation mehr als gerechtfertigt. Überraschenderweise leistet hier eine Kodierung aus der Nachrichtentechnik gute Dienste.

3.4.1. Der Graycode

Neben der normalen Dualcodierung (gewöhnliche Binärzahlen) gibt es einen weiteren wichtigen Minimalcode, den Graycode [HRL], der häufig bei der Umwandlung analoger Größen in binäre Zeichen verwendet wird. Während beim Dualcode beim Übergang von einer Zahl zur nächsthöheren Zahl sich meist mehrere Stellen verändern, ändert sich beim Graycode jeweils nur eine einzige Stelle. Im Gegensatz zum Dualcode, wo sich der Wert bei Veränderung eines Bits um bis zur Hälfte des Zahlenbereiches ändern kann, ändert sich der Zahlenwert des Graycodes bei einer Unsicherheit von einem Bit maximal um ±1. Dies führt zu seiner besonderen Bedeutung für Übertragungszwecke und Analog/Digital-Umwandlungen. Tabellen 5 und 6 geben den Zusammenhang zwischen der gewohnten Dualcodierung und dem Graycode wieder.

	Dual-Code	Gray-Code
i	$d_2\ d_1\ d_0$	$g_2\ g_1\ g_0$
0	0 0 0	0 0 0
1	0 0 1	0 0 1
2	0 1 0	0 1 1
3	0 1 1	0 1 0
4	1 0 0	1 1 0
5	1 0 1	1 1 1
6	1 1 0	1 0 1
7	1 1 1	1 0 0

Tabelle 5: Gray- und Dualcode

Graycode ⇒ Dualcode	Dualcode ⇒ Graycode
$d_{n-1} = g_{n-1}$	$g_{n-1} = d_{n-1}$
$d_{n-2} = (d_{n-1} \neq g_{n-2}) = (g_{n-1} \neq g_{n-2})$	$g_{n-2} = (d_{n-1} \neq d_{n-2})$
$d_{n-3} = (d_{n-2} \neq g_{n-3}) = (g_{n-1} \neq g_{n-2} \neq g_{n-3})$	$g_{n-3} = (d_{n-2} \neq d_{n-3})$
....	....
$d_0 = (d_1 \neq g_0) = (g_{n-1} \neq g_{n-2} \neq ... \neq g_0)$	$g_0 = (d_1 \neq d_0)$

Tabelle 6: Umwandlung zwischen Gray und Dualcode. Ist die Bedingung in den Klammerausdrücken erfüllt, so ergeben diese 1, ansonsten 0.

Folgende Beobachtungen führen nun zur Verwendung des Graycodes als Ersatz für die Skalarproduktberechnung:

- Für eine d-stellige Binärzahl decken sowohl der gewöhnliche Dualcode als auch der Graycode alle 2^d möglichen Werte ab. Die Reihenfolge dieser Werte ist bei beiden Codes verschieden.
- Identifiziert man die Ecken aus Q_d mit einer d-stelligen Binärzahl, so ist die Reihenfolge, in der diese Ecken bezüglich der Hyperebene überprüft werden, unwesentlich. Der Graycode entspricht dabei einem Hamiltonschen Pfad im Q_d, d.h. es folgen stets nur benachbarte Ecken aufeinander.

- Betrachtet man zwei, in der Darstellung als Graycode um den Wert Eins verschiedene Ecken, so gilt für deren jeweiliges Skalarprodukt mit dem Orthogonalvektor a, daß sie sich genau um einen Koordinatenwert des Vektors a unterscheiden. Insbesondere ist dies der Wert jenes Vektorelementes, dessen Position in a der Position der unterschiedlichen Stelle der beiden Ecken entspricht. Dabei ist natürlich das Skalarprodukt mit jener Ecke, die an dieser Stelle eine Eins hat, um den Wert des Vektorelementes von a größer.

Damit läßt sich nun folgende, sehr effiziente Methode zur Ermittlung der Eckeigenschaften bezüglich der durch den Orthogonalvektor a gebildeten Hyperebene H angeben:

1. Das Skalarprodukt der Nullecke mit a ist 0.

2. Berechne das Skalarprodukt der Ecken 1 bis 2^d-1 iterativ:
   Sei k die Stelle, an der sich der Graycode beim Übergang von i-1 zu i ändert. Das Skalarprodukt der Ecke i (Graycode!) mit a entspricht dann dem Skalarprodukt der Ecke i-1 mit a, plus dem Element a_k falls sich eine 0 zu einer 1 geändert hat, bzw. minus dem Element a_k für den umgekehrten Fall.

i	gray[i]	graydiff[i]
0	0	-
1	1	1
2	3	2
3	2	-1
4	6	3
5	7	1
6	5	-2
7	4	-1

Tabelle 7: gray[i] und graydiff[i] für d=3.

Für das Programm bedeutet das nun, daß bei einer einmaligen (!) Initialisierung die Werte des Graycodes sowie die Position der Bitdifferenz zweier aufeinanderfolgender Zahlen in je einem Feld der Größe 2^d gespeichert werden (gray[i] und graydiff[i]). Das Feld gray[i] enthält dabei (in normaler Binärform) den Wert der i-ten Graycode-Zahl. Das Feld graydiff[i] gibt die veränderte Stelle des Graycodes beim Übergang von i-1 auf i an, das Vorzeichen von graydiff[i] bestimmt, ob nun das Vektorelement $a_{|graydiff[i]|}$ addiert (Übergang 0 → 1) oder subtrahiert (Übergang 1 → 0) werden muß. Damit sich das Vorzeichen auch für die letzte Stelle angeben läßt, werden die Stellen mit 1 bis d, anstatt, wie sonst üblich, mit 0 bis d-1 durchnumeriert.

Natürlich könnte der durch den Graycode gebildete Pfad auch durch ein rekursiv arbeitendes Programm durchlaufen werden, womit ein Speichern der Werte gray und graydiff nicht notwendig wäre. Aufgrund des dabei jedoch zwangsweise notwendigen höheren Rechenaufwandes (Fallunterscheidungen etc.) würde es zu einer Verschlechterung der Laufzeit kommen. Daher wurde der zusätzliche Speicherbedarf in Kauf genommen, um eine möglichst effektive Ausführung zu gewährleisten.

Als Formel gilt damit: Skalarprodukt(i) = <gray(i),a> = Skalarprodukt(i-1) + sign(graydiff(i)) * $a_{|graydiff(i)|}$. Da jede Ecke anhand des Skalarprodukts sofort klassifiziert werden kann, müssen die Werte nicht gespeichert werden und es genügt daher für das Skalarprodukt eine einfache Variable, die fortlaufend angepaßt wird.

Damit ergibt sich etwa folgendes Programmstück:

```
skalarprodukt := 0;
FOR i := 1 TO 2^d-1 DO
  INC(skalarprodukt,sign(graydiff[i])*a[ABS(graydiff[i])]);
  IF skalarprodukt = 0 THEN
    Ecke gray[i] liegt auf H
    gray[i] in liste[anzahl_ecken] eintragen;
  ELSE IF skalarprodukt > 0 THEN
    Ecke gray[i] liegt über H
  ELSE
    Ecke gray[i] liegt unter H
```

Der Zugriff `INC(skalarprodukt,sign(graydiff[i])*a[ABS(graydiff[i])])` ist hier jedoch noch unnötig zeitraubend und kann verbessert werden. Dazu definiert man einen "Doppelvektor" vek mit Elementen von -d bis +d und weist ihm die Werte des Orthogonalvektors a folgendermaßen zu: $vek_{-i}:=-a_i$, $vek_{+i}:=a_i$, i=1,..,d und $vek_0:=0$ (Das Element vek[0] wird hier nicht benötigt). Dann vereinfacht sich jetzt obiger Zugriff zu `INC(skalarprodukt,vek[graydiff[i]])`. Damit wird jetzt pro Skalarproduktberechnung nur mehr eine einzige Addition benötigt! Die zweifache Indexberechnung wird auch beim Produkt a[i] * x[i] oder ähnlichen Verfahren benötigt und stellt daher keinen Nachteil dar. Eine Analyse mit dem Profiler ergab, daß nun die Laufzeit des Programms gleichmäßig über alle zentralen Programmteile verteilt ist, die Optimierung also erfolgreich war.

Wurden alle auf H liegenden Ecken gefunden, so muß noch überprüft werden, ob diese Ecken tatsächlich eine Hyperebene aufspannen. Sind weniger als d Ecken vorhanden, so kann dies trivialerweise nicht der Fall sein, und die weitere Überprüfung entfällt. Ansonsten wird durch Zusammenfassung der gefundenen Ecken eine 0-1-Matrix gebildet und der Rang dieser Matrix bestimmt. (Diese Bestimmung erfolgt analog der Gaußelimination beim Lösen von linearen Gleichungssystemen [SEDG].) Ist der Rang gleich d-1, so wurde tatsächlich eine Hyperebene gefunden. Die Behandlung dieser Hyperebene wird sodann an den nächsten Abschnitt weitergereicht.

3.5. Hyperebenen Klassifizierung

Für jede gefundene Hyperebene wird nun die Gesamtzahl der zu ihr äquivalenten Hyperebenen im Q_d mittels des Multinomialkoeffizienten der Elemente des Orthogonalvektors nach Lemma 4 berechnet. Dieser Wert wird für zu sich selbst negative Vektoren noch halbiert, wozu die vom Generator gesetzte Markierung solcher Vektoren verwendet wird (vergleiche dazu die Beschreibung im Kapitel 3.3. ab Seite 32).

Die Anzahl von Ecken aus Q_d, die auf, über bzw. unter der Hyperebene liegen, kann direkt aus dem Test der Hyperebene übernommen werden, da sie dort ohnehin berechnet wurden. Für jeden möglichen Typ (j,p,k) einer Hyperebene ist ein Zähler [j,p] vorhanden, der ihre Anzahl aufsummiert. Zu diesem Zähler wird, entsprechend dem Typ der gefundenen Hyperebene, der ermittelte Faktor addiert. Ein weiterer, ebenfalls nach Typen unterteilter Zähler summiert die Anzahl verschiedener Orthogonalvektor-Arten auf. Dieser Zähler wird pro

gefundener Hyperebene um eins erhöht und enthält somit am Ende die Anzahl unterschiedlicher Orthogonalvektor-Arten für denselben Hyperebenen-Typ. Zusätzlich wird noch ein (redundanter) Zähler mitgeführt, der die Anzahl von Ebenen mit derselben Eckanzahl p aufsummiert. Dieser entspricht also der Summe der Zähler [j,p] mit demselben p.

3.6. Auswertung

Nachdem alle Hyperebenen klassifiziert wurden, kann nun die Berechnung der Gesamtanzahl von Hyperebenen im Q_d sowie die Ausgabe aller Daten erfolgen. Da bisher nur die durch den Nullpunkt gehenden Hyperebenen berücksichtigt wurden, müssen die Werte noch entsprechend Lemma 5 mit dem Faktor $\frac{2^d \text{ (=Alle Ecken aus } Q_d)}{p = \text{Anzahl Ecken auf der Hyperebene}}$ für jeden Typ (j,p,k) multipliziert werden. Die Anzahl aller schrägen Hyperebenen im Q_d, $e_0(d)$, ergibt sich dann als Summe der Ergebnisse für die einzelnen Typen.

Alle ermittelten Resultate können nun ausgegeben und somit zur Interpretation freigegeben werden...

Zusammengefaßt ergibt sich für das Hauptprogramm die auf der nächsten Seite abgebildete Struktur.

Generator

Erzeugt alle Vektoren $a=(a_1,a_2,..,a_d)^T$, die folgende Bedingungen erfüllen (A sei dabei die Menge aller solcher Vektoren):

i. $a_i \in Z$ für i=1,..,d

ii. $|a_i| \leq \max K_d$ für i=1,..,d

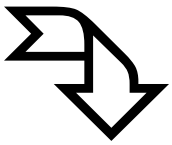
iii. $a_i \neq 0$ für i=1,..,d (nur schräge HE testen)

iv. $ggT(a_1,a_2,..,a_d) = 1$

v. $a,b \in A \Rightarrow a \neq b, a \neq -b$

vi. $a,b \in A \Rightarrow \forall_{(i_1,i_2,..,i_d)\in I} (a_{i_1},a_{i_2},...,a_{i_d})^T \neq b, (a_{i_1},a_{i_2},...,a_{i_d})^T \neq -b$

$\forall$ (a,0)



Hyperebenen Test

Bestimmt alle auf der Hyperebene (a,0) liegenden Ecken $q_i \in Q_d$:

$H := \{q_i \mid q_i \text{ e } Q_d, <q_i,a> = 0\}$

$h := |H|$

Falls $h \geq d$ bestimmen der Dimension des durch diese Eckpunkte aufgespannten Raums:

$r := \text{rang}\ (q_1,q_2,..,q_h),\ q_i \in H,\ i=1,..,h$

Falls $r \geq d-1$ kann die Hyperebene (a,0) durch Eckpunkte des Hyperwürfels aufgespannt werden und ist daher zu klassifizieren.

Abbildung 12: Struktur des Programms zur Untersuchung von Hyperebenen im Q_d.

Hyperebenen Klassifizierung

Die gefundene Hyperebene wird zur weiteren Auswertung mit folgenden Merkmalen eingetragen:

i. Erzeugender Vektor a

ii. Anzahl der auf dieser HE liegenden Eckpunkte von Q_d ($\equiv$ p)

iii. Typ (j,p,k)

iv. Vielfalt; Dies ist die Anzahl möglicher Permutationen der Skalare von a und entspricht dem bekannten Multinomialkoeffizienten (näheres siehe Text).

Falls r = d-1

Auswertung

Aus den eingetragenen Hyperebenen wird nun die gesuchte Anzahl schräger Hyperebenen im Q_d berechnet:

$$e_0(d) := \sum_{\forall \text{ eingetr. HE}} \frac{\text{Vielfalt(HE)} * 2^d}{p}$$

3.7. Programmtechnisches und Zeitverhalten

Das Programm wurde in TURBO PASCAL 6.0 entwickelt und ist auf jedem kompatiblen PC mit 640 KB Speicher lauffähig. Die Benutzerschnittstelle wurde entsprechend den Programmanforderungen rudimentär gehalten, da es sich um ein reines Hilfsprogramm für Berechnungen handelt.

Der Quelltexte wurde in die drei Units MULTI_FO.PAS, TOOLS1.PAS und TOOLS2.PAS sowie das Hauptprogramm HYPER.PAS aufgeteilt. Die Aufgaben der Units sowie die genaue Beschreibung der einzelnen Routinen sind in den entsprechenden Listings zu finden. Eine ebenfalls verwendete Unit TIMER.PAS ist für die Programmfunktion selbst nicht notwendig; sie stellt lediglich eine Art Stoppuhr zur Verfügung, die für lange Berechnungen eine Abschätzung der notwendigen Zeit und damit des Berechnungsendes ermöglicht.

Zur Optimierung des Laufzeitverhaltens wurde das Programm mit einem Profiler analysiert und entsprechend verbessert (siehe z.B. die Bemerkungen über die Berechnung des Skalarprodukts mittels Graycode). Eine abschließende Analyse ergab, daß die Laufzeitverteilung relativ gleichmäßig (je ca. 20 bis 30 %) auf die vier Blöcke: Generator, Ecktest, Rangtest und ggT-Berechnung entfällt. Da kein Block im Vergleich zu seiner Aufgabe und Größe unverhältnismäßig viel Zeit beansprucht, ist dieses Ergebnis durchaus zufriedenstellend.

Die meisten Auswertungen wurden auf einem PC mit 80386 und 80387 Prozessoren mit 33 MHz vorgenommen. Dabei ist der Zeitbedarf für Dimensionen bis einschließlich fünf praktisch nicht meßbar. Selbst für d = 6 benötigt das Programm für den theoretisch maximalen Koordinatenwert maxK=6 weniger als zwei Sekunden zur vollständigen Berechnung. Da 6-dimensionale 0-1-Matrizen eine maximale Determinante von 9 aufweisen, kann für den 7-dimensionalen Fall ein maxK=9 verwendet werden. Die Berechnungen sind dann nach ca. 105 Sekunden abgeschlossen. Aber selbst für die schwächere obere Schranke maxK=14, wie sie aus Satz 5 folgt, werden weniger als 21 Minuten benötigt.

Zeitintensiv wird die Berechnung für d = 8. Der maximale Koordinatenwert liegt hier bei 32. Einige Testläufe ließen aber bereits eine tatsächliche Grenze von maxK=18 vermuten. Erste Berechnungen mit d=8 / maxK=18 benötigten knapp 16 Stunden. Eine weitere Berechnung mit d=8 / maxK=20 wurde auf einem PC 80486 mit 33 MHz (ca. doppelt so schnell wie der 80386) gestartet und benötigte dafür 18:15 Stunden. Eine Hochrechnung ergab für den zu untersuchenden Fall von d=8 / maxK=32 eine Laufzeit von über 25 Tagen. Zuviel um einen PC so lange zu blockieren! Das Programm wurde daher zusätzlich in C übersetzt und auf eine Workstation (DEC Station 5000/240) portiert. Nach mehreren Testläufen mit zufriedenstellenden Ergebnissen wurde das Programm für d=8 und maxK=32 gestartet. Nach 12 Tagen und 9 Stunden waren die Berechnungen beendet und bestätigten die Vermutung, daß der größte auftretende Koordinatenwert eines Orthogonalvektors 18 ist. Die Datei mit den berechneten Werten hat dabei einen Umfang von über 8 MB und liefert dieselben Resultate wie bei der Berechnung mit maxK=18.

Und Dimensionen größer als acht?

Um Berechnungen ab der Dimensionen neun durchführen zu können, muß mittels der Zeile `{$DEFINE keine_auswertung}` am Programmanfang die Erweiterte Auswertung abgeschaltet werden.

Dadurch werden die Hyperebenen nicht mehr jeweils nach Typ (j,p,k) getrennt gezählt, sondern nur mehr nach der Anzahl der auf ihnen liegenden Ecken p. Dies ist aus Speichergründen notwendig und für die Berechnung von $e_0(d)$ vollkommen ausreichend. Allerdings gehen dadurch natürlich Zusatzinformationen verloren, die evt. weitere Interpretationen ermöglicht hätten. Dies ist aber aus zwei Gründen vertretbar:

- Bereits die Daten aus Q_8 haben einen Umfang von über 8 MB, wer könnte wohl die zu erwartenden Daten von mehreren Gigabyte aus Q_9 vollständig durchsehen?
- Die Laufzeit des Programms würde sehr viel Geduld benötigen! Nimmt man die Anzahl der Generatorschritte als Maß für die Rechenzeit und extrapoliert man den Bedarf von Q_8, so erhält man folgende Werte:

Dimension d / maxK	Generatorschritte	Zeitbedarf
8 / 32	$5{,}2581\ 10^9$	12 Tage 9 Stunden ☺
9 / 76	$7{,}4391\ 10^{13}$	≈ 479 Jahre 😐
10 / 195	$1{,}2549\ 10^{19}$	≈ $81\ 10^6$ Jahre ☹

Tabelle 8: Zeitbedarf für höhere Dimensionen

Aus diesen Gründen scheint eine enumerative Berechnung der Werte für Dimensionen größer als acht derzeit nicht möglich. (Aber bei der ersten Programmversion schien dies für Dimensionen größer als sechs ebenso, und nun werden für d=7 statt 200 Jahren nur knapp 100 Sekunden benötigt ....)

3.8. Bedienung des Programms

Die Bedienung des Programms ist denkbar einfach. Nach dem Programmstart werden die gewünschten Parameter eingegeben, wonach die Berechnung beginnt. Ist diese abgeschlossen, so ertönt ein akustisches Signal und die Ausgabe erfolgt.

Folgende Eingaben werden abgefragt:

1. `Dimension (2..8) =` ⇒ Eingabe der zu untersuchenden Dimension innerhalb des angegebenen Bereichs.
2. `Maximaler Koordinatenwert des Orthogonalvektors =` ⇒ Eingabe des Betrages des maximalen Koordinatenwertes.
3. `Soll die Auswertung auf den Bildschirm, Drucker oder in ein File erfolgen (B/D/F) ?` ⇒ Je nach gewünschter Ausgabeart kann hier durch drücken der Tasten B (für Bildschirm), D (für Drucker) oder F (für Datei (File)) die Ausgabe entsprechend umgeleitet werden. Erfolgt die Ausgabe an den Drucker, so wird der Druckerport abgefragt: `Parallele Schnittstelle 1 oder 2 ?` ⇒ Durch drücken der Taste 1 (für LPT1) oder 2 (für LPT2) wird die gewünschte

Schnittstelle gewählt. Wurde eine Datei zur Ausgabe gewählt, so muß deren vollständiger Name angegeben werden: `Dateiname:` ⇒ Eingabe des vollständigen Dateinamens.

4. `Erweiterte Auswertung (vollst. Liste aller Orthogonalvektoren) (J/N)` ⇒ Soll eine Liste aller Orthogonalvektor-Arten ausgegeben werden, so ist hier J zu drücken, ansonsten N. (Achtung, die vollständige Liste benötigt relativ viel Speicher, z.B. für d=8 ca. 8 MB.)

5. `Zeitanzeige während des Berechnungsvorganges (J/N)` ⇒ Ist eine Zeitanzeige erwünscht, so ist hier J zu drücken, ansonsten N. Bei eingeschalteter Zeitanzeige wird ca. nach jedem zweitausendsten überprüften Vektor die bisher benötigte Laufzeit, der bereits berechnete Anteil in % und die voraussichtlich benötigte Gesamtzeit[16] ausgegeben.

Nach der letzten Eingabe beginnt sofort die Berechnung und gegebenenfalls die Ausgabe der Liste aller Orthogonalvektor-Arten. Nach dem Ertönen des Endsignals erfolgt die Ausgabe der Ergebnisse auf das gewünschte Medium. Bei der Ausgabe auf den Bildschirm wird nach jeweils 20 Zeilen auf einen Tastendruck gewartet.

Beispiele für die Ausgabe des Programms sind in Kapitel 6. ab Seite 90 zu finden

[16]Die Angabe der Gesamtzeit ist lediglich ein Richtwert. Erfahrungsgemäß übersteigt die tatsächlich benötigte Zeit die ersten Abschätzungen um bis zu 50%.

4. Auswertung und Interpretation der Daten sowie theoretische Sätze

»Holmes«, rief ich, »das ist unmöglich.«
»Bewundernswert!« sagte er. »Eine höchst lichtvolle Bemerkung. Es ist unmöglich, so wie ich es formuliere, also muß ich es irgendwie falsch formuliert haben. ...«

SIR ARTHUR CONAN DOYLE
In: *Das Abenteuer der Priorschule*

Dieser Absatz behandelt die durch das Programm gewonnenen Informationen. Eine Liste der berechneten Daten ist im Kapitel 6. ab Seite 90 zu finden. Die wichtigsten Zahlenwerte sind in Kapitel 7. ab Seite 109 in Tabellenform zusammengefaßt. Die folgenden Abschnitte behandeln, jeweils nach Thema sortiert, entsprechende Erkenntnisse aus diesen Daten.

4.1. Quantitative Auswertung

Betrachtet man die in den Tabellen in Kapitel 7. zusammengefaßten Ergebnisse der Berechnungen, so ergeben sich daraus über die Anzahl möglicher Hyperebenen im Q_d folgende Schlüsse:

- Die Anzahl schräger Hyperebenen dominiert die Anzahl k-paralleler Hyperebenen für $k \geq 1$. Mit zunehmender Dimension verlieren daher bei der quantitativen Betrachtung k-parallele Hyperebenen an Bedeutung. In Q_8 beträgt z.B. ihr Anteil bereits weniger als 0,7 %. Dies entspricht auch dem in Satz 1 gezeigten Zusammenhang.
- Die Anzahl möglicher Hyperebenen wächst tatsächlich überexponentiel ($\approx 2^{d^2}$). Vergleicht man damit die triviale obere Schranke von $\binom{2^d}{d}$, so zeigt sich, daß diese Schranke nicht so überhöht ist, wie ihre Herleitung vermuten hätte lassen.
- Bei den schrägen Hyperebenen überwiegen jene mit einer möglichst geringen Anzahl von Eckpunkten aus Q_d. So liegen bis zur achten Dimension auf über 30 % der schrägen Hyperebenen aus Q_d nur genau d Eckpunkte. Dies erklärt auch, warum die vorher erwähnte triviale Schranke so gut ist, denn sie wurde ja unter Verwendung solcher Hyperebenen hergeleitet. Da mit zunehmender Dimension dieser Anteil abnimmt, dürfte diese Schranke für höhere Dimensionen immer schlechter werden. Im Q_8 besitzen aber noch über 92 % aller Hyperebenen zwischen 8 und 12 Eckpunkte (von maximal 128 pro Hyperebene).

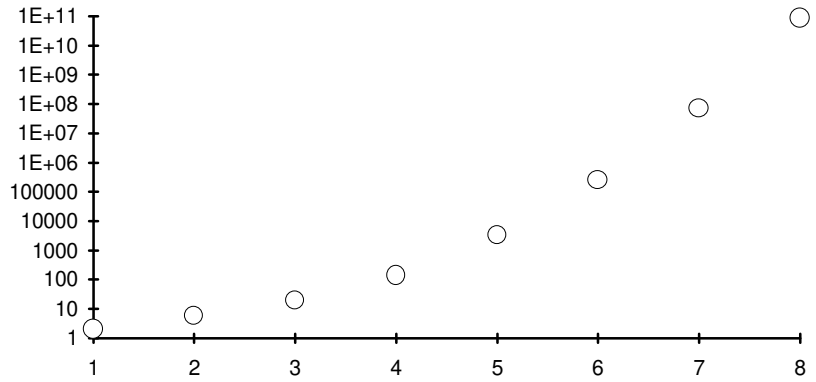
Betrachtet man die verschiedenen Arten von Hyperebenen, so finden sich in jeder Dimension neue Typen, welche sich nicht aus Typen tieferer Dimensionen aufbauen lassen - dazu mehr in den folgenden Kapiteln.

Aber auch das Wachstum innerhalb derselben Art von Hyperebenen ist exponentiel, wie das folgende Beispiel veranschaulicht:

Der Hyperebenen-Typ $(1,d,2^d\text{-}d\text{-}1)$ tritt in allen Dimensionen $d \geq 2$ auf und kann folgend konstruiert werden: Für einen Eckpunkt aus Q_d, z.B. den Nullpunkt (0,..,0), bilden die d Ecken aus Q_d mit einem Raumabstand von eins zu dieser Ecke (das sind hier (1,0,..,0), (0,1,0,..,0), .. ,(0,..,0,1)) eine d-1 dimensionale Hyperebene H. Wie man leicht sieht, sind das auch die einzigen Eckpunkte, die auf H liegen können (Betrachte dazu für das Beispiel den Orthogonalvektor a = (1,..,1) und b=1.)[17].

Man kann also genau um jede Ecke aus Q_d eine Hyperebene von Typ $(1,d,2^d\text{-}d\text{-}1)$ bilden. Damit gibt es aber im Q_d zumindest 2^d solcher Hyperebenen, womit das exponentielle Wachstum allein dieses Typs gezeigt wäre.

Das folgende Diagramm veranschaulicht das Wachstum der Anzahl von Hyperebenen im Q_d:



Logarithmische Darstellung der Anzahl von Hyperebenen im Q_d

4.2. Länge der Orthogonalvektoren

Aufgrund der Untersuchungen von Kapitel 2.4. konnte eine Grenze für den maximalen Betrag der Koordinatenwerte eines Orthogonalvektors gefunden werden. Diese Grenze wurde mit Hilfe von Determinanten von Matrizen ermittelt und ist für diese Matrizen für gewisse Stützpunkte $(2^k\text{-}1,k=1,2,..)$ scharf. Mit Hilfe eines Programms wurden die maximalen Determinanten von allen möglichen 0-1-Matrizen bis zur Dimension 6 ermittelt. Für die Dimension 7 kann man eine 0-1-Matrix aus der Hadamar-Matrix für die Dimension 8 konstruieren (vergleiche dazu den Beweis zu Satz 4), für welche die Determinante maximal wird.

---

[17]Diese Hyperebene H entspricht übrigens der in der Einleitung bereits erwähnten Verallgemeinerung des Tetraeders auf höhere Dimensionen. Da alle auf H liegenden Eckpunkte zueinander eine Hamming-Distanz von 2 aufweisen, bilden sie einen regelmäßigen Hypertetraeder mit der Kantenlänge $\sqrt{2}$.

Hadamar-Matrix für Dimension 8:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \text{ und } |\det(H)| = 8^{8/2} = 4096$$

Multipliziert man die Zeilen 2 bis 8 mit -1 und addiert man die erste Zeile zu jeder dieser Zeilen, so erhält man eine Matrix mit 1,1,..,1 in der ersten Zeile und 0/2 in den restlichen Zeilen. Dividiert man jetzt die Zeilen 2 bis 8 durch 2 (dadurch wird die Determinante um den Faktor 2^7 gekürzt), so ergibt sich folgende 0-1-Matrix:

$$H' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \text{ und } |\det(H')| = \frac{4096}{2^7} = 32$$

Deren rechte untere 7x7-Teilmatrix A ist eine 0-1-Matrix der Ordnung 7 mit derselbe Determinante wie die Gesamtmatrix (Entwicklung nach der ersten Spalte). Damit ist eine 0-1-Matrix A der Ordnung 7 gefunden, für die |det(A)| = 32 den maximal Wert annimmt.

Für die Ordnung 8 konnten Beispiele von 0-1-Matrizen mit Determinanten bis zu 42 gefunden werden. Möglicherweise existieren aber noch Matrizen mit größeren Determinanten, weshalb dieser Wert nur eine untere Schranke bildet. Ein Beispiel für einen entsprechenden Orthogonalvektor für Q_9 ist in den Tabellen im Abschnitt 7. zu finden.

Für den maximalen Betrag von Determinanten von 0-1-Matrizen ergibt sich somit folgende Tabelle:

Dimension	1	2	3	4	5	6	7	8
$2*\frac{d+1}{4}^{(d+1)/2}$	1,00	1,30	2,00	3,49	6,75	14,18	32,00	76,89
tatsächliche Grenze	1	1	2	3	5	9	32	≥ 42

Tabelle 9: Maximaler Betrag der Determinante von 0-1-Matrizen der Ordnung d.

Für die Ordnungen 1 bis 6 stimmen die tatsächlichen Grenzen für Determinanten auch mit den gefundenen maximalen Koordinatenwerten von Orthogonalvektoren der Dimensionen 2 bis 7 überein. (Dimensionsverschiebung um +1 beachten!) Für die 7x7 Matrix gibt es jedoch ein einfaches Beispiel mit der maximalen

Determinante 32. Dennoch ergab die vollständige Untersuchung aller möglichen Orthogonalvektoren für Q_8 (bis hin zum theoretisch maximalen Koordinatenwert von 32) als größten, tatsächlich auftretenden Koordinatenwert 18. Da alle Hyperebenen untersucht wurden, entspricht dies einer vollständigen Betrachtung aller 0-1-Matrizen A der Ordnung 7 mit beliebiger rechter Seite b und Rang(A|b) = 7 (alle möglichen Gruppen von 7 linear unabhängige Ecken aus Q_8)[18]. Das bedeutet aber, daß für alle eindeutig lösbaren linearen Gleichungssysteme der Ordnung sieben mit 0-1-Koeffizienten die Elemente des Lösungsvektors der Form $\frac{m}{n}$ im Zähler und Nenner durch $|m| \leq 18$ und $|n| \leq 18$ beschränkt sind[19]. Vergleicht man dies mit der Regel von Cramer zum Lösen von linearen Gleichungssystemen, so folgt daraus, daß alle dabei auftretenden Determinanten (welche ja bis zu 32 betragen) sich immer derart kürzen, daß kein Bruch mit Nenner oder Zähler größer als 18 zurück bleibt! Das erstaunt umso mehr, da es über $3{,}2\ 10^9$ solcher Gleichungssysteme gibt (≡ Anzahl schräger Hyperebenen im Nullpunkt im Q_8), und in keinem einzigen davon solche Kürzungen nicht möglich sind!

Untersucht man zum Beispiel für die oben hergeleitete 7x7-Matrix A das lineare Gleichungssystem der Form Ax=b für alle 2^7=128 möglichen Vektoren b, so erhält man den maximalen Lösungsvektor für b = (0,0,0,0,1,1,1) mit x = ¼*(1,1,1,3,-1,-1,-1). Dies ergibt dann für den Q_8 den Orthogonalvektor (1,1,1,3,-1,-1,-1,-4), der eine Hyperebene durch den Nullpunkt mit den Ecken (1,0,1,0,1,0,1,0), (0,1,1,0,0,1,1,0), (1,1,0,0,1,1,0,0), (0,0,0,1,1,1,1,0), (1,0,1,1,0,1,0,1), (0,1,1,1,1,0,0,1), (1,1,0,1,0,0,1,1) und (0,0,0,0,0,0,0,0) (entsprechend $\left[\begin{array}{c|c} A & b \\ \hline 0\ldots,0 & 0 \end{array}\right]$) sowie weiteren 28 Ecken aus Q_8 erzeugt. D.h. obwohl det(A) = -32 ist, liefert die Lösung des linearen Gleichungssystems maximal den Faktor 4.

Daraus folgt auch, daß die Abschätzung des maximalen Koordinatenwertes durch Determinanten nicht optimal ist, denn obwohl die Grenze für Determinanten scharf ist, ist sie dies für die entsprechenden Koordinatenwerte offensichtlich nicht. Eine scharfe Grenze müßte gleichzeitig auch die Lösungen von linearen Gleichungssystemen mit 0-1-Koeffizienten einschränken, und wäre daher eventuell auch für die Gebiete der linearen Algebra oder diskreten Optimierung interessant.

---

[18]Die entsprechende Hyperebene in Q_8 ist dann die affine Hülle dieser sieben Ecken plus dem Nullpunkt.

[19]Dies gilt daher analog auch für alle homogenen, linearen Gleichungssysteme der Ordnung acht mit 0-1-Koeffizienten, die einparametrig lösbar sind (Vergleiche Kapitel 2.4.1.).

4.2.1. Abschätzung der Anzahl von Hyperebenen mittels Orthogonalvektoren

Natürlich gibt es im Q_d nur so viele verschiedene Hyperebenen durch den Nullpunkt, wie zugehörige Orthogonalvektoren existieren. Daher liefert eine Abschätzung der Anzahl von Orthogonalvektoren zugleich eine Schranke für die Anzahl von Hyperebenen. Folgende Beobachtungen der vorhergehenden Kapitel für einen Orthogonalvektor a einer schrägen Hyperebene aus Q_d können dafür verwendet werden:

- $a_i \neq 0$, i=1,..,d (nur schräge Hyperebenen)
- $|a_i| \leq 2*\left(\frac{d}{4}\right)^{d/2} =: \max K$, i=1,..,d (nach Satz 5)
- O.b.d.A. : $a_1 > 0$ (um zueinander inverse Vektoren nicht doppelt zu zählen)

Damit läßt sich für die Anzahl von Orthogonalvektoren bzw. für durch den Nullpunkt gehende schräge Hyperebenen, folgendes Lemma zeigen:

Lemma 9: Die Anzahl von schrägen Hyperebenen durch den Nullpunkt im Q_d, d>1, ist mit

$$\left|\{H \in HE(Q_d) \mid H \text{ schräg},\ 0 \in H\}\right| \leq 2^{2d-d^2-1} * d^{d^2/2} \text{ beschränkt.}$$

Beweis: Betrachtet man alle möglichen Orthogonalvektoren mit Koordinatenwerten innerhalb der Grenzen ±maxK sowie der Einschränkung $a_i \neq 0$, i=1,..,d und $0 < a_1 \leq \max K$ so folgt:

$$\text{Anzahl Orthogonalvektoren} \leq \max K * \ldots * \max K^{d-1} \leq 2*\left(\frac{d}{4}\right)^{d/2} * \left(\left(\frac{d}{4}\right)^{d/2}\right)^{d-1} =$$

$$\frac{2*4^{d-1}}{4^{d/2}*4^{\frac{d}{2}*(d-1)}} * d^{d/2} * d^{\frac{d}{2}*(d-1)} = 2^{2d-d^2-1} * d^{d^2/2} \qquad \square$$

Leider läßt sich diese Schranke nicht auf alle Hyperebenen aus Q_d direkt erweitern, da es schräge, zueinander parallele Hyperebenen gibt, welche denselben Orthogonalvektor haben[20]. Für ganz Q_d müßte daher noch mit der Anzahl vorhandener Ecken (2^d) multipliziert werden oder besser eine Obergrenze für die Anzahl zueinander paralleler Hyperebenen gefunden werden. Auch liefert diese Schranke zwar für niedere Dimensionen gute Werte, zeigt aber für höhere Dimensionen ein asymptotisch schlechteres Verhalten als die triviale Obergrenze von $\binom{2^d}{d}$. Eine Verbesserung der Abschätzung für den maximalen Koordinatenwert hätte natürlich auch für diese Betrachtungen eine Verbesserung der Schranke zur Folge.

[20] In Q_3 sind z.B. die durch (a=(1,1,1), b=1) und (a=(1,1,1), b=2) definierten, schrägen Hyperebenen zueinander parallel.

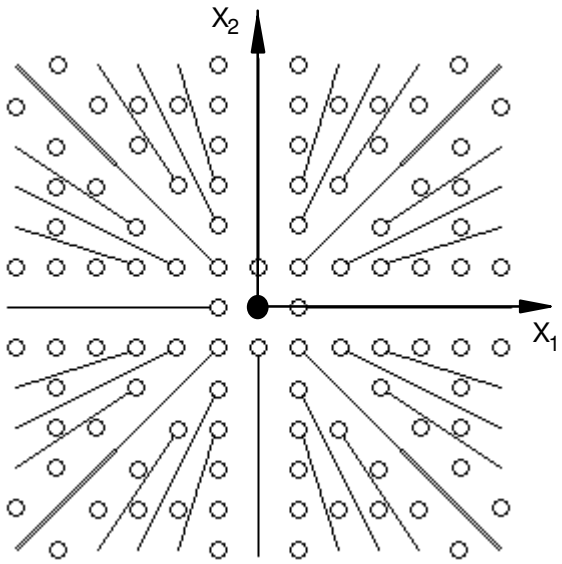


Abbildung 13: Redundante, ganzzahlige Vektoren.

Eine weitere Überlegung wäre es, die Kürzbarkeit von Orthogonalvektoren zu berücksichtigen, d.h. ein Vektor (2,4,-2,-4,-4) sollte nicht gezählt werden, da er ja durch (1,2,-1,-2,-2) bereits abgedeckt ist. In Abbildung 13 wird dies für d=2 und maxK=6 dargestellt. (Die Kreise entsprechen dabei den Orthogonalvektoren und die Linien deuten die dadurch abgedeckten Bereiche an.) Für niedere Dimensionen (2,3) bringt dies eine gewisse Verschärfung der Schranke; für den interessanteren, höherdimensionalen Bereich ist der Gewinn aber minimal! Durch vollständiges Abzählen mittels eines Programms wurden folgende Werte ermittelt: für d=7, maxK=9 werden nur 0,63 % und für d=8, maxK=18 gar nur 0,51 % der Orthogonalvektoren eingespart. Eine genauere Analyse der Kürzbarkeit kann daher vernachlässigt werden.

4.2.2. Maximaler Skalarwert b der Vektordarstellung einer Hyperebene

Bisher wurde für die Darstellung (a,b) einer Hyperebene gezeigt, daß der Betrag der Elemente des Orthogonalvektors a nach oben beschränkt ist. Dieser Abschnitt liefert ein analoges Ergebnis für den Skalarwert b. Dazu wird nochmals die bereits auf Seite 24 erklärte Transformation einer beliebigen Hyperebene in den Nullpunkt auf Veränderungen des Wertes von b näher untersucht:

Für eine Hyperebene $H \in HE(Q_d)$ mit der Vektordarstellung (a,b) betrachte man eine beliebige Ecke $x \in H$. Für alle Koordinatenrichtungen, an denen x = 1 gilt, wird die Hyperebene am Mittelpunkt des Würfels gespiegelt. D.h. für alle Ecken auf H wird in dieser Koordinate eine 0 zur 1 und eine 1 zur 0. Weiters ändert der Orthogonalvektor für diesen Koordinatenwert sein Vorzeichen. Der Skalarwert b vermindert sich dabei um den Wert des Orthogonalvektors (vor dem Vorzeichenwechsel) an dieser Koordinatenstelle. Wird dieser Vorgang für alle Einsen aus x wiederholt, so wird x schließlich zur Nullecke und man hat nun eine (zur ursprünglichen Hyperebene äquivalente) Hyperebene durch den Nullpunkt.

Da diese Konstruktion umkehrbar ist, kann jede Hyperebene und somit jeder Skalarwert b aus einer Hyperebene durch den Nullpunkt (b=0) konstruiert werden. Die Veränderung von b hängt nur von den Koordinatenwerten des Orthogonalvektors ab; daher gilt folgender Zusammenhang: Der Betrag von b kann maximal dem Betrag der Summe aller positiven Elemente oder dem Betrag der Summe aller negativen Elemente eines Orthogonalvektors für eine Hyperebene im Nullpunkt entsprechen.

So können z.B. für $H \in HE(Q_5)$ mit dem Orthogonalvektor a=(1,1,1,-2,-3) Hyperebenen mit b Werten von -3 (a´=(-1,-1,-1,-2,-3)) bis +5 (a´=(1,1,1,2,3)) konstruiert werden.

Folgendes Lemma gibt nun eine Schranke für den Betrag der Summe aller positiven (bzw. negativen) Elemente eines Orthogonalvektors an:

Lemma 10: Für $H \in HE(Q_d)$, $(0,..,0) \in H$ mit Orthogonalvektor $a=(a_1,..,a_d)$ gilt:

$\sum_{a_i \geq 0} a_i$ sowie $\sum_{a_i \leq 0} |a_i| \leq$ maximaler Betrag des Koordinatenwertes für Orthogonalvektoren aus Q_{d+1}

Beweis: O.B.d.A. genügt es, den Satz nur für die Summe über alle positiven Elemente zu zeigen; für den negativen Fall betrachte man -a.

Weiters seien o.B.d.A. $a_1,..,a_k$ die positiven Elemente des Orthogonalvektors a. Man setze nun $a_{d+1} := -\sum_{i=1}^{k} a_i$ womit durch $a' = (a_1,..,a_d,a_{d+1})$ eine Hyperebene H' aus Q_{d+1} durch den Nullpunkt erzeugt wird. Dies ist leicht zu sehen, da alle Ecken aus H natürlich auch auf H´ liegen (zusätzliche Koordinate auf Null gesetzt) und so einen d-1 dimensionalen Unterraum bilden. Weiters liegt noch die Ecke $x=(x_1,..,x_{d+1})$ mit $x_1=1,..,x_k=1$, $x_{k+1}=0,..,x_d=0$, $x_{d+1}=1$ ebenfalls auf H' und spannt einen zu den restlichen Ecken orthogonalen, eindimensionalen Raum auf. Alle Ecken zusammen bilden daher eine d-dimensionale Hyperebene des Q_{d+1}. Damit wurde gezeigt, daß $a_{d+1} := -\sum_{i=1}^{k} a_i$ ein gültiger Koordinatenwert eines Orthogonalvektors aus Q_{d+1} ist, womit unmittelbar die Aussage des Lemmas folgt. ❑

Zusammen mit den vorhergehenden Betrachtungen ergibt sich nun, daß der Betrag des Skalarwerts b einer Hyperebene H maximal den Betrag des Koordinatenwertes für Orthogonalvektoren aus Q_{d+1} erreichen kann. Daraus folgt mit Satz 5 von Seite 26 sofort folgendes

Lemma 11: Für $H \in HE(Q_d)$ mit Orthogonalvektor a und Skalarwert b gilt: $|b| \leq 2 * \left(\frac{d+1}{4}\right)^{(d+1)/2}$

Natürlich wird dieses Lemma durch eine schärfere Schranke für Satz 5 ebenfalls verbessert. Weiters können bekannte Werte für den maximalen Koordinatenwert in Q_{d+1} zur besseren Abschätzung von b für Dimension d verwendet werden. Soweit diese Grenzen berechnet wurden, ist dies in folgender Tabelle zusammengefaßt:

Dimension d	1	2	3	4	5	6	7	8	9
max. Koordinatenwert $\|a_i\|$	1	1	1	2	3	5	9	18	≥ 42
max. Skalarwert $\|b\|$	1	1	2	3	5	9	18	42	?

Tabelle 10: Maximaler Skalarwert b

Dabei fällt auf, daß bis Q_7 der maximale Wert von b tatsächlich immer den maximalen Koordinatenwert von a in Q_{d+1} erreicht. Für d=8 nimmt b Werte bis zu 42 an[21] und dies entspricht genau dem maximalen, bisher gefundenen Koordinatenwert eines Orthogonalvektors für Q_9. Diese Tatsachen legen die Vermutung nahe, daß der maximale Wert von b gleich dem maximalen Koordinatenwert der nächsthöheren Dimension ist.

Eine Folge dieser Vermutung wäre, daß die Berechnungen für Q_9 durch das Programm nur bis zu einem Koordinatenwert von 42 durchgeführt werden müßten. Damit würde die dafür notwendige Berechnungszeit auf unter drei Jahre sinken ...

4.3. Anzahl der Ecken auf Hyperebenen

Mit Lemma 6 wurde bereits gezeigt, daß für die Anzahl von Ecken p auf einer Hyperebene im Q_d die Schranken $d \le p \le 2^{d-1}$ gelten. Für schräge Hyperebenen ist die untere Schranke scharf, wogegen die Programmauswertungen eine wesentlich kleinere obere Schranke erwarten lassen.

Vermutung: Schräge Hyperebenen mit der maximalen Anzahl von Ecken aus Q_d besitzen einen Orthogonalvektor, der nur aus den Werten +1 und -1 besteht.

Falls diese Vermutung richtig ist (bis Q_{10} konnte sie bestätigt werden) lassen sich folgende Ergebnisse herleiten:

Lemma 12: Sei $H \in HE(Q_d)$ schräg. Dann ist die maximale Anzahl von Eckpunkten aus Q_d, die auf H liegen, genau $\binom{d}{\lfloor d/2 \rfloor}$ und es gibt $\begin{cases} 2^d & \text{für d ungerade} \\ 2^{d-1} & \text{für d gerade} \end{cases}$ solche Hyperebenen H.

Beweis: Sei a der Orthogonalvektor, der aus n mal +1 und d-n mal -1 bestehe (Vermutung!). O.B.d.A. sei $n \le \frac{d}{2}$. Für eine Ecke $x \in Q_d$ dürfen dann i = 0..n Stellen eins sein, an denen a = +1 ist und ebenso viele Stellen müssen in x eins sein, an denen a = -1 ist. Dann gilt aber für die Anzahl von Ecken, die auf H liegen mit $\sum_{i=0}^{\min(r,s)} \binom{n}{i}\binom{d-n}{r-i} = \binom{d}{r}$ [KNU1], S58):

$$\#\text{Ecken}(H) = \sum_{i=0}^{n} \binom{n}{i}\binom{d-n}{i} = \binom{d}{n}$$

Bekanntlich nimmt der Binomialkoeffizient dabei für $n = \left\lfloor \frac{d}{2} \right\rfloor$ sein Maximum an. Besitzt a also $\left\lfloor \frac{d}{2} \right\rfloor$ mal +1, so enthält die Hyperebene $\binom{d}{\lfloor d/2 \rfloor}$ Ecken aus Q_d. Für den Orthogonalvektor selbst

---

[21] Beispiel: a=(-9,-16,2,3,4,10,11,12) ist ein Orthogonalvektor aus Q_8 und 2+3+4+10+11+12 = 42.

gibt es ebenfalls $\binom{d}{d/2}$ Möglichkeiten, die $\binom{d}{d/2}$ Stellen für die +1en auszuwählen und dies ist daher die Anzahl von schrägen Hyperebenen im Nullpunkt, welche die maximale Eckenzahl besitzen. Multipliziert man dies mit der Anzahl aller Ecken (jede Ecke ist gleichberechtigt zur Nullecke) und dividiert durch die Anzahl Ecken pro Hyperebene (ansonsten würde ja jede Hyperebene für alle auf ihr liegenden Punkte einmal gezählt werden), so erhält man die Gesamtanzahl solcher Hyperebenen. Dabei ist noch zu berücksichtigen, daß für gerades d die zu sich selbst negativen Vektoren doppelt gezählt werden. Also muß dabei noch durch 2 dividiert werden. Es ergibt sich hiermit:

$$\#H = \binom{d}{d/2} \frac{2^d}{d} * \begin{cases} 1 & \text{falls d gerade} \\ \frac{1}{2} & \end{cases} = \begin{cases} 2^d & \text{für d ungerade} \\ 2^{d-1} & \text{für d gerade} \end{cases}$$

❑

Dimension d	1	2	3	4	5	6	7	8	9	10
max. Anzahl von Ecken einer schrägen HE	1	2	3	6	10	20	35	70	126	252
Anzahl von HE mit maximaler Eckanzahl	2	2	8	8	32	32	128	128	512	512

Tabelle 11: Maximale Anzahl von Ecken einer schrägen Hyperebene aus Q_d und Anzahl dieser Hyperebenen.

K-parallele Hyperebenen aus Q_d enthalten nach Lemma 7 genau $p*2^k$ Ecken, wobei p die Anzahl von Ecken auf einer schrägen Hyperebene aus Q_{d-k} ist. Daraus folgt:

d-1 parallel: $H \in HE(Q_{d-(d-1)}) \equiv H \in HE(Q_1)$: p = 1 Ecke $\Rightarrow$ d-1 parallele HE haben $1*2^{d-1} = 2^{d-1}$ Ecken

d-2 parallel: $H \in HE(Q_2)$: p = 2 Ecken $\Rightarrow$ d-2 parallele HE haben $2*2^{d-2} = 2^{d-1}$ Ecken

d-3 parallel: $H \in HE(Q_3)$: p = 3 Ecken $\Rightarrow$ d-3 parallele HE haben $3*2^{d-3}$ Ecken

d-4 parallel: $H \in HE(Q_4)$: p = 4 oder 6 Ecken $\Rightarrow$ d-4 parallele HE haben $4*2^{d-4} = 2^{d-2}$ Ecken oder $6*2^{d-4} = 3*2^{d-3}$ Ecken

Setzt man diese Liste fort, so bemerkt man, daß die Anzahl von Ecken mit abnehmender Parallelität sinkt. Diese Beobachtung wird auch durch folgendes Lemma gestützt:

Lemma 13 Die Anzahl von Hyperebenen $H \in HE(Q_d)$, $d \geq 2$, mit 2^{d-1} Ecken aus Q_d ist genau $d*(d+1)$, und diese Hyperebenen sind d-1 parallel oder d-2 parallel.

Beweis: Man zeigt zuerst, daß nur d-1 parallele und d-2 parallele Hyperebenen 2^{d-1} Ecken beinhalten:

Da die Anzahl von Ecken auf schrägen Hyperebenen für $d \geq 3$ mit $\left\lfloor \frac{d}{2} \right\rfloor < 2^{d-1}$ beschränkt ist (falls obige Vermutung stimmt), gilt für d-k parallele Hyperebenen mit $k \geq 3$ daß sie echt weniger als $2^{d-k} * 2^{d-(d-k)-1} = 2^{d-1}$ Ecken haben. Also können nur d-1 parallele und d-2 parallele Hyperebenen 2^{d-1} Ecken besitzen.

Es gilt aber $e_{d-1}(d) = \binom{d}{d-1} * e_0(1) = 2d$ und $e_{d-2}(d) = \binom{d}{d-2} * e_0(2) = d*(d-1)$ und damit $e_{d-1}(d)+e_{d-2}(d) = 2d + d*(d-1) = d*(d+1)$. ❑

Für die minimale Anzahl von Ecken auf nicht schrägen Hyperebenen gilt:

Lemma 14 Sei $H \in HE(Q_d)$ eine nicht schräge Hyperebene. Dann ist die minimale Anzahl von Ecken auf H 2(d-1) und solche Hyperebenen sind für $d \neq 3$ genau 1-parallel.

Beweis: Schräge Hyperebenen aus Q_d haben zumindest d Ecken, also ist die Anzahl von Ecken auf k-parallelen Hyperebenen mit $\geq 2^k*(d-k)$ nach unten begrenzt. Die Ableitung nach k: $\left(2^k*(d-k)\right)' = 2^k(d-k*\ln 2-1)$ ist für $k < d - \frac{1}{\ln(2)} = d - 1{,}44$ größer als Null und daher wird entweder für k=d-1 oder für k=1 das Minimum erreicht. Für d>3 gilt $2(d-1) < 2^{d-1}$, also wird für k=1 das Minimum von 2(d-1) Ecken erreicht. Für d=2 ist ohnehin d-1=1, und für die einzige Ausnahme d=3 gilt $2^{d-1}=2(d-1)$, also besitzen hier 1-parallele und 2-parallele Hyperebenen jeweils das Minimum von 4 Ecken. ❑

Daraus folgt auch, daß für $d \geq 3$ nur schräge Hyperebenen genau d Ecken besitzen können. Lediglich für d = 2 sind vier der sechs möglichen Hyperebenen (hier gewöhnliche Geraden) mit zwei Ecken aus Q_2 1-parallel.

4.4. Symmetrische Hyperebenen

Vergleicht man die Anzahl von schrägen Hyperebenen des Typs (j,p,k) aus Q_d mit der Anzahl von schrägen, symmetrischen (siehe Definition 8) Hyperebenen des Typs (j+k,2*p,j+k) aus Q_{d+1}, so fällt folgende Übereinstimmung auf: Sind im Q_d keine symmetrischen Hyperebenen vom Typ (i,p,i) vorhanden, so sind die Werte identisch, ansonsten differieren sie genau um die Anzahl dieser symmetrischen Hyperebenen. Daraus läßt sich ein direkter Zusammenhang von schrägen, unsymmetrischen Hyperebenen im Q_d mit schrägen, symmetrischen Hyperebenen im Q_{d+1} vermuten. Im folgenden soll nun der Zusammenhang hergeleitet werden.

Definition 15: Sei $x = (x_1,x_2,..,x_d)$ eine Ecke des Q_d. Dann wird mit $\overline{x} = (1-x_1,1-x_2,..,1-x_d)$ die dazu *komplementäre Ecke* bezeichnet.

Lemma 15: Sei $H \in HE(Q_d)$ und (a,b) der zugehörige Orthogonalvektor und Skalarwert. Dann gilt für alle Ecken x aus Q_d: $\langle a,x\rangle + \langle a,\overline{x}\rangle = \sum_{i=1}^{d} a_i$.

Beweis: $\langle a,x\rangle + \langle a,\overline{x}\rangle = \sum_{x_i=1} a_i + \sum_{x_i=0} a_i = \sum_{i=1}^{d} a_i$ ❑

Lemma 16: Sei $H \in HE(Q_d)$ eine symmetrische Hyperebene vom Typ (i,p,i). Dann ist p gerade.

Beweis: Da $i + p + i = 2^d \Rightarrow p = 2^d - 2*i = 2*(2^{d-1}-i)$, also p gerade. ❑

Lemma 17: Sei $H \in HE(Q_d)$ und (a,b) der zugehörige Orthogonalvektor und Skalarwert. Dann sind folgende Aussagen äquivalent:

i. H ist symmetrisch
ii. $\sum_{i=1}^{d} a_i = 2*b$
iii. $\forall$ Ecken $x \in Q_d$ gilt: $x \in H \Leftrightarrow \overline{x} \in H$
x liegt unter $H \Leftrightarrow \overline{x}$ liegt über H

Beweis: i. $\Rightarrow$ ii.: Sei H symmetrisch. Dann wird ii. durch Widerspruch bewiesen:

Fall 1: Angenommen $\sum_{i=1}^{d} a_i > 2*b$. Aus Lemma 15 folgt $\langle a,\overline{x}\rangle = \sum_{i=1}^{d} a_i - \langle a,x\rangle$ und damit:

$\forall x \in H$: $\langle a,\overline{x}\rangle = \sum_{i=1}^{d} a_i - b > 2*b - b = b$, d.h. $\overline{x}$ liegt über H.

$\forall$ x unter H: $\langle a,x\rangle < b \Rightarrow \langle a,\overline{x}\rangle = \sum_{i=1}^{d} a_i - \langle a,x\rangle > 2*b - b = b$, d.h. $\overline{x}$ liegt über H.

Da also für jede Ecke auf und unter H jeweils die komplementäre Ecke über H liegt, muß H vom Typ (j,p,≥p+j) sein. Da aber $p \neq 0$ kann H nicht symmetrisch sein, was im Widerspruch zur Annahme steht.

Fall 2: Angenommen $\sum_{i=1}^{d} a_i < 2*b$. Analog zu Fall 1 gilt:

$\forall x \in H$: $\langle a,\overline{x}\rangle = \sum_{i=1}^{d} a_i - b < 2*b - b = b$, d.h. $\overline{x}$ liegt unter H.

$\forall$ x über H: $\langle a,x\rangle > b \Rightarrow \langle a,\overline{x}\rangle = \sum_{i=1}^{d} a_i - \langle a,x\rangle < 2*b - b = b$, d.h. $\overline{x}$ liegt unter H.

womit auch dieser Fall einen Widerspruch zur Symmetrie von H darstellt und daher $\sum_{i=1}^{d} a_i = 2*b$ gelten muß.

ii. $\Rightarrow$ iii.: Sei $\sum_{i=1}^{d} a_i = 2*b$. Dann gilt mit Lemma 15 $\langle a,\overline{x}\rangle = \sum_{i=1}^{d} a_i - \langle a,x\rangle = 2*b - \langle a,x\rangle$ und

damit: $x \in H \Leftrightarrow \langle a,x\rangle = b \Leftrightarrow \langle a,\overline{x}\rangle = 2*b - b = b \Leftrightarrow \overline{x} \in H$

x liegt unter $H \Leftrightarrow \langle a,x\rangle < b \Leftrightarrow \langle a,\overline{x}\rangle > 2*b - b = b \Leftrightarrow \overline{x}$ liegt über H.

iii. $\Rightarrow$ i.: Da nach iii. für jede Ecke oberhalb von H jeweils genau die komplementäre Ecke unter H liegt und die auf H liegenden Ecken bezüglich ihres Komplements abgeschlossen sind, ist die Anzahl oberhalb und unterhalb liegender Ecken gleich, und H damit symmetrisch. ❑

Mit diesem Lemma läßt sich nun der eingangs erwähnte Zusammenhang von schrägen, unsymmetrischen Hyperebenen im Q_d mit schrägen, symmetrischen Hyperebenen im Q_{d+1} zeigen.

SATZ 7: Sei $M_p = \{H \mid H \in HE(Q_d)$ schräge, unsymmetrische Hyperebene vom Typ (j,p,k)$\}$ und $M'_p = \{H \mid H \in HE(Q_d)$ schräge, symmetrische Hyperebene vom Typ (i,p,i)$\}$. Dann gilt:

i. $|\{H \mid H \in HE(Q_{d+1})$ schräg, symmetrisch vom Typ (j+k,2*p,j+k)$\}| = |M_p|$, d.h. die Anzahl schräger, unsymmetrischer Hyperebenen mit p Ecken im Q_d ist gleich der Anzahl von schrägen, symmetrischen Hyperebenen mit 2*p Ecken im Q_{d+1}.

ii. $|\{H \mid H \in HE(Q_{d+1})$ 1-parallel, unsymmetrisch vom Typ (2*j,2*p,2*k)$\}| = (d+1) * |M_p|$.

iii. $|\{H \mid H \in HE(Q_{d+1})$ 1-parallel, symmetrisch vom Typ (2*i,2*p,2*i)$\}| = (d+1) * |M'_p|$.

Beweis: i.) Sei $H \in HE(Q_d)$ eine schräge, unsymmetrische Hyperebene vom Typ (j,p,k) und (a,b) der zugehörige Orthogonalvektor und Skalarwert. Mit Lemma 17 folgt:

$$H \text{ unsymmetrisch} \Leftrightarrow \sum_{i=1}^{d} a_i =: s \neq 2*b.$$

Setzt man $a'_{d+1} := 2*b-s$ und $a'_i = a_i$ für i=1,..,d. Sei $L \in HE(Q_{d+1})$ die durch (a',b) beschriebene Hyperebene. Da $a_i \neq 0$, i=1,..,d und $s \neq 2*b$ folgt $a'_i \neq 0$, i=1,..,d+1 und daher ist L nach Lemma 3 schräg. Da $\sum_{i=1}^{d+1} a'_i = \sum_{i=1}^{d} a_i + (2*b - s) = s + 2*b - s = 2*b$ ist L nach Lemma 17 symmetrisch.

Wird für eine Ecke x aus Q_d mit (x,0) die um eine Dimension erweiterte Ecke von Q_{d+1} bezeichnet (analog (x,1)), so gilt:

$x \in H \Leftrightarrow$ $(x,0) \in L$ (wegen $\langle (x,0),a'\rangle = \langle x,a\rangle + 0*a'_{d+1} = b + 0 = b$)

$(\overline{x},1) \in L$ ($\langle (\overline{x},1),a'\rangle = \langle \overline{x},a\rangle + a'_{d+1} = s - \langle x,a\rangle + 2*b - s = -b + 2*b = b$)

Da H d-1 dimensional ist und durch die Punkte (x,0) und $(\overline{x},1)$ für L zusätzlich ein zu H orthogonaler, eindimensionaler Raum aufgespannt wird, ist L tatsächlich d dimensional und damit eine Hyperebene von Q_{d+1}.

x über H $\Leftrightarrow$ (x,0) über L ($\langle (x,0),a'\rangle = \langle x,a\rangle + 0*a'_{d+1} > b + 0 = b$)

$(\overline{x},1)$ unter L ($\langle (\overline{x},1),a'\rangle = \langle \overline{x},a\rangle + a'_{d+1} = s - \langle x,a\rangle + 2*b - s < b$)

x unter H $\Leftrightarrow$ (x,0) unter L ($\langle (x,0),a'\rangle = \langle x,a\rangle + 0*a'_{d+1} < b + 0 = b$)

$(\overline{x},1)$ über L ($\langle (\overline{x},1),a'\rangle = \langle \overline{x},a\rangle + a'_{d+1} = s - \langle x,a\rangle + 2*b - s > b$)

Damit folgt aber, daß, wenn H schräg, unsymmetrisch vom Typ (j,p,k) ist, L schräg, symmetrisch vom Typ (j+k,2∗p,j+k) sein muß. Da H d-1 dimensional ist und durch die Punkte (x,0) und ($\overline{x}$,1) für L zusätzlich ein zu H orthogonaler, eindimensionaler Raum aufgespannt wird, ist L tatsächlich d dimensional und damit eine Hyperebene von Q_{d+1}.

Zu beachten ist dabei, daß der Orthogonalvektor für L aus verschiedenen Hyperebenen H gewonnen werden kann, da natürlich zwei nur in einem Wert unterschiedliche Orthogonalvektoren a_1 und a_2 von H_1 und H_2 durch obige Ergänzung denselben Orthogonalvektor a' für L liefern können.

Da für eine symmetrische Hyperebene $L \in HE(Q_{d+1})$ die Anzahl der auf ihr liegenden Ecken immer gerade ist (Lemma 16), kann umgekehrt aus L immer auch ein Orthogonalvektor für eine schräge, unsymmetrische Hyperebene $H \in Q_d$ konstruiert werden. Da es dabei egal ist, welche der d+1 Koordinaten von a' weggelassen wird, können auch hier mehrere Hyperebenen H aus L gewonnen werden. Betrachtet man nun für beide Richtungen die Anzahl möglicher Orthogonalvektoren für die einander entsprechenden Hyperebenen H und L, so entspricht diese nach Lemma 4 genau dem Multinomialkoeffizienten. Sind also $n_1,..,n_k$, ($n_1 + .. + n_k = d+1$) die Häufigkeit des Auftretens der Koeffizienten von a', so gilt:

Anzahl möglicher Orthogonalvektoren für $L = \frac{(d+1)!}{n_1! * n_2! * ... * n_k!}$ und

Anzahl möglicher Orthogonalvektoren für $H = \frac{d!}{(n_1-1)! * n_2! * ... * n_k!} + \frac{d!}{n_1! * (n_2-1)! * ... * n_k!} + ... + \frac{d!}{n_1! * n_2! * ... * (n_k-1)!}$

Es gilt aber:

$$\frac{d!}{(n_1-1)! * n_2! * ... * n_k!} + \frac{d!}{n_1! * (n_2-1)! * ... * n_k!} + ... + \frac{d!}{n_1! * n_2! * ... * (n_k-1)!} =$$

$$d!\left(\frac{n_1}{n_1! * n_2! * ... * n_k!} + \frac{n_2}{n_1! * n_2! * ... * n_k!} + ... + \frac{n_k}{n_1! * n_2! * ... * n_k!}\right) =$$

$$d!\left(\frac{n_1 + n_2 + ... + n_k}{n_1! * n_2! * ... * n_k!}\right) = d!\left(\frac{d+1}{n_1! * n_2! * ... * n_k!}\right) = \frac{(d+1)!}{n_1! * n_2! * ... * n_k!}$$

D.h. die Anzahl schräger, unsymmetrischer Hyperebenen vom Typ (j,p,k) im Q_d ist gleich der Anzahl von schrägen, symmetrischen Hyperebenen vom Typ (j+k,2∗p,j+k) im Q_{d+1} womit i. bewiesen ist.

ii.,iii.: Sei $H \in HE(Q_d)$ eine schräge Hyperebene vom Typ (j,p,k) und (a,b) der zugehörige Orthogonalvektor und Skalarwert. Ähnlich der vorhergehenden Konstruktion wird nun $a'_{d+1} = 0$ gesetzt, womit (a,b) eine 1-parallele Hyperebene L beschreibt (da insbesondere die parallele Richtung orthogonal zu H steht, bildet L tatsächlich eine d-dimensionale Hyperebene). Da sich nun (x,0) und (x,1) bezüglich L immer gleich verhalten wie x bezüglich H, ist L vom Typ (2∗j,2∗p,2∗k). Damit bleibt auch die Eigenschaft der Symmetrie unverändert. Da eine der d+1

Koordinaten als parallel gewählt werden kann ($a'_i = 0$; $a'_{d+1} = a_i$) entsprechen jedem Paar (a,b) genau (d+1) Paare (a',b) womit ii. und iii. bewiesen wären. ❑

Bemerkung: Die Konstruktion des Beweises impliziert auch, daß für eine Hyperebene der Term $2*b-\sum_{i=1}^{d} a_i$ derselben Schranke wie die Elemente des Orthogonalvektors unterliegt. Insbesonders für b = 0 bedeutet dies, daß der Betrag der Summe der Komponenten des Orthogonalvektors nicht größer ist, als dies ein einzelnes Element sein kann.

Die Punkte ii und iii des Satzes entsprechen der Aussage von Satz 1. Neu ist lediglich, daß der Zusammenhang für symmetrische und unsymmetrische Hyperebenen auch getrennt gültig ist. Wesentlich wichtiger ist jedoch der erste Punkt, denn damit kann aus der Kenntnis der schrägen, symmetrischen Hyperebenen des Einheitshyperwürfels auf alle schrägen Hyperebenen und damit mit Satz 1 auf alle möglichen Hyperebenen geschlossen werden. Es gilt der Zusammenhang: Anzahl schräger Hyperebenen im Q_d ist gleich der Anzahl schräger, symmetrischer Hyperebenen im Q_d plus der Anzahl schräger, symmetrischer Hyperebenen im Q_{d+1}. Dies ist für die Theorie vorteilhaft, da man sich nun auf die Untersuchung von symmetrischen Hyperebenen beschränken kann. Für die explizite enumerative Berechnung der Anzahl von Hyperebenen liefert dieser Satz jedoch keine Ersparnis, da die Berechnung von schrägen, symmetrischen Hyperebenen im Q_{d+1} zumindest ebenso aufwendig ist, wie die direkte Berechnung aller schrägen Hyperebenen im Q_d.

Der erste Teil des Beweises zu Satz 7 zeigt, daß nicht nur die Elemente des Orthogonalvektors beschränkt sind, sondern daß auch der Betrag der Summe dieser Elemente mit derselben Schranke begrenzt ist, da die Summe ja selbst wiederum Element eines Orthogonalvektors sein kann (einfach zu sehen für b=0).

Aus dem Satz folgt auch, daß symmetrische Hyperebenen aus Q_d stets vom Typ $(2^{d-1}-m, 2*m, 2^{d-1}-m)$ mit $d-1 \le m \le 2^{d-2}$ sind[22]. Weiters folgt, daß, wenn es keine schrägen, unsymmetrischen Hyperebenen mit p Ecken im Q_d gibt, es auch keine schrägen, symmetrischen Hyperebenen mit 2*p Ecken im Q_{d+1} gibt. Dieser und ähnliche Zusammenhänge werden in folgender Tabelle dargestellt:

$H \in HE(Q_d)$, schräg, p Ecken			$H \in HE(Q_{d+1})$, schräg, 2*p Ecken	
symmetrisch	unsymmetrisch		symmetrisch	unsymmetrisch
Nein	Nein	⇒	Nein	Ja+Nein
Nein	Ja	⇒	Ja	Ja+Nein
Ja	Nein	⇒	Nein	Nein (? s.Text)
Ja	Ja	⇒	Ja	Ja+Nein

Tabelle 12: Zusammenhang zwischen schrägen Hyperebenen in Q_d und Q_{d+1}

[22]Diese Grenzen hängen direkt mit den Grenzen für die Anzahl von Ecken auf schrägen Hyperebenen im Q_{d-1} zusammen. Daher ist die untere Grenze scharf. Für eine Diskussion der oberen Grenze siehe Kapitel 4.3.

Die Spalte für die symmetrischen Hyperebenen ergibt sich direkt aus Satz 7. Für die Ja+Nein Fälle der unsymmetrischen Hyperebenen aus Q_{d+1} dienen Beispiele aus Q_5 bis Q_8:

- Zeile 1: In Q_5 gibt es keine schrägen Hyperebenen mit 6 bzw. 9 Ecken[23]. In Q_6 gibt es schräge, unsymmetrische Hyperebenen mit 12 Ecken, jedoch keine mit 18 Ecken.
- Zeile 2: In Q_5 gibt es nur unsymmetrische, schräge Hyperebenen mit 5,7 bzw. 10 Ecken. In Q_6 gibt es schräge, unsymmetrische Hyperebenen mit 10 Ecken, jedoch keine mit 14 oder 20 Ecken.
- Zeile 3: Für den Fall, daß es im Q_d zwar symmetrische aber keine unsymmetrischen Hyperebenen mit p Ecken gibt, konnten bis zum Paar $Q_8 \rightarrow Q_9$ für Q_{d+1} keine unsymmetrischen Hyperebenen festgestellt werden. Möglicherweise tritt hier erst in höheren Dimensionen ein solcher Fall auf, oder es kann ein allgemein gültiger Satz gezeigt werden.[24]
- Zeile 4: In Q_6 gibt es symmetrische und unsymmetrische, schräge Hyperebenen mit 10 Ecken, und im Q_7 gibt es schräge, unsymmetrische Hyperebenen mit 20 Ecken. In Q_7 gibt es symmetrische und unsymmetrische, schräge Hyperebenen mit 12, 14, 16, 18, 20 und 22 Ecken. In Q_8 gibt es schräge, unsymmetrische Hyperebenen mit 24, 28, 32, 36 und 40 Ecken jedoch keine mit 44 Ecken.

4.5. Zueinander parallele Hyperebenen

Dieser Abschnitt behandelt zueinander parallele Hyperebenen aus Q_d. In Q_2 und Q_3 ist leicht zu sehen, daß maximal zwei Hyperebenen zueinander parallel sein können; - in höheren Dimensionen ist dies weniger anschaulich:

Betrachtet man Hyperebenen und deren Darstellung (a,b) als Orthogonalvektor und Skalarwert, so ist die Parallelität zueinander darauf reduzierbar, daß sie denselben Orthogonalvektor a, aber verschiedene Skalarwerte b besitzen. Damit ist nun ein abstrakter Zugang möglich, und der folgende Satz liefert Schranken für die maximale Anzahl von zueinander parallelen Hyperebenen.

---

[23]Um bei diesen Beispielen keine trivialen Fälle zu betrachten werden immer nur solche Eckzahlen gewählt, sodaß es im Q_d stets noch Hyperebenen mit weniger bzw. mehr Ecken gibt.

[24]Mit großer Wahrscheinlichkeit tritt von Q_9 auf Q_{10} ein Gegenbeispiel auf: Im Q_9 gibt es zumindest 129024 schräge Hyperebenen (berechnet mit einem maximalem Koordinatenwert des Orthogonalvektors von 8) mit 70 Ecken. Im Q_8 gibt es ebenfalls genau 129024 schräge, unsymmetrische Hyperebenen mit 35 Ecken. Falls es also im Q_9 nicht doch mehr Hyperebenen mit 70 Ecken gibt, sind diese alle symmetrisch (Satz 7). Nun gibt es aber im Q_{10} schräge Hyperebenen mit 140 Ecken (zumindest 337920), die dann unsymmetrisch sein müssen. Damit wäre ein Gegenbeispiel gefunden.

Satz 8: Sei Kmax der maximal mögliche Betrag der Koordinatenwerte eines Orthogonalvektors im Q_d. Dann gilt für die Anzahl P_d von maximal zueinander parallelen Hyperebenen $H \in HE(Q_d)$:

$$d\text{-}1 \le P_d \le Kmax+1 \le 2*\left\lfloor \frac{d}{2} \right\rfloor + 1$$

Beweis: Untere Schranke: Durch vollständige Induktion über die Dimension d wird zuerst gezeigt, daß es im Q_d immer Hyperebenen durch den Nullpunkt mit den Orthogonalvektor-Elementen ±1 gibt. Dabei existieren Orthogonalvektoren mit jeweils genau 1 bis d-1 negativen Komponenten:

Als Induktionsanfang sind dies für Q_2 a = (1,-1) und für Q_3 a = (1,1,-1) sowie a = (1,-1,-1), wie leicht zu sehen ist.

Besitzt nun Q_d Orthogonalvektoren (1,..,1,-1) bis (1,-1,..,-1), so gilt dies auch für Q_{d+1}: Wird ein Vektor a für $H \in HE(Q_d)$ um +1 zu (a,1) erweitert, so liegen für alle Ecken $x \in H$ die erweiterten Ecken (x,0) auch auf $H' \in HE(Q_{d+1})$ und spannen damit einen d-1 dimensionalen Raum auf. Da a zumindest eine negative Komponente besitzt, liegt zusätzlich noch eine Ecke der Form (0,..,0,1,0,..,0,1) mit je einer Eins an der Stelle der negativen Komponente sowie der zusätzlichen Dimension, ebenfalls auf H. Da diese Ecke außerhalb des d-1 dimensionalen Raumes liegt, ergibt sich gesamt eine d dimensionale Hyperebene für Q_{d+1}. Da diese Erweiterung für alle ±1 Kombinationen möglich ist, sofern zumindest eine Koordinate positiv bzw. negativ ist, folgt daraus, daß auch im Q_{d+1} alle gewünschten Orthogonalvektoren mit 1 bis d negativen Komponenten existieren.

Analog zur Konstruktion von Kapitel 4.2.2. können nun aus diesen Orthogonalvektoren Hyperebenen mit dem Orthogonalvektor a'=(1,..,1) und b-Werten von 1 bis d-1 konstruiert werden, womit zumindest d-1 zueinander parallele Hyperebenen gefunden sind.

Obere Schranke: Für einen beliebigen, aber festen Orthogonalvektor a aus Q_d folgt aus der Bemerkung zum Beweis von Satz 7: $|2*b-\sum_{i=1}^{d} a_i| \le Kmax$. Mit $s:=\sum_{i=1}^{d} a_i$ gilt dann:

$$|2*b-s| \le Kmax \Leftrightarrow -Kmax \le 2*b-s \le Kmax \Leftrightarrow s-Kmax \le 2*b \le s+Kmax$$

Damit kann der Term 2∗b jeden zweiten Wert aus einem Intervall der Größe 2∗Kmax+1 annehmen. Da b ganzzahlig ist, folgt damit, daß b maximal Kmax+1 verschiedene Werte annehmen kann. Somit sind höchstens Kmax+1 verschiedene Hyperebenen mit demselben Orthogonalvektor a möglich. Zusammen mit Satz 5 ist damit ist auch die obere Schranke gezeigt.

❑

Ähnliche Betrachtungen wie im ersten Teil des Beweises sind auch im Werk von Coxeter zu finden ([COX],§13). Interessant ist dabei, daß die Anzahl von Eckpunkten auf diesen Hyperebenen genau einer Zeile im Pascal'schen Dreieck entsprechen (ohne Einer!), also $\binom{d}{1}, \binom{d}{2} \ldots \binom{d}{d-1}$. Dies wird verständlich, wenn man den Orthogonalvektor a=(1,..,1) betrachtet und sich klarmacht, mit wieviele Binärzahlen das Skalarprodukt mit a gerade den gewünschten b-Wert ergibt.

Nebenstehende Tabelle gibt die tatsächliche maximale Anzahl von zueinander parallelen Hyperebenen im Q_d an. Erst in Q_8 weicht dabei der Wert von der oberen Schranke ab. Dies bedeutet aber, daß weder die untere noch die obere Schranke scharf ist. Auch finden sich in den Orthogonalvektoren mit einer maximalen Anzahl von parallelen Hyperebenen keine Regelmäßigkeiten, weshalb sich daraus kein konstruktiver Beweis für eine genaue Anzahl ableiten läßt.

Dimension d	Anzahl paralleler Hyperebenen	Beispiel für Orthogonalvektor	Wertebereich von b
2	2	(1,0)	0,1
3	2	(1,1,1)	1,2
4	3	(1,1,1,1)	1,2,3
5	4	(1,1,1,1,1)	1,2,3,4
6	6	(1,1,1,2,2,2)	2..7
7	10	(1,1,2,2,3,4,4)	4..13
8	17	(1,2,3,4,5,6,7,8)	10..26

Tabelle 13: Maximale Anzahl zueinander paralleler Hyperebenen im Q_d.

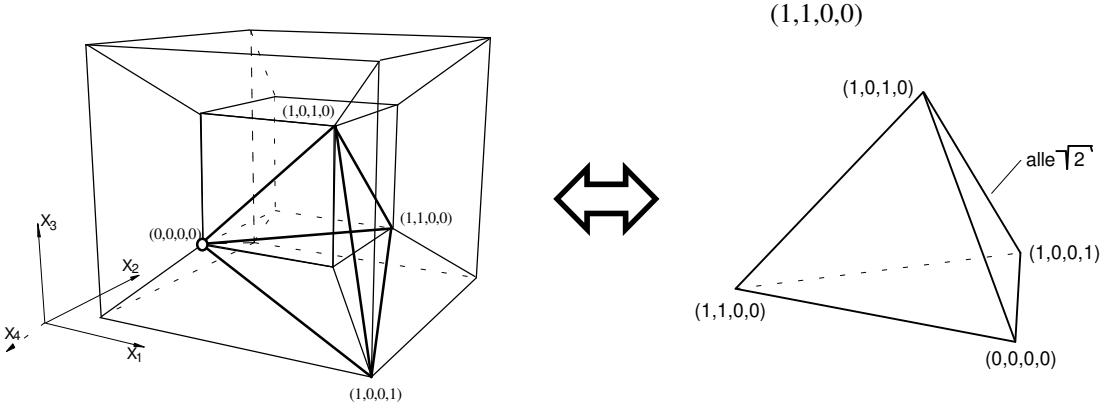
4.6. Schräge Hyperebenen und Spuren im Q_4

Bis zur Dimension 3 sind die Hyperebenen einfach vorstellbar und somit einer Untersuchung leicht zugänglich. Für Q_3 ist z.B. leicht überprüfbar, daß es tatsächlich nur eine Art schräger Hyperebenen gibt, die unsymmetrisch und die konvexe Hülle von Ecken aus Q_3 sind. Für Q_4 ist dies schon etwas schwieriger! Hier folgt daher eine vollständige Auflistung der möglichen Arten von Hyperebenen im Q_4.

Um auch für höhere Dimensionen verschiedene Hyperebenen oder deren Spur rasch vergleichen zu können, ist es praktisch, den Raumabstand der sie bildenden Ecken aus Q_d zu berechnen. Seien x_1 und x_2 Ecken von Q_d. Mit $hd(x_1,x_2)$ = Hamming-Distanz von x_1 und x_2 wird die Anzahl unterschiedlicher Koordinatenwerte (bei Betrachtung als Binärzahl: Anzahl unterschiedlicher Bits) bezeichnet. Aus dem verallgemeinerten Satz von Pythagoras folgt dann für den Raumabstand zweier Ecken sofort: $d(x_1,x_2) = \sqrt{hd(x_1,x_2)}$. Für schräge Hyperebenen ergibt sich damit zumindest ein Abstand von $\sqrt{2}$ für zwei darauf liegende Ecken. Der maximale Raumabstand zweier Ecken aus Q_d ist natürlich $\sqrt{d}$. Für symmetrische Hyperebenen ist zu jeder daraufliegenden Ecke genau die dazu komplementäre Ecke maximal (also $\sqrt{d}$) entfernt. Für unsymmetrische Hyperebenen gilt $\sqrt{d-1}$ als obere Schranke, da ja zueinander komplementäre Ecken nicht enthalten sein können.

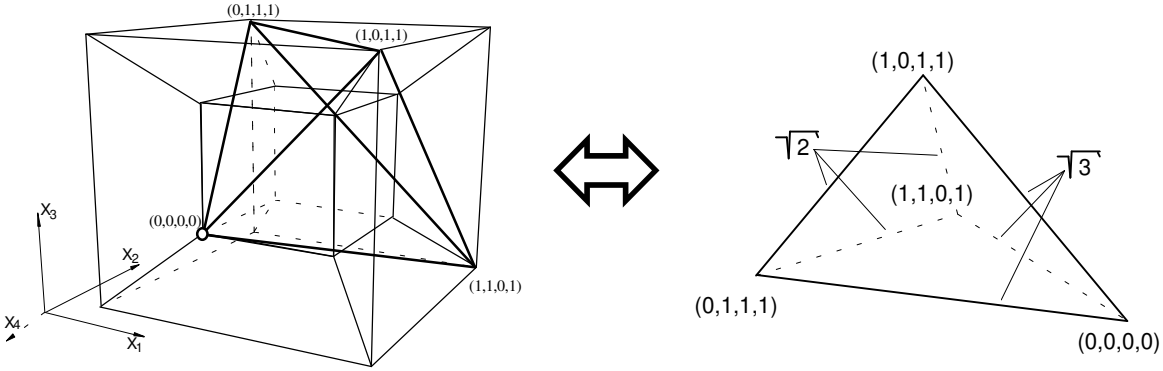
Insgesamt gibt es vier verschiedene Arten von Orthogonalvektoren für schräge Hyperebenen im Q_4, die durch den Nullpunkt gehen:

HE Typ (1,4,11) Orthogonalvektor: (-1,1,1,1) Ecken: (0,0,0,0) 4 mal im Nullpunkt
(1,0,0,1)
(1,0,1,0)
(1,1,0,0) 4∗(16/4) = 16 mal im Q_4



Die Spur dieser Hyperebene ist im Q_3 ein gleichseitiges Tetraeder mit der Seitenlänge $\sqrt{2}$. Der einzige Punkt unterhalb der Ebene ist (1,0,0,0), die vier benachbarten[25] Ecken (0,0,0,0), (1,1,0,0), (1,0,1,0) und (1,0,0,1) liegen auf der Ebene. Daher ist hier die Spur die Konvexe Hülle von Ecken aus Q_4. Diese Ebene kann auch als vierdimensionale Verallgemeinerung von verstanden werden. (Wie bereits gezeigt, existiert der Typ (1,d,2^d-d-1) für beliebige Dimensionen.)

HE Typ (4,4,8) Orthogonalvektor: (1,1,1,-2) Ecken: (0,0,0,0) 4 mal im Nullpunkt
(0,1,1,1)
(1,0,1,1)
(1,1,0,1) 4∗(16/4) = 16 mal im Q_4



Diese Hyperebene entspricht der bereits in Abbildung 9 dargestellten. Die Spur bildet einen Tetraeder mit verschiedenen Kantenlängen. Das gleichseitige Basisdreieck hat eine Kantenlänge von $\sqrt{2}$, wohingegen die Spitze (die Ecke (0,0,0,0)) von allen anderen Ecken um $\sqrt{3}$ entfernt ist. Wegen dieser "Unregelmäßigkeit" ist auch die Spur nicht die konvexe Hülle der Ecken! Die jeweils benachbarten Ecken (0,0,1,1) und (0,0,1,0),

[25]Unter benachbart versteht man Ecken, die durch eine Kante aus Q_d verbunden sind, also Ecken mit einer Hamming-Distanz von 1.

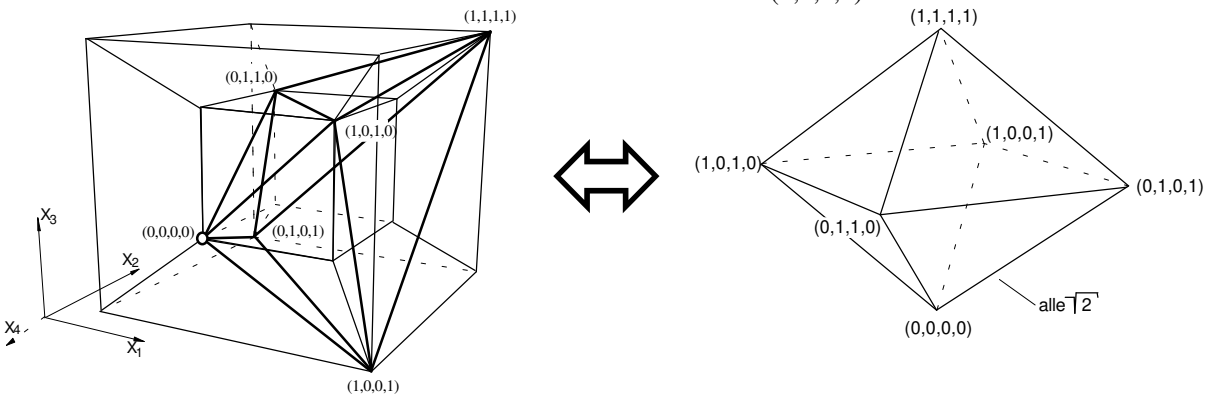
(0,1,0,1) und (0,1,0,0) sowie (1,0,0,1) und (1,0,0,0) liegt dabei unter bzw. über H, daher schneidet die Hyperebene H die Kanten zwischen diesen Punkten. Also erhält man mit (0,0,1,½), (0,1,0,½) und (1,0,0,½) zusätzliche Eckpunkte der Spur.

Da es im Q_3 keine Hyperebene mit einem Dreieck mit den Seiten $\sqrt{2}$,$\sqrt{3}$und $\sqrt{3}$ gibt, gibt es auch keine Möglichkeit diesen Typ der Hyperebene aus dem Q_3 abzuleiten. Analog gilt dies auch für höhere Dimensionen, d.h. es treten ständig neue Typen von Hyperebenen auf.

HE Typ (4,4,8) Orthogonalvektor: (1,1,-1,-2) Ecken: (0,0,0,0) 12 mal im Nullpunkt
(0,1,1,0)
(1,0,1,0) 12∗(16/4) = 48 mal im Q_4
(1,1,0,1)

Hier wird derselbe Typ durch einen unterschiedlichen Orthogonalvektor erzeugt. Dies liegt daran, daß das durch die Hyperebene gebildete Tetraeder nicht vollkommen symmetrisch ist. Je nachdem, ob nun der Nullpunkt die weiter entfernte Spitze ist (wie im vorhergehenden Fall), oder ob er eine Ecke des Basisdreiecks ist (wie hier), ist der Orthogonalvektor unterschiedlich. Daraus darf jedoch nicht gefolgert werden, daß derselbe Typus einer Hyperebene trotz unterschiedlichen Orthogonalvektoren immer dasselbe Polytop im Q_{d-1} erzeugt (vergleiche dazu den nächsten Abschnitt).

HE Typ (5,6,5) Orthogonalvektor: (1,1,-1,-1) Ecken: (0,0,0,0) 3 mal im Nullpunkt
(0,1,0,1)
(0,1,1,0) 12∗(16/6) = 8 mal im Q_4
(1,0,0,1)
(1,0,1,0)
(1,1,1,1)



Die Spur der Hyperebene vom Typ (5,6,5) im Q_4 ist ein regelmäßiger Oktaeder mit der Seitenlänge $\sqrt{2}$. Da es kein Paar benachbarter Ecken gibt, die unter und über H liegen, ist auch hier die Spur die konvexe Hülle von Ecken der Hyperebene. Weiters ist dies auch die einzige schräge und symmetrische Art von Hyperebenen im Q_4: Wie bei den symmetrischen Hyperebenen beschrieben[26], läßt sie sich aus dem Q_3, aus der dort einzigen Art von schrägen Hyperebenen, dem Typ (1,3,4), erzeugen. Dies sieht man auch, wenn man die beteiligten Ecken bezüglich der Achse X_4 in zwei Gruppen teilt. Dann bilden sowohl (0,0,0,0), (0,1,1,0), (1,0,1,0) als auch

[26]Vergleiche Satz 7.

(0,1,0,1), (1,0,0,1), (1,1,1,1) jeweils eine Hyperebene des Typs (1,3,4) im betreffenden Sub-Hyperwürfel. Wie aus dem zugehörigem Beweis zu erwarten, sind die Ecken der beiden Gruppen zueinander komplementär.

Symmetrische Hyperebenen müssen nicht immer so regelmäßig sein, wie es hier der Fall ist. Allerdings muß man bis Q_6 untersuchen, um eine vollkommen unregelmäßige Spur zu finden. Der Typ (12,8,12) im Q_5 entsteht zwar auch aus den unregelmäßigen Typen (4,4,8) aus Q_4, das heißt aber noch nicht, daß er selbst auch vollkommen unregelmäßig sein muß. Tatsächlich gibt es für diese Hyperebene nur eine Orthogonalvektor-Art für den Nullpunkt[27], weshalb alle seine Ecken "gleichberechtigt" sein müssen. "Ungleiche" Ecken des Typs (4,4,8) "ergänzen" sich hier einander also derart, daß die Spur im Q_5 tatsächlich regelmäßig wird: Jede Ecke der Spur ist von drei Ecken um $\sqrt{2}$, von weiteren drei um $\sqrt{3}$ und von genau einer (dem Komplement) genau $\sqrt{5}$ entfernt[28]. Erst im Q_6 treten für den symmetrischen Typ (27,10,27) erstmals drei verschiedene Orthogonalvektor-Arten auf, womit ein Beispiel einer unregelmäßige Spur einer symmetrischen Hyperebene gefunden ist.

Auch für höhere Dimensionen d gibt es stets Hyperebenen, deren Spur die konvexe Hülle von Ecken aus Q_d ist. Dies gilt z.B. immer für die Typen $(1,d,2^d\text{-}d\text{-}1)$ mit dem Orthogonalvektor (1,1,..,1). Da hier nur eine Ecke unterhalb (oder oberhalb) liegt, und die dazu benachbarten Ecken genau auf der Hyperebene liegen, wird keine Kante des Q_d geschnitten. Einen genauen Zusammenhang liefert folgender

Satz 9: Sei $H \in HE(Q_d)$ und a der zugehörige Orthogonalvektor. Die Spur $S = H \cap Q_d$ ist genau dann nicht die konvexe Hülle von Ecken aus Q_d , wenn eine Koordinate X_i existiert, für die $|a_i| \geq 2$ gilt.

Beweis: Für die Fälle $d \leq 3$ gibt es weder Orthogonalvektoren mit $|a_i| \geq 2$ noch entsprechende Spuren. Es genügt also, den Satz für Dimensionen $d \geq 4$ zu zeigen:

"$\Rightarrow$": Ist S nicht die konvexe Hülle, so muß zumindest eine Kante aus Q_d von H geschnitten werden, d.h. es gibt zwei benachbarte Ecken x_1 und x_2 die über bzw. unter H liegen. Also gibt es ein b mit $\langle a,x_1\rangle > b$ und $\langle a,x_2\rangle < b$. Da x_1, x_2, a und b alle ganzzahlig sind, gilt $\langle a,x_1\rangle \geq b+1$ und $\langle a,x_2\rangle \leq b-1$ und damit $\langle a,x_1\rangle \geq \langle a,x_2\rangle+2$. Da sich x_1 und x_2 aber nur an einer Position i unterscheiden (benachbart!), muß $|a_i| \geq 2$ gelten.

"$\Leftarrow$": Es genügt, nur jene Hyperebenen zu betrachten, die durch den Nullpunkt gehen. Alle anderen Hyperebenen sind dazu äquivalent (siehe dazu Seite 24). Für auf der Hyperebene liegende Ecken bedeutet dies, daß ihr Skalarprodukt mit dem Orthogonalvektor a zu 0 wird.
O.B.d.A. sei ein betragsmäßig größtes Element a_i von a positiv (sonst multipliziert man den gesamten Orthogonalvektor mit -1) und nach Voraussetzung gilt $a_i \geq 2$. Weiters gilt nach Definition $ggT(a_1,..,a_d) = 1$. $\Rightarrow \exists\, a_j$ mit $0 < |a_j| < a_i$.

---

[27] Dies läßt sich auch daran erkennen, daß der Orthogonalvektor (1,1,1,-1,-2) (für den Typ (12,8,12) im Q_5) beide Orthogonalvektoren (1,1,1,-2) und (1,1,-1,-2) (für den Typ (4,4,8) im Q_4) als "Teilvektor" enthält.

[28] Analog dazu ist ja auch im normalem Würfel (Q_3) jede Ecke von drei Ecken um 1 (Kante), von weiteren drei um $\sqrt{2}$ (Seitendiagonale) und von genau einer (dem Komplement) um $\sqrt{3}$ (Raumdiagonale) entfernt.

Fall 1: $a_j < 0$: $x := (0,..,0_i,..,1_j,..,0) \in Q_d \quad \Rightarrow \langle a,x \rangle = a_j < 0$, d.h. x liegt unter H

$y := (0,..,1_i,..,1_j,..,0) \in Q_d \quad \Rightarrow \langle a,x \rangle = a_i + a_j > 0$, d.h. y liegt über H

Da die Ecken x und y aber durch eine Kante aus Q_d verbunden sind (parallel zur i-Achse) schneidet die Hyperebene diese Kante. Daher kann die Spur nicht die konvexe Hülle von Ecken aus Q_d sein.

Fall 2: $a_j > 0$: $\Rightarrow \exists\, a_k < 0$ mit $-a_k \leq a_i$. (Falls es keine negativen Koordinatenwerte von a gäbe, könnte keine Ecke $\neq (0,..0)$ auf H liegen.)

Fall 2a: $|a_k| < a_i$: $x := (0,..,0_i,..,1_k,..,0) \in Q_d \quad \Rightarrow \langle a,x \rangle = a_k < 0$, d.h. x liegt unter H

$y := (0,..,1_i,..,1_k,..,0) \in Q_d \quad \Rightarrow \langle a,x \rangle = a_i + a_k > 0$, d.h. y liegt über H

Folgerung analog zu Fall 1.

Fall 2b: $|a_k| = a_i$: $x := (0,..,0_i,..,1_j,..,1_k,..,0) \in Q_d \Rightarrow \langle a,x \rangle = a_j + a_k < 0$, d.h. x liegt unter H

$y := (0,..,1_i,..,1_j,..,1_k,..,0) \in Q_d \Rightarrow \langle a,x \rangle = a_i + a_j + a_k = a_j > 0$, d.h. y liegt über H

Folgerung analog zu Fall 1.

Daher kann in allen Fällen mit $|a_i| \geq 2$ die Spur nicht die konvexe Hülle von Ecken aus Q_d sein.

❑

4.7. Nichteindeutigkeit der Typen

Von den 2704 schrägen Hyperebenen im Q_5 gibt es insgesamt acht verschiedene Typen. Zu diesen gehören insgesamt 17 verschiedene Orthogonalvektor-Arten. Die folgende Tabelle gibt dazu einen Überblick:

Anzahl Ecken auf der Hyperebene	Anzahl Hyperebenen	Typ	Anzahl Orthogonalvektoren
5	2112:		
	32	(1,5,26)	1
	320	(4,5,23)	2
	640	(8,5,19)	3
	1120	(11,5,16)	5
7	480:		
	160	(5,7,20)	2
	320	(9,7,16)	2
8	80:		
	80	(12,8,12)	1
10	32:		
	32	(6,10,16)	1
$e_0(5)=$	2704		17

Tabelle 14: Schräge Hyperebenen im Q_5.

Eine nähere Untersuchung der verschiedenen Typen liefert mehrere regelmäßige und unregelmäßige Spuren im Sub-Hyperkubus Q_4 analog den vorhergehenden Betrachtungen. Besonderes Interesse verdient aber der Typ (11,5,16). Hier tritt erstmals eine Mehrdeutigkeit auf, die im Q_1 bis Q_4 nicht beobachtet werden kann.

Folgende fünf Orthogonalvektor-Arten gibt es für Hyperebenen vom Typ (11,5,16):

Orthogonal-vektor (b=0)	Ecken aus Q_5	Kurzname für die Ecke	Hammingdistanz zu e1	e2	e3	e4	e5	Gruppe I (s. Text)	Gruppe II	Gruppe II
(-3,1,1,1,1)	(0,0,0,0,0)	e1	-	4	4	4	4	e2,e3, e4,e5	e1	
	(1,0,1,1,1)	e2	4	-	2	2	2			
	(1,1,0,1,1)	e3	4	2	-	2	2			
	(1,1,1,0,1)	e4	4	2	2	-	2			
	(1,1,1,1,0)	e5	4	2	2	2	-			
(-1,-3,1,1,1)	(0,0,0,0,0)	e2	4	-	2	2	2	e2,e3, e4,e5	e1	
	(0,1,1,1,1)	e1	-	4	4	4	4			
	(1,0,0,0,1)	e3	4	2	-	2	2			
	(1,0,0,1,0)	e4	4	2	2	-	2			
	(1,0,1,0,0)	e5	4	2	2	2	-			
(-1,-3,1,2,2)	(0,0,0,0,0)	e1	-	2	3	3	4	e1,e2	e3,e4	e5
	(0,1,1,0,1)	e3	3	3	-	2	3			
	(0,1,1,1,0)	e4	3	3	2	-	3			
	(1,0,1,0,0)	e2	2	-	3	3	4			
	(1,1,0,1,1)	e5	4	4	3	3	-			
(-2,-2,1,1,3)	(0,0,0,0,0)	e5	4	4	3	3	-	e1,e2	e3,e4	e5
	(0,1,1,1,0)	e3	3	3	-	2	3			
	(1,0,1,1,0)	e4	3	3	2	-	3			
	(1,1,0,1,1)	e1	-	2	3	3	4			
	(1,1,1,0,1)	e2	2	-	3	3	4			
(-2,-3,1,1,2)	(0,0,0,0,0)	e3	3	3	-	2	3	e1,e2	e3,e4	e5
	(0,1,0,1,1)	e1	-	2	3	3	4			
	(0,1,1,0,1)	e2	2	-	3	3	4			
	(1,0,0,0,1)	e4	3	3	2	-	3			
	(1,0,1,1,0)	e5	4	4	3	3	-			

Tabelle 15: Die fünf Orthogonalvektor-Arten für Hyperebenen vom Typ (11,5,16) im Q_5.

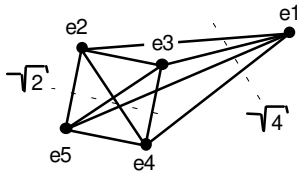


Abbildung 14: Hyperebene vom Typ (11,5,16) aus Q_5 mit Kantenlängen $\sqrt{2}$ und $\sqrt{4}$.

Für die Spur der Hyperebenen H der beiden ersten Orthogonalvektoren ergeben sich zwei Gruppen von Ecken aus Q_5: Die Ecken e2, e3, e4 und e5 die alle zueinander einen Raumabstand von $\sqrt{2}$ haben, sowie die Ecke e1 welche zu allen anderen Ecken einen Abstand von $\sqrt{4}$ hat (siehe Abbildung 14). Aufgrund dieser beiden verschiedenartigen Gruppen von Ecken aus H gibt es zwei verschiedene Arten von Orthogonalvektoren, die aber dieselbe Hyperebene erzeugen.

Elf Ecken aus Q_5 liegen unter H und dazu gibt es (für den Orthogonalvektor (-3,1,1,1,1)) insgesamt zehn benachbarte Ecken, die über H liegen. Die Spur von H im Q_5 hat also insgesamt 5+10 = 15 Eckpunkte.

Die folgende Tabelle listet diese zehn zusätzlichen Punkte auf:

Ecke unter H	benachbarte Ecke über H	Neuer Eckpunkt der Spur von H	Ecke unter H	benachbarte Ecke über H	Neuer Eckpunkt der Spur von H
(1,0,0,0,0)	keine	keiner	(1,0,1,0,1)	(0,0,1,0,1)	(2/3,0,1,0,1)
(1,0,0,0,1)	(0,0,0,0,1)	(1/3,0,0,0,1)	(1,0,1,1,0)	(0,0,1,1,0)	(2/3,0,1,1,0)
(1,0,0,1,0)	(0,0,0,1,0)	(1/3,0,0,1,0)	(1,1,0,0,1)	(0,1,0,0,1)	(2/3,1,0,0,1)
(1,0,1,0,0)	(0,0,1,0,0)	(1/3,0,1,0,0)	(1,1,0,1,0)	(0,1,0,1,0)	(2/3,1,0,1,0)
(1,1,0,0,0)	(0,1,0,0,0)	(1/3,1,0,0,0)	(1,1,1,0,0)	(0,1,1,0,0)	(2/3,1,1,0,0)
(1,0,0,1,1)	(0,0,0,1,1)	(2/3,0,0,1,1)			

Tabelle 16: Die elf unter H liegenden Ecken aus Q_5 (Orthogonalvektor: (-3,1,1,1,1)) und daraus resultierende neue Ecken der Spur von H.

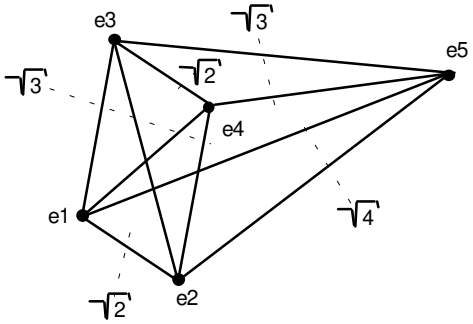


Abbildung 15: Hyperebene vom Typ (11,5,16) aus Q_5 mit Kantenlängen $\sqrt{2}$,$\sqrt{3}$ und $\sqrt{4}$.

Die Spur der Hyperebenen der restlichen Orthogonalvektoren ergeben insgesamt drei verschiedene Arten von Ecken aus Q_5. Die erste Gruppe bilden die Ecken e1 und e2, die zueinander den Abstand $\sqrt{2}$ haben. Die zweite Gruppe bilden die Ecken e3 und e4, welche ebenfalls zueinander den Abstand $\sqrt{2}$ haben. Die Abstände zwischen den Ecken aus diesen Gruppen beträgt jeweils $\sqrt{3}$. Die dritte Gruppe wird nur aus der Ecke e5 gebildet. Diese hat zu den Ecken aus der Gruppe 1 den Abstand $\sqrt{4}$ und zu den Ecken aus der Gruppe 2 den Abstand $\sqrt{3}$. Eine Skizze dieser Hyperebene ist in Abbildung 15 zu sehen. Da es diese 3 Gruppen von Ecken gibt, existieren auch hier drei verschiedene Arten von Orthogonalvektoren, die diese Hyperebene erzeugen.

Bemerkenswert ist nun, daß diese beiden Spuren im Q_5 voneinander verschieden sind. *Das bedeutet, daß derselbe Typ einer Hyperebene (hier (11,5,16)) verschiedene Spuren erzeugen kann!* Dies ist auch aus der verschiedenen Anzahl von Schnittpunkten von H mit Kanten aus Q_5 ersichtlich. Für den Orthogonalvektor (-1,-3,1,2,2) liegen elf Ecken aus Q_5 unter H und dazu gibt es insgesamt dreizehn benachbarte Ecken, die über H liegen. Hier hat die Spur von H im Q_5 also insgesamt 5+13 = 18 Eckpunkte.

In der Tabelle werden alle benachbarten Ecken und diese dreizehn zusätzlichen Punkte aufgelistet:

Ecke unter H	benachbarte Ecke über H	Neuer Eckpunkt der Spur von H	Ecke unter H	benachbarte Ecke über H	Neuer Eckpunkt der Spur von H
(1,0,0,0,0)	(1,0,0,0,1) (1,0,0,1,0)	(1,0,0,0,1/2) (1,0,0,1/2,0)	(1,1,1,0,0)	keine	keiner
(0,1,0,0,0)	keine	keine	(1,1,0,1,0)	(1,0,0,1,0)	(1,1/3,0,1,0)
(0,1,1,0,0)	(0,0,1,0,0)	(0,1/3,1,0,0)	(1,1,0,0,1)	(1,0,0,0,1)	(1,1/3,0,0,1)
(0,1,0,1,0)	(0,0,0,1,0) (0,1,0,1,1)	(0,2/3,0,1,0) (0,1,0,1,1/2)	(1,1,1,1,0)	(1,0,1,1,0) (1,1,1,1,1)	(1,2/3,1,1,0) (1,1,1,1,1/2)
(0,1,0,0,1)	(0,0,0,0,1) (0,1,0,1,1)	(0,2/3,0,0,1) (0,1,0,1/2,1)	(1,1,1,0,1)	(1,0,1,0,1) (1,1,1,1,1)	(1,2/3,1,0,1) (1,1,1,1/2,1)
(1,1,0,0,0)	keine	keine			

Tabelle 17: Die elf unter H liegenden Ecken aus Q_5 (Orthogonalvektor: (-1,-3,1,2,2)) und daraus resultierende neue Ecken der Spur von H.

5. Quelltexte des Programms

🖫 HYPER.PAS

```
   {-------------------------------------------------------------------

                        HYPERPLANES IN HYPERCUBES

   Diplomarbeit am Institut für Grundlagen der Informationsverarbeitung

   Betreuer    :  Ao. Univ.-Prof. DI. Dr. Franz Aurenhammer

   Diplomant   :  Oswin Aichholzer, Matr. Nr.: 8631450
   ___________________________________________________________________

   Programmteil:  HYPER.PAS, Hauptprogramm

   Letzte Änderung: 1992-08-31

   -------------------------------------------------------------------}


{-------------------------------------------------------------------------}
{
Programm zur Analyse von schrägen, durch den Nullpunkt laufenden
Hyperebenen im d-dimensionalen Einheitshyperwürfel.
Theorie, Programmschema und Aufbau der einzelnen Abschnitte wurden
in der Diplomarbeit ausführlich dargestellt.

Benötigt werden die Units MULTI_FO, TOOLS1 und TOOLS2, welche jeweils
mit entsprechenden Kommentaren erklärt werden bzw. in der Diplomarbeit
beschrieben sind. Die ebenfalls benötigte Unit TIMER ist für die
Programmfunktion selbst nicht notwendig: es stellt lediglich eine Art
Stoppuhr zur Verfügung, welche für lange Berechnungen eine Abschätzung
der notwendigen Zeit und damit des Berechnungsendes ermöglicht.

Das Programm wurde in Turbo Pascal 6.0 entwickelt und ist auf jedem
kompatiblen PC mit 640 KB Speicher lauffähig. Die Benutzerschnittstelle
wurde entsprechend den Programmanforderungen rudimentär gehalten, da
es sich um ein reines Hilfsprogramm für Berechnungen handelt und daher
ein übertriebener Aufwand in dieser Richtung nicht notwendig erscheint.
Weitere programmtechnische Erklärungen befinden sich bei den einzelnen
Routinen.
}
{-------------------------------------------------------------------------}

PROGRAM HYPER;

USES Crt,multi_fo,tools1,tools2,timer;

{ Die folgende Definition schließt eine detailiertere Auswertung aus
  und muß aus Speichergründen für Dimensionen >= 9 verwendet werden!.
```

```
  Dazu ist nach der öffnenden Klammer ein $-Zeichen einzusetzen. }

{ DEFINE keine_auswertung}


CONST logo = '##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##';
      min_dimension = 2; { Minimale Dimension ist zwei! }
      max_dimension = 8; { Maximal zu untersuchende Dimension hier anpassen}
      max_ecken     = 256;  {=2^max_dimension}

TYPE vektor = ARRAY [1..max_dimension] OF integer; {Typ eines Vektors}
     doppelvektor = ARRAY [-max_dimension..max_dimension] OF integer;
                    { Typ eines doppelten Vektors für Skalarprodukt-
                      berechnung mittels Graycode. }
{$IFNDEF keine_auswertung}
     HE_Typ_Feld = ARRAY [1..max_ecken DIV 2,min_dimension..max_ecken DIV 2]
                   OF longint; { Speichert in [i,j] die Anzahl von
                                 Hyperebenen des Typs (i,j,...). }
{$ENDIF}

VAR dimension,  { zu untersuchende Dimension }
    anz_ecken,  { Aktuelle Anzahl von Ecken des Hyperwürfels }
    kmax      : integer; {maximaler Koordinatenwert des Orthogonalvektors }
    graydiff  : ARRAY [1..max_ecken-1] OF integer;
                { Position der Änderung des Graycodes von n-1 auf n }
    gray      : ARRAY [1..max_ecken-1] OF vektor;
                { Graycode }
    fakultaet : ARRAY [0..max_dimension] OF longint;
                { Enthält n! zur schnelleren Verwendung }
    HE        : ARRAY [min_dimension..max_ecken DIV 2] OF longint;
                { Anzahl von Hyperebenen mit n Ecken }
    overflow_count : ARRAY [min_dimension..max_ecken DIV 2] OF word;
                { Höherwertige "Stelle" von HE }
{$IFNDEF keine_auswertung}
    anz_art   : ARRAY [1..max_ecken DIV 2,min_dimension..max_ecken DIV 2]
                OF word; {Anzahl verschiedener Orthogonalvektoren je Typ}
    HE_typen  : ^HE_Typ_Feld;{Speichert in [i,j] die Anzahl von Hyperebenen
                              des Typs (i,j,...). Muß wegen des großen
                              Speicherbedarfs dynamisch definiert werden.}
    test_anzahl,
    ende_anzahl : longint; {Zähler für die Durchlaufe des Generators}
{$ELSE}
    test_anzahl,
    ende_anzahl : double;  {für d>=9 wäre longint zu klein als Zähler}
{$ENDIF}
    max_skalar  : integer; {Größtes, tatsächlich ermitteltes Vektorelement}
    ausgabe     : TEXT;    {Ausgabe auf Datei/Bildschirm/Drucker}
    anzeige_ein : boolean; {TRUE falls Zeitanzeige aktiv}
    erweiterte_ausgabe,    {TRUE falls vollst. Ausgabe mit allen OVs}
    warten      : BOOLEAN;

{------------------------------------------------------------------------}

PROCEDURE INIT;

{ Initialisieren aller Zähl-Felder, der Fakultätsberechnung sowie
  des Graycodes und der Graycode-Differenzpositionen. }
```

```
VAR i,k,indx,bitpos,
    step,ende  : integer;
    j          : longint;

BEGIN
  {Init Gray-Feld}
  ende:=max_ecken-1;
  FOR k:=1 TO max_dimension DO BEGIN
    indx:=zweihoch(k-1);
    step:=zweihoch(k);
    bitpos:=k;
    REPEAT
      graydiff[indx]:=bitpos;
      INC(indx,step);
      bitpos:=-bitpos;
    UNTIL indx>ende;
  END;
  gray[1][1]:=1;
  FOR i:=2 TO max_dimension DO
    gray[1][i]:=0;
  FOR k:=2 TO max_ecken-1 DO BEGIN
    gray[k]:=gray[PRED(k)];
    gray[k][ABS(graydiff[k])]:=1-gray[k][ABS(graydiff[k])];
  END;

  {Initialisieren der Fakultäten}
  fakultaet[0]:=1;
  fakultaet[1]:=1;
  FOR j:=2 TO max_dimension DO
    fakultaet[j]:=fakultaet[j-1]*j;

  {Initialisieren der Zählvariablen}
{$IFNDEF keine_auswertung}
  NEW(he_typen);
{$ENDIF}
  FOR i:=min_dimension TO max_ecken DIV 2 DO BEGIN
    he[i]:=0;
    overflow_count[i]:=0;
{$IFNDEF keine_auswertung}
    FOR j:=1 TO max_ecken DIV 2 DO BEGIN
      anz_art[j,i]:=0;
      HE_typen^[j,i]:=0;
    END;
{$ENDIF}
  END;
  max_skalar:=0;
  test_anzahl:=0;
END; {INIT}

{-------------------------------------------------------------------}

FUNCTION GENERATOR_SCHRITTE(d,kmax:integer):double;

{ Berechnet die vom Generator zu erzeugende Anzahl von Orthogonalvektoren.}

VAR i     : integer;
    hlp,
    summe : double;
```

```
BEGIN
  summe:=0;
  FOR i:=1 TO (d DIV 2) DO
    summe:=summe+anzmultifor(i,kmax,0)*anzmultifor(d-i,kmax,0);
  IF NOT odd(d) THEN BEGIN
    hlp:=anzmultifor(d DIV 2,kmax,0);
    summe:=summe-(hlp*(hlp-1))/2;
  END;
  generator_schritte:=summe;
END; {GENERATOR_SCHRITTE}

{-------------------------------------------------------------------}

PROCEDURE VEKTOR_TESTEN(vek:vektor; halbieren:boolean);

{ Testet den übergebenen Orthogonalvektor, ob auf ihm zumindest
  d Ecken liegen und ob diese eine Hyperebene aufspannen. Ist dies
  der Fall, so wird die Hyperebene mit dem entsprechenden Multiplikator
  eingetragen. Die Variable halbieren gibt dabei an, wann der Vektor
  zu sich selbst negativ ist, der Multiplikator also halbiert werden muß.}

VAR i,j        : integer;
    SeiteA,SeiteB,        { Anzahl von Ecken über/unter H }
    skalarprod : integer;
    anz_liste  : integer; { Anzahl von Ecken auf H }
    liste      : ARRAY [1..max_ecken DIV 2] OF vektor;
                 { Liste der auf H liegenden Ecken }
    vielfach   : longint;
                 { Multinomialkoeffizient=Anzahl von Hyperebenen }
    vek2       : doppelvektor;
                 { Hilfsvektor für Skalarproduktbildung }


  FUNCTION RANG_TEST:BOOLEAN;

  { Diese Funktion testet den Rang der gefundenen Ecken von H. Ist
    dieser Rang=1, so wird eine Hyperebene aufgespannt (TRUE),
    ansonsten ist rang_test FALSE.
    Die Berechnung erfolgt mittels eines angepaßten, beim Lösen von
    linearen Gleichungssystemen verwendeten Gauß-Eliminationsverfahren.}


  VAR zeile,spalte,i,j : integer;
      max,help1,help2  : integer;
      rangverlust      : boolean;

  BEGIN
    rangverlust:=FALSE;
    {Elimination nach Gauß}
    zeile:=1;
    FOR spalte:=1 TO dimension DO BEGIN
      j:=0;
      max:=maxint;
      FOR i:=zeile TO anz_liste DO {Zeile mit kleinstem Wert suchen}
        IF (liste[i][spalte]<>0) AND (ABS(liste[i][spalte])<max) THEN BEGIN
          max:=ABS(liste[i][spalte]);
          j:=i;
```

```
        END;
      IF j=0 THEN BEGIN
        { Hier muß ein Parameter verwendet werden. }
        IF rangverlust THEN BEGIN {Bereits zweiter Rangverlust!}
          rang_test:=FALSE;
          EXIT;
        END ELSE
          rangverlust:=TRUE;
      END ELSE BEGIN
        IF j<>zeile THEN
          FOR i:=spalte TO dimension DO BEGIN
            {Tauschen der Zeilen zeile und j}
            help1:=liste[zeile][i];
            liste[zeile][i]:=liste[j][i];
            liste[j][i]:=help1;
          END;
        { Elemente unterhalb von liste[zeile,spalte] auf 0 bringen }
        { Dabei werden die restlichen Werte ganzzahlig, aber möglichst
          klein gehalten }
        FOR i:=SUCC(zeile) TO anz_liste DO
          IF liste[i][spalte]<>0 THEN BEGIN
            max:=ggT(liste[i][spalte],liste[zeile][spalte]);
            help1:=liste[zeile][spalte] DIV max;
            help2:=liste[i][spalte] DIV max;
            liste[i][spalte]:=0;
            FOR j:=SUCC(spalte) TO dimension DO
              liste[i][j]:=help1*liste[i][j]-help2*liste[zeile][j];
        END;
        INC(zeile);
      END;
    END; {FOR spalte:=...}
    IF NOT rangverlust THEN BEGIN
      {Dieser Fall darf nie eintreten, es wäre ein Programmfehler ...}
      writeln;
      writeln('Fehler: Kein Rangverlust aufgetreten ...');
      laerm;
      pause;
    END;
    rang_test:=TRUE;
  END; {RANG_TEST}


BEGIN {VEKTOR TESTEN}
  { Orthogonale Eckpunkte über Skalarproduktberechnung mittels
    Graycode ermitteln }
  FOR i:=1 TO dimension DO BEGIN
    vek2[i]:=vek[i];
    vek2[-i]:=-vek[i];
  END;
  anz_liste:=0;
  SeiteA:=0;
  skalarprod:=0;
  FOR i:=1 TO anz_ecken-1 DO BEGIN
    INC(skalarprod,vek2[graydiff[i]]);
    IF skalarprod=0 THEN BEGIN
      INC(anz_liste);
      liste[anz_liste]:=gray[i];
    END ELSE IF skalarprod>0 THEN
```

```
      INC(SeiteA);
  END;
  { Falls weniger als dimension-1 Ecken gefunden wurden, kann keine
    Hyperebene erzeugt werden. }
  IF SUCC(anz_liste)<dimension THEN EXIT; { Zu wenige Ecken }

  { Rang der Eckvektoren ermitteln }
  IF NOT rang_test THEN EXIT; { Keine HE, da Rang <> dimension-1 }

  { HE gefunden! Vielfachheit mittels Multinomialkoeffizienten berechnen. }
  { Gleichzeitige Ermittlung des größten Koordinatenwerts des Orthogonalv.}
  vielfach:=fakultaet[dimension];
  IF ABS(vek[1]) > max_skalar THEN
    max_skalar:=ABS(vek[1]);
  j:=1;
  FOR i:=2 TO dimension DO BEGIN
    IF (vek[i]=vek[PRED(i)]) THEN
      INC(j)
    ELSE BEGIN
      IF ABS(vek[i]) > max_skalar THEN
        max_skalar:=ABS(vek[i]);
      IF (j<>1) THEN BEGIN
        vielfach:=vielfach DIV fakultaet[j];
        j:=1;
      END;
    END;
  END;
  IF (j<>1) THEN
    vielfach:=vielfach DIV fakultaet[j];
  IF halbieren THEN
    vielfach:=vielfach DIV 2;

  { Ecken über und unter H berechnen }
  INC(anz_liste);
  SeiteB:=anz_ecken-anz_liste-SeiteA;
  IF SeiteA>SeiteB THEN BEGIN
    i:=SeiteA; SeiteA:=SeiteB; SeiteB:=i;
  END;
  { Hyperebene eintragen }
  IF (MaxLongint-he[anz_liste]) < vielfach THEN BEGIN
    INC(overflow_count[anz_liste]);
    DEC(he[anz_liste],MaxLongint);
  END;
  INC(he[anz_liste],vielfach);
{$IFNDEF keine_auswertung}
  INC(anz_art[SeiteA,anz_liste]);
  INC(he_typen^[SeiteA,anz_liste],vielfach);
{$ENDIF}
  IF erweiterte_ausgabe THEN BEGIN
    { Orthogonalvektor ausgeben }
    write(ausgabe,'(');
    FOR i:=1 TO dimension DO
      write(ausgabe,vek[i]:3);
    writeln(ausgabe,'); ',vielfach:7,' mal;   Typ: (',
            SeiteA,',',anz_liste,',',SeiteB,')');
  END;
END; {VEKTOR_TESTEN}
```

```
{-------------------------------------------------------------------}

PROCEDURE GENERATOR;

{ Erzeugt alle möglichen d-dimensionalen Orthogonalvektoren mit
  Koordinatenwerten v[i]:
    *) 1<=ABS(v[i])<=kmax
    *) GGT(v[i])=1;
    *) Inversionsfrei
    *) Permutationsfrei (d.h. für alle möglichen Permutationen wird nur ein
                         stellvertretender Vektor ausgegeben!!!)
  Verwendet wird dazu die MULTI-FOR-Schleife. Eine Ausführliche Beschreibung
  dieser Routine ist in der Diplomarbeit zu finden.
}

VAR neg,pos  : for_type; { Laufvariablen für negative und
                           positive Koordinatenwerte}
    anzneg,i : integer;  { Anzahl negativer Koordinatenwerte }

  PROCEDURE MACHE_VEKTOR(halb:boolean);
  { Erzeugt aus den Laufvariablen neg und pos einen Vektor und testet
    in auf den größten gemeinsamen Teiler seiner Elemente. Ist dieser
    Eins, so wird die Routine VEKTOR_TESTEN(halb) aufgerufen.
    Weiters wird für ca. alle zweitausend Vektoren die benötigte Zeit,
    die Anzahl erledigter Vektoren in % und die voraussichtliche
    Dauer angezeigt (Routine show_info aus timer.pas). }

  VAR k,teiler : integer;
      vek      : vektor;

  BEGIN
    {Anzeige des Berechnungsfortschrittes }
{$IFNDEF keine_auswertung}
    INC(test_anzahl);
    IF anzeige_ein AND ((test_anzahl AND $07FF) = 0) THEN
      show_info(25,20,100*(test_anzahl/ende_anzahl));
{$ELSE}
    test_anzahl:=test_anzahl+1;
    IF anzeige_ein AND (frac(test_anzahl/2000) < 0.0005) THEN
      show_info(25,20,100*(test_anzahl/ende_anzahl));
{$ENDIF}
    { GGT ermitteln }
    teiler:=neg[1];
    FOR k:=1 TO anzneg DO BEGIN
      vek[k]:=-neg[k];
      IF teiler<>1 THEN teiler:=ggt(neg[k],teiler);
    END;
    FOR k:=1 TO dimension-anzneg DO BEGIN
      vek[k+anzneg]:=pos[k];
      IF teiler<>1 THEN teiler:=ggt(pos[k],teiler);
    END;
    IF teiler=1 THEN
      vektor_testen(vek,halb);
  END; {MACHE_VEKTOR}

BEGIN {GENERATOR}
  FOR anzneg:=1 TO ((dimension-1) DIV 2) DO
    IF multifor(neg,anzneg,1,kmax,0) THEN REPEAT
```

```
      IF multifor(pos,dimension-anzneg,1,kmax,0) THEN REPEAT
        mache_vektor(FALSE);
      UNTIL endmultifor(pos);
    UNTIL endmultifor(neg);
  IF NOT odd(dimension) THEN BEGIN {Gerade Anzahl noch extra behandeln}
    anzneg:=dimension DIV 2;
    IF multifor(neg,anzneg,1,kmax,0) THEN REPEAT
      IF multifor(pos,anzneg,1,kmax,0) THEN; {dummy}
      { Manipulation um doppelte Vektoren auszuschließen: }
      { Vektor pos beginnt erst ab Vektor neg ! }
      FOR i:=1 TO anzneg DO
        pos[i]:=neg[i];
      { Der Vektor mit genau gleichem negativen und positiven Teilen muß
        extra behandelt werden ! }
      mache_vektor(TRUE);
      IF NOT endmultifor(pos) THEN REPEAT
        mache_vektor(FALSE);
      UNTIL endmultifor(pos);
    UNTIL endmultifor(neg);
  END;
END; {GENERATOR}

{-------------------------------------------------------------------}

PROCEDURE HEADER;

{ Initialisieren des Bildschirms und einlesen der Grundlegenden
  Parameter. Neben der Dimension und dem maximalen Koordinatenwert
  wird auch die gewünschte Ausgaberichtung (Datei/Bildschirm/Drucker)
  abgefragt. Weiters erfolgt die Abfrage, ob alle ermittelten
  Orthogonalvektoren ausgegeben werden sollen und ob eine Zeitanzeige
  während der Berechnung gewünscht ist. }

VAR key       : CHAR;
    dateiname : string;

BEGIN
  ClrScr;
  writeln(logo);
  writeln;
  write('Dimension (',min_dimension,'..',max_dimension,') = ');
  readln(dimension);
  IF (dimension<min_dimension) OR (dimension>max_dimension) THEN BEGIN
    writeln;
    writeln('Dimension außerhalb der zulässigen Grenzen!');
    writeln('Programm beendet ...');
    writeln;
    HALT;
  END;
  anz_ecken:=zweihoch(dimension);
  writeln;
  write('Maximaler Koordinatenwert des Orthogonalvektors = ');
  readln(kmax);
  writeln;
  writeln('Soll die Auswertung auf den Bildschirm, Drucker oder');
  write('in ein File erfolgen (B/D/F) ? ');
  REPEAT
    key:=UpCase(ReadKey);
```

```
  UNTIL key IN ['B','D','F',ESC];
  writeln(key);
  writeln;
  warten:=FALSE;
  CASE key OF
    'B' : BEGIN
            AssignCrt(ausgabe);
            warten:=TRUE;
          END;
    'D' : BEGIN
            write('Parallele Schnittstelle 1 oder 2 ? ');
            REPEAT
              key:=ReadKey;
            UNTIL key IN ['1','2',ESC];
            writeln(key);
            CASE key OF
              '1' : assign(ausgabe,'LPT1');
              '2' : assign(ausgabe,'LPT2');
              ESC : HALT;
            END;
          END;
    'F' : BEGIN
            write('Dateiname: ');
            readln(dateiname);
            assign(ausgabe,dateiname);
          END;
    ESC : HALT;
  END; {CASE}
  erweiterte_ausgabe:=
   jakey('Erweiterte Auswertung (vollst. Liste aller Orthogonalvektoren) ');
  writeln;
  anzeige_ein:=jakey('Zeitanzeige während des Berechnungsvorganges ');
  writeln;
  writeln;
{$IFNDEF keine_auswertung}
  ende_anzahl:=round(generator_schritte(dimension,kmax));
  writeln('Benötigte Durchläufe: ',ende_anzahl);
{$ELSE}
  ende_anzahl:=generator_schritte(dimension,kmax);
  writeln('Benötigte Durchläufe: ',ende_anzahl:1:0);
{$ENDIF}
  writeln;
  REWRITE(ausgabe);
  IF erweiterte_ausgabe THEN BEGIN
    writeln(ausgabe,logo);
    writeln(ausgabe);
    writeln(ausgabe,date,'    ',time);
    writeln(ausgabe);
    writeln(ausgabe,
      'Vollständige Liste der schrägen Hyperebenen-Typen im Nullpunkt für
Q',
      dimension,':');
    writeln(ausgabe);
  END;
END; {HEADER}

{-------------------------------------------------------------------}
```

```
PROCEDURE AUSWERTUNG;

{ Als Kennung für das Berechnungsende folgt eine akustisches Signal.
  Aus den gewonnenen Daten wird hier e0(d) berechnet und ausgegeben.
  Weiters werden alle Hyperebenen-Typen in zusammengefaßter Form,
  nach Eckanzahl sortiert ausgegeben. Erfolgt die Ausgabe auf den
  Bildschirm, so wird nach jeweils 20 Zeilen auf einen Tastendruck
  gewartet. Weiters wird der größte gefundene Koordinatenwert des
  Orthogonalvektors ausgegeben. }

VAR E0,Summe,
    anz_ecken2 : double;
    zeilen,
    i,j        : longint;

BEGIN
  { e0(d) berechnen }
  e0:=0.0;
  anz_ecken2:=anz_ecken;
  FOR i:=min_dimension TO anz_ecken DIV 2 DO
    e0:=e0 + (anz_ecken2*he[i]) / i
           +  overflow_count[i]*((anz_ecken2*MaxLongint)/i);
  IF erweiterte_ausgabe THEN BEGIN
    writeln(ausgabe);
    writeln(ausgabe);
  END;
  { AUSGABE }
  writeln(ausgabe,logo);
  writeln(ausgabe);
  writeln(ausgabe,date,'    ',time);
  writeln(ausgabe);
  writeln(ausgabe,'Auswertung der schrägen Hyperebenen für
Q',dimension,':');
  writeln(ausgabe);
  writeln(ausgabe,'e0(',dimension,') = ',e0:10:3);
  writeln(ausgabe);
  laerm;
  IF warten THEN pause;
  writeln(ausgabe);
  writeln(ausgabe,'Schräge Hyperebenen in Q',dimension,':');
  writeln(ausgabe);
  writeln(ausgabe,'#Pkt      #HE      #Typ');
  zeilen:=4;
  { Zusammengefaßte Ausgabe der HE-Typen }
  FOR i:=min_dimension TO anz_ecken DIV 2 DO BEGIN
    IF he[i]<>0 THEN BEGIN
      writeln(ausgabe);
      Summe := (anz_ecken2*he[i]) / i
               +  overflow_count[i]*((anz_ecken2*MaxLongint)/i);
      writeln(ausgabe,i:4,'  ',Summe:9:0);
      INC(zeilen,2);
      IF warten AND ((zeilen MOD 20) = 0) THEN pause;
{$IFNDEF keine_auswertung}
      FOR j:=1 TO anz_ecken DIV 2 DO
        IF anz_art[j,i]<>0 THEN BEGIN
          writeln(ausgabe,'      ',(anz_ecken2*he_typen^[j,i]) / i:9:0,
                  ' mal Typ (',j:2,',',i:2,',',anz_ecken-j-i:2,')  [ ',
                  anz_art[j,i],' verschiedene OV ]');
```

```
          Summe:=Summe - (anz_ecken2*he_typen^[j,i])/i;
          INC(zeilen);
          IF warten AND ((zeilen MOD 20) = 0) THEN pause;
        END;
      IF Summe > 0.1 THEN BEGIN
        { Sollte nie Auftreten! }
        writeln('Fehler bei Kontrollsumme für HE mit ',i,
                ' Punkte aufgetreten!');
        writeln('Differenz: ',Summe:9:2);
        laerm;
        pause;
      END;
{$ENDIF}
    END;
  END;
  writeln(ausgabe);
  writeln(ausgabe,'Größtes aufgetretenes Vektorelement: ',max_skalar,
                  '  (kmax=',kmax,')');
  writeln(ausgabe);
  writeln(ausgabe,'   - - - -   E N D E   - - - -    ');
  IF warten THEN pause;
END; {AUSWERTUNG}

{-----------------------------------------------------------------------}

BEGIN {MAIN}
  init;        { Initialisieren }
  header;      { Parameter einlesen }
  writeln('Berechnung läuft ...');
  writeln;
  start_watch; { "Stopuhr" starten }
  generator;   { Berechnung durchführen }
  writeln;
  writeln;
  writeln('Benötigte Zeit: ',time_to_strng(stop_watch));
  writeln;
  writeln('Auswertung läuft ...');
  writeln;
  auswertung;  { Auswertung und Ausgabe der Ergebnisse }
  CLOSE(ausgabe);
{$IFNDEF keine_auswertung}
  DISPOSE(HE_typen);
{$ENDIF}
END. {MAIN}
```

🖫 TOOLS1.PAS

```
   {-------------------------------------------------------------------

                        HYPERPLANES IN HYPERCUBES

   Diplomarbeit am Institut für Grundlagen der Informationsverarbeitung

   Betreuer    :  Ao. Univ.-Prof. DI. Dr. Franz Aurenhammer

   Diplomant   :  Oswin Aichholzer, Matr. Nr.: 8631450
   ___________________________________________________________________

   Programmteil:  Unit TOOLS1.PAS, Hilfsprogramme

   Letzte Änderung: 1992-08-31

   -------------------------------------------------------------------}

UNIT TOOLS1;

{ Diese Unit enthält Hilfsroutinen zur Lösung von linearen Gleichungs-
  systemen, zur Bestimmung des größten gemeinsamen Teilers zweier Zahlen
  und zur Berechnung des Binomialkoeffizienten; weiters eine Routine zur
  Berechnung von Zweierpotenzen und zur Berechnung der 1-Bits einer
  Binärzahl.
  Das Gleichungssystem wurde speziell zur Untersuchung folgender Frage
  entwickelt: Bilden Ecken aus dem Einheitshyperwürfel (übergeben als
  0-1-Werte der Matrix A) eine Hyperebene durch den Nullpunkt (b=0)?
  Liegt eine einparametrige Lösung vor, so ist dies der Fall, sonst
  wird der Einheitshyperwürfel selbst oder ein niederdimensionaler
  Körper aufgespannt.
  Erklärungen zu den Aufruf-Parametern befinden sich bei den
  Procedure/Function-Vereinbarungen.
}
{-------------------------------------------------------------------------}

INTERFACE

CONST lgs_size = 10; { Bestimmt die Größe des Gleichungssystems, also
                       die maximale Dimension des zu untersuchenden
                       Einheitshyperwürfels. }

TYPE  lgs_typ    = integer; { Typ der Vektor und Matrizenelemente }
      lgs_vektor = ARRAY [1..lgs_size] OF lgs_typ;
                   {Typ eines Vektors}
      lgs_matrix = ARRAY [1..lgs_size,1..lgs_size] OF lgs_typ;
                   {Typ einer Matrix}

{-------------------------------------------------------------------------}

PROCEDURE LLGS(n:integer; A:lgs_matrix;
               VAR x:lgs_vektor; VAR anz_param:integer);
{ Lösung Lineare Gleichungssysteme }
```

```
FUNCTION GGT(a,b:integer):integer;
{ Größter Gemeinsamer Teiler }

FUNCTION BINOM(n,m:integer):comp;
{ Binomialkoeffizient }

FUNCTION ZWEIHOCH(x:word):longint;
{ Zweierpotenz }

FUNCTION BIT_ANZAHL(x:longint):byte;
{ Anzahl von 1-Bits in x }

{-------------------------------------------------------------------------}

IMPLEMENTATION

{-------------------------------------------------------------------------}


PROCEDURE LLGS(n:integer; A:lgs_matrix;
               VAR x:lgs_vektor; VAR anz_param:integer);

{ Auflösen eines linearen Gleichungssystems Ax=b mittels Gauß-Elimination.
  Übergeben werden die Größe n und die Matrix A. Dabei gilt a[ij] ist
  entweder 0 oder 1 und b=(0,0,..,0) (homogenes Gleichungssystem).
  Als Ergebnis wird der Lösungsvektor x rückübergeben.
  Ist das System eindeutig lösbar, so ist anz_param=0 und x=(0,0,..,0),
  ansonsten gibt die Variable anz_param die Anzahl der Parameter
  (=Freiheitsgrade) an. Ist diese 1, so wird durch die Übergebenen Ecken
  eine Hyperebene gebildet und x bildet dazu den Orthogonalvektor.
}

VAR zeile,spalte,i,j : integer; { Laufvariablen }
    position         : integer; { Position des Lösungsparameters }
    max,help1,help2  : lgs_typ; { Hilfsvariablen }

BEGIN
  anz_param:=0;
  position:=0;

  {Elimination nach Gauß}
  zeile:=1;
  FOR spalte:=1 TO n DO BEGIN
    j:=0;
    max:=maxint;
    FOR i:=zeile TO n DO {Zeile mit kleinstem Wert in Spalte spalte suchen}
      IF (A[i,spalte]<>0) AND (ABS(A[i,spalte]) < max) THEN BEGIN
        max:=ABS(A[i,spalte]);
        j:=i;
      END;
    IF j=0 THEN BEGIN
      { Hier muß ein Parameter verwendet werden. }
      INC(anz_param);
      position:=spalte;
    END ELSE BEGIN
      FOR i:=spalte TO n DO BEGIN {Tauschen der Zeilen zeile und j}
        help1:=A[zeile,i]; A[zeile,i]:=A[j,i]; A[j,i]:=help1;
      END;
```

```
      { Elemente unterhalb von A[zeile,spalte] auf 0 bringen }
      { Dabei werden die restlichen Werte ganzzahlig, aber möglichst
        klein gehalten }
      FOR i:=SUCC(zeile) TO n DO IF A[i,spalte]<>0 THEN BEGIN
        max:=ggT(A[zeile,spalte],A[i,spalte]);
        help1:=A[zeile,spalte] DIV max;
        help2:=A[i,spalte] DIV max;
        A[i,spalte]:=0;
        FOR j:=SUCC(spalte) TO n DO
          A[i,j]:=help1*A[i,j]-help2*A[zeile,j];
      END;
      INC(zeile);
    END;
  END; {FOR spalte:=...}

  { Jetzt liegt eine rechte, obere Dreiecksmatrix vor }
  FOR i:=1 TO n DO
    x[i]:=0;
  IF anz_param=1 THEN BEGIN
    { Es wird eine Hyperebene aufgespannt}
    x[position]:=1; {variabler Parameter zu Beginn auf 1 setzen}

    { Restliche Werte des Orthogonalvektors aus oberer Dreiecksmatrix
      berechnen. Damit alle Werte ganzzahlig sind, werden sie mit
      entsprechenden Faktoren multipliziert (der variable Parameter
      wird damit größer als eins) }
    FOR zeile := PRED(position) DOWNTO 1 DO BEGIN
      help1:=0;
      FOR spalte:=SUCC(zeile) TO position DO
        INC(help1,A[zeile,spalte]*x[spalte]);
      IF help1=0 THEN
        x[zeile]:=0
      ELSE BEGIN
        help2:=ABS(A[zeile,zeile] DIV ggt(help1,A[zeile,zeile]));
        x[zeile]:=(-help1*help2) DIV A[zeile,zeile];
        IF help2<>1 THEN
          FOR i:=SUCC(zeile) TO position DO
            x[i]:=x[i]*help2;
      END;
    END;
    { Normieren: niedrigste Dimension<>0 wird positiv gesetz. Damit ist
      der Orthogonalvektor eindeutig festgelegt.}
    i:=1;
    WHILE x[i]=0 DO INC(i);
    IF x[i]<0 THEN
      FOR i:=i TO position DO
        x[i]:=-x[i];
  END;
END;  { PROCEDURE LLGS }


{-------------------------------------------------------------------------}

FUNCTION GGT(a,b:integer):integer;

{ Berechnung des größten gemeinsamen Teilers von a und b nach Euklid}
{ b<>0 ! }
```

```
VAR rest:integer;

BEGIN
  a:=ABS(a);
  b:=ABS(b);
  REPEAT
    rest:=a MOD b;
    a:=b;
    b:=rest;
  UNTIL rest=0;
  ggt:=a;
END; {GGT}

{------------------------------------------------------------------------}

FUNCTION BINOM(n,m:integer):comp;

{Berechnung des Binomialkoeffizienten von n über m. Um auch größere Zahlen
 verarbeiten zu können, erfolgt die Berechnung nicht über Fakultäten,
 sondern mittels fortlaufender Bruchbildungen. Das Ergebnis ist vom
 genauerem Typ comp. }

VAR i:integer;
    bino:comp;

BEGIN
  bino:=1;
  FOR i:=1 TO m DO BEGIN
    bino:=bino*n/i;
    DEC(n);
  END;
  binom:=bino;
END; {BINOM}

{------------------------------------------------------------------------}

FUNCTION ZWEIHOCH(x:word):longint;

{ Berechnung der Zweierpotenz 2^x. }

VAR hlp:longint;

BEGIN
  hlp:=1;
  hlp:= hlp SHL x;
  zweihoch:=hlp;
END; {ZWEIHOCH}

{------------------------------------------------------------------------}

FUNCTION BIT_ANZAHL(x:longint):byte;

{ Berechnung der Anzahl von 1-Bits in der binären Zahl x. }

VAR anz:byte;

BEGIN
  anz:=0;
```

```
  WHILE x<>0 DO BEGIN
    IF odd(x) THEN INC(anz);
    x:=x SHR 1;
  END;
  bit_anzahl:=anz;
END; {BIT_ANZAHL}

{-------------------------------------------------------------------------}

BEGIN { UNIT TOOLS1 }

END.  { UNIT TOOLS1 }
```

TOOLS2.PAS

```
   {-------------------------------------------------------------------

                        HYPERPLANES IN HYPERCUBES

   Diplomarbeit am Institut für Grundlagen der Informationsverarbeitung

   Betreuer    :  Ao. Univ.-Prof. DI. Dr. Franz Aurenhammer

   Diplomant   :  Oswin Aichholzer, Matr. Nr.: 8631450
   ___________________________________________________________________

   Programmteil:  Unit TOOLS2.PAS, Hilfsprogramme für Ein/Ausgabe

   Letzte Änderung: 1992-08-31

   -------------------------------------------------------------------}

UNIT TOOLS2;

{ Diese Unit enthält Hilfsroutinen zur Ein-/Ausgabe sowie zur
  Zeit und Datumsermittlung. }

{-----------------------------------------------------------------------}

INTERFACE

USES CRT,DOS;

CONST ESC=#27;

{-----------------------------------------------------------------------}

FUNCTION JAKEY(txt:string):BOOLEAN;
{ Ausgeben der Meldung txt und warten auf eine J/N Eingabe. }

PROCEDURE PAUSE;
{ Ausgeben einer Pausenmeldung und warten auf Tastendruck }

PROCEDURE LAERM;
{ Ausgeben eines akustischen Signals }

FUNCTION  DATE:STRING;
{ Aktuelles Datum als String }

FUNCTION  TIME:STRING;
{ Aktuelle Zeit als String }

{-----------------------------------------------------------------------}

IMPLEMENTATION

{-----------------------------------------------------------------------}
```

```
FUNCTION JAKEY(txt:string):BOOLEAN;
{ Ausgabe der Meldung txt mit dem Anhang "(J/N) ?". Anschließend wird auf
  eine Taste "J", "N" oder ESC gewartet. Das Ergebnis ist TRUE, falls j
  oder J eingegeben wurde, ansonsten FALSE. }

VAR key:char;

BEGIN
  write(txt,' (J/N) ? ');
  REPEAT
    key:=UpCase(ReadKey);
  UNTIL key IN [ESC,'J','N'];
  IF key='J' THEN writeln(' J')
             ELSE writeln(' N');
  jakey:=(key='J');
END; {JAKEY}

{-------------------------------------------------------------------------}

PROCEDURE PAUSE;
{ Ausgeben der Meldung "Bitte Taste drücken ..." und warten auf einen
  Tastendruck. Nach dem Drücken einer Taste wird diese Meldung wieder
  gelöscht. }

BEGIN
  write('Bitte Taste drücken ...');
  IF ReadKey=#0 THEN IF ReadKey=#0 THEN;
  GotoXY(1,WhereY);
  ClrEOL;
END; {PAUSE}

{-------------------------------------------------------------------------}

PROCEDURE LAERM;
{ Ausgabe eines Vierton-Signals. }
{ Verwendung zur Kennzeichnung des Programmendes nach langen Berechnungen.}

CONST delaytime=80; {Dauer eines Tons in ms}

VAR i:integer;

BEGIN
  FOR i:=1 TO 3 DO BEGIN
    Sound(220); Delay(delaytime);
    Sound(440); Delay(delaytime);
    Sound(660); Delay(delaytime);
    Sound(440); Delay(delaytime);
  END;
  NoSound;
END; {LAERM}

{-------------------------------------------------------------------------}

FUNCTION DATE:STRING;
{ Aktuelles Systemdatum als String.}

VAR yy,mm,dd,dw:word;
    yys,mms,dds:string[4];
```

```
BEGIN {Function Date}
  GetDate(yy,mm,dd,dw);
  STR(yy,yys);
  STR(mm,mms);IF mm<10 THEN mms:='0'+mms;
  STR(dd,dds);IF dd<10 THEN dds:='0'+dds;
  date:=yys+'-'+mms+'-'+dds;
END; {DATE}

{-------------------------------------------------------------------------}

FUNCTION TIME:STRING;
{Aktuelle Systemzeit als String.}

VAR hh,mm,ss,s1 : word;
    hhs,mms,sss : string[2];

BEGIN {Function Time}
  GetTime(hh,mm,ss,s1);
  STR(hh,hhs);IF hh<10 THEN hhs:='0'+hhs;
  STR(mm,mms);IF mm<10 THEN mms:='0'+mms;
  STR(ss,sss);IF ss<10 THEN sss:='0'+sss;
  time:=hhs+':'+mms+':'+sss;
END; {TIME}

{-------------------------------------------------------------------------}

BEGIN {UNIT}

END.  {UNIT}
```

MULTI_FO.PAS

```
   {-------------------------------------------------------------------

                        HYPERPLANES IN HYPERCUBES

   Diplomarbeit am Institut für Grundlagen der Informationsverarbeitung

   Betreuer    :  Ao. Univ.-Prof. DI. Dr. Franz Aurenhammer

   Diplomant   :  Oswin Aichholzer, Matr. Nr.: 8631450
   ___________________________________________________________________

   Programmteil:  Unit MULTI_FO.PAS, Implementation der MULTI-FOR-Schleife

   Letzte Änderung: 1992-08-31

   -------------------------------------------------------------------}


UNIT MULTI_FO;

{ In dieser Unit befindet sich die vollständige Implementation einer
  MULTI-FOR-Schleife. Verwendungszweck und etwas Theorie dazu sind
  ausführlich in den entsprechenden Abschnitten der Diplomarbeit
  ausgeführt.
  Die MULTI-FOR-Schleife hat folgende Gestalt:

    Simulation einer variablen Anzahl von FOR-Schleifen :
    FOR i[1]:=start TO ende DO
      FOR i[2]:=i[1]+diff TO ende DO
        FOR i[3]:=i[2]+diff TO ende DO
          ......
            FOR i[anz] := i[anz-1]+diff TO ende DO BEGIN
               [ Schleifenkörper ... ]
            END;

  Der Aufruf einer MULTI-FOR-Schleife hat dabei die Form:

  IF MULTIFOR(i,anz,start,ende,diff) THEN REPEAT
    ...
    [ Schleifenkörper ... ]
    ...
  UNTIL ENDMULTIFOR(i);

  Es dürfen durchaus auch mehrere MULTI-FOR-Schleifen mit verschiedenen
  Laufvariablen ineinander verschachtelt sein. Wird eine Laufvariable im
  Schleifenkörper manipuliert, so wird bei ENDMULTIFOR mit den veränderten
  Werten gearbeitet. Es muß anz < max_multi gelten.
  Weiter programmtechnische Erklärungen sind in den Proceduren zu finden.
}

{-------------------------------------------------------------------------}

INTERFACE
```

```
USES TOOLS1;

CONST max_multi=10; { Maximale Anzahl von ineinandergeschachtelten
                      FOR-Schleifen }

TYPE for_type = ARRAY [-2..max_multi] OF integer;
                { Typ der Laufvariable; in den Feldern [-2..0] werden
                  die Grenzen und der aktuelle Zustand der Variable
                  gespeichert: i[0]  = Schleifenanzahl anz
                               i[-1] = Schleifenende ende
                               i[-2] = Differenz zwischen zwei Schleifen}

VAR min_index : integer;
    { Gibt nach jedem ENDMULTIFOR den niedersten geänderten Index an }

{-------------------------------------------------------------------------}

FUNCTION MULTIFOR(VAR i:for_type; anz,start,ende,diff:integer):BOOLEAN;
{ Erläuterungen siehe oben ... }
{ Das Ergebnis ist FALSE, falls kein Durchlauf möglich ist. }

FUNCTION ENDMULTIFOR(VAR i:for_type):BOOLEAN;
{ Erläuterungen siehe oben ... }
{ Das Ergebnis ist TRUE, falls alle Schleifen beendet sind. }

FUNCTION ANZMULTIFOR(m,n,d:integer):comp;
{ Berechnet die Anzahl von Durchläufen von m FOR-Schleifen der Gestalt:
    FOR i(1):=1 TO n DO
      FOR i(2):=i(1)+d TO n DO
        ..
          FOR i(m) := i(m-1)+d TO n DO BEGIN
            ! Hier wird die Durchlaufsanzahl gezählt !
          END;
}

{-------------------------------------------------------------------------}

IMPLEMENTATION

{-------------------------------------------------------------------------}

FUNCTION MULTIFOR(VAR i:for_type; anz,start,ende,diff:integer):BOOLEAN;

VAR l:integer;

BEGIN
  {Initialisierung der Laufvariable}
  i[0]:=anz;
  i[-1]:=ende;
  i[-2]:=diff;
  i[1]:=start;
  FOR l:=2 TO anz DO
    i[l]:=i[PRED(l)]+diff;
  min_index:=1;
  multifor:=(i[anz]<=ende);
END; {MULTIFOR}
```

```
{-------------------------------------------------------------------------}

FUNCTION ENDMULTIFOR(VAR i:for_type):BOOLEAN;

VAR anzahl,
    index,
    ende,
    diff,l :integer;

BEGIN
  anzahl:=i[0];
  index:=anzahl;
  ende:=i[-1];
  diff:=i[-2];
  {Ermitteln des kleinsten zu ändernden Index }
  WHILE (index>0) AND (i[index] >= ende) DO BEGIN
    DEC(index);
    DEC(ende,diff);
  END;
  min_index:=index;
  IF index=0 THEN BEGIN
    endmultifor:=TRUE;
    EXIT; {Ende der FOR-Schleifen !}
  END;
  { Anpassen der Indizes }
  INC(i[index]);
  FOR l:=SUCC(index) TO anzahl DO
    i[l]:=i[PRED(l)]+diff;
  endmultifor:=FALSE;
END; {ENDMULTIFOR}

{-------------------------------------------------------------------------}

FUNCTION ANZMULTIFOR(m,n,d:integer):comp;
{ Berechnet die Anzahl von Durchläufen von m FOR-Schleifen der Gestalt:
    FOR i(1):=1 TO n DO
      FOR i(2):=i(1)+d TO n DO
        ..
          FOR i(m) := i(m-1)+d TO n DO
            ! Hier wird die Durchlaufsanzahl gezählt !
}
BEGIN
  anzmultifor:=binom(n-(d-1)*(m-1),m);
END; {ANZMULTIFOR}

{-------------------------------------------------------------------------}

BEGIN {UNIT}

END. {UNIT}
```

6. Programmergebnisse:

Dimension 2

```
##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:08:20

Vollständige Liste der schrägen Hyperebenen-Typen im Nullpunkt für Q2:

( -1  1);       1 mal;   Typ: (1,2,1)


##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:08:20

Auswertung der schrägen Hyperebenen für Q2:

e0(2) =      2.000


Schräge Hyperebenen in Q2:

#Pkt      #HE      #Typ

   2          2
              2 mal Typ ( 1, 2, 1)  [ 1 verschiedene OV ]

Größtes aufgetretenes Vektorelement: 1  (kmax=1)

   - - - -   E N D E   - - - -
```

Dimension 3

```
##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:08:39

Vollständige Liste der schrägen Hyperebenen-Typen im Nullpunkt für Q3:

( -1  1  1);       3 mal;   Typ: (1,3,4)


##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:08:39

Auswertung der schrägen Hyperebenen für Q3:

e0(3) =      8.000


Schräge Hyperebenen in Q3:

#Pkt      #HE      #Typ

   3          8
              8 mal Typ ( 1, 3, 4)  [ 1 verschiedene OV ]

Größtes aufgetretenes Vektorelement: 1  (kmax=1)

   - - - -   E N D E   - - - -
```

Dimension 4

```
##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:09:02

Vollständige Liste der schrägen Hyperebenen-Typen im Nullpunkt für Q4:

( -1  1  1  1);       4 mal;   Typ: (1,4,11)
( -2  1  1  1);       4 mal;   Typ: (4,4,8)
( -1 -1  1  1);       3 mal;   Typ: (5,6,5)
( -1 -1  1  2);      12 mal;   Typ: (4,4,8)


##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:09:02

Auswertung der schrägen Hyperebenen für Q4:

e0(4) =     88.000


Schräge Hyperebenen in Q4:

#Pkt      #HE      #Typ

   4         80
             16 mal Typ ( 1, 4,11)  [ 1 verschiedene OV ]
             64 mal Typ ( 4, 4, 8)  [ 2 verschiedene OV ]

   6          8
              8 mal Typ ( 5, 6, 5)  [ 1 verschiedene OV ]

Größtes aufgetretenes Vektorelement: 2  (kmax=2)

   - - - -   E N D E   - - - -
```

Dimension 5

```
##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:09:20

Vollständige Liste der schrägen Hyperebenen-Typen im Nullpunkt für Q5:

( -1  1  1  1  1);       5 mal;   Typ: (1,5,26)
( -2  1  1  1  1);       5 mal;   Typ: (5,7,20)
( -2  1  1  1  2);      20 mal;   Typ: (4,5,23)
( -3  1  1  1  1);       5 mal;   Typ: (11,5,16)
( -3  1  1  1  2);      20 mal;   Typ: (8,5,19)
( -1 -1  1  1  1);      10 mal;   Typ: (6,10,16)
( -1 -1  1  1  2);      30 mal;   Typ: (5,7,20)
( -1 -1  1  2  2);      30 mal;   Typ: (4,5,23)
( -1 -2  1  1  1);      20 mal;   Typ: (12,8,12)
( -1 -2  1  1  2);      60 mal;   Typ: (9,7,16)
( -1 -2  1  1  3);      60 mal;   Typ: (8,5,19)
( -1 -3  1  1  1);      20 mal;   Typ: (11,5,16)
( -1 -3  1  2  2);      60 mal;   Typ: (11,5,16)
( -2 -2  1  1  1);      10 mal;   Typ: (9,7,16)
( -2 -2  1  1  3);      30 mal;   Typ: (11,5,16)
( -2 -3  1  1  1);      20 mal;   Typ: (8,5,19)
( -2 -3  1  1  2);      60 mal;   Typ: (11,5,16)

##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:09:20

Auswertung der schrägen Hyperebenen für Q5:

e0(5) =   2704.000

Schräge Hyperebenen in Q5:

#Pkt      #HE      #Typ

   5       2112
             32 mal Typ ( 1, 5,26)  [ 1 verschiedene OV ]
            320 mal Typ ( 4, 5,23)  [ 2 verschiedene OV ]
            640 mal Typ ( 8, 5,19)  [ 3 verschiedene OV ]
           1120 mal Typ (11, 5,16)  [ 5 verschiedene OV ]

   7        480
            160 mal Typ ( 5, 7,20)  [ 2 verschiedene OV ]
            320 mal Typ ( 9, 7,16)  [ 2 verschiedene OV ]

   8         80
             80 mal Typ (12, 8,12)  [ 1 verschiedene OV ]

  10         32
             32 mal Typ ( 6,10,16)  [ 1 verschiedene OV ]

Größtes aufgetretenes Vektorelement: 3  (kmax=3)

   - - - -   E N D E   - - - -
```

Dimension 6

```
##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:10:28

Vollständige Liste der schrägen Hyperebenen-Typen im Nullpunkt für Q6:

( -1  1  1  1  1  1);       6 mal;   Typ: (1,6,57)
( -2  1  1  1  1  1);       6 mal;   Typ: (6,11,47)
( -2  1  1  1  1  2);      30 mal;   Typ: (5,8,51)
( -2  1  1  1  2  2);      60 mal;   Typ: (4,6,54)
( -3  1  1  1  1  1);       6 mal;   Typ: (16,11,37)
( -3  1  1  1  1  2);      30 mal;   Typ: (12,9,43)
( -3  1  1  1  1  3);      30 mal;   Typ: (11,6,47)
( -3  1  1  1  2  2);      60 mal;   Typ: (9,8,47)
( -3  1  1  1  2  3);     120 mal;   Typ: (8,6,50)
( -4  1  1  1  1  1);       6 mal;   Typ: (26,6,32)
( -4  1  1  1  1  2);      30 mal;   Typ: (20,8,36)
( -4  1  1  1  1  3);      30 mal;   Typ: (16,6,42)
( -4  1  1  1  2  2);      60 mal;   Typ: (16,8,40)
( -4  1  1  2  2  3);     180 mal;   Typ: (11,6,47)
( -5  1  1  1  2  2);      60 mal;   Typ: (23,6,35)
( -5  1  1  1  2  3);     120 mal;   Typ: (19,6,39)
( -5  1  1  2  2  3);     180 mal;   Typ: (16,6,42)
( -1 -1  1  1  1  1);      15 mal;   Typ: (7,15,42)
( -1 -1  1  1  1  2);      60 mal;   Typ: (6,11,47)
( -1 -1  1  1  2  2);      90 mal;   Typ: (5,8,51)
( -1 -1  1  2  2  2);      60 mal;   Typ: (4,6,54)
( -1 -2  1  1  1  1);      30 mal;   Typ: (17,15,32)
( -1 -2  1  1  1  2);     120 mal;   Typ: (13,12,39)
( -1 -2  1  1  1  3);     120 mal;   Typ: (12,9,43)
( -1 -2  1  1  2  2);     180 mal;   Typ: (10,10,44)
( -1 -2  1  1  2  3);     360 mal;   Typ: (9,8,47)
( -1 -2  1  1  3  3);     180 mal;   Typ: (8,6,50)
( -1 -3  1  1  1  1);      30 mal;   Typ: (27,10,27)
( -1 -3  1  1  1  2);     120 mal;   Typ: (21,11,32)
( -1 -3  1  1  1  3);     120 mal;   Typ: (17,9,38)
( -1 -3  1  1  1  4);     120 mal;   Typ: (16,6,42)
( -1 -3  1  1  2  2);     180 mal;   Typ: (17,10,37)
( -1 -3  1  2  2  2);     120 mal;   Typ: (14,8,42)
( -1 -3  1  2  2  3);     360 mal;   Typ: (12,7,45)
( -1 -3  1  2  2  4);     360 mal;   Typ: (11,6,47)
( -1 -4  1  1  1  1);      30 mal;   Typ: (26,6,32)
( -1 -4  1  1  2  2);     180 mal;   Typ: (24,8,32)
( -1 -4  1  1  2  3);     360 mal;   Typ: (20,8,36)
( -1 -4  1  2  2  3);     360 mal;   Typ: (17,7,40)
( -1 -5  1  1  3  3);     180 mal;   Typ: (23,6,35)
( -1 -5  1  2  2  2);     120 mal;   Typ: (26,6,32)
( -1 -5  1  2  2  4);     360 mal;   Typ: (19,6,39)
( -2 -2  1  1  1  1);      15 mal;   Typ: (25,14,25)
( -2 -2  1  1  1  2);      60 mal;   Typ: (20,12,32)
( -2 -2  1  1  1  3);      60 mal;   Typ: (17,10,37)
( -2 -2  1  1  1  4);      60 mal;   Typ: (16,8,40)
( -2 -2  1  1  2  3);     180 mal;   Typ: (14,8,42)
```

```
( -2 -2  1  1  3  3);      90 mal;   Typ: (12,7,45)
( -2 -2  1  1  3  4);     180 mal;   Typ: (11,6,47)
( -2 -3  1  1  1  1);      30 mal;   Typ: (21,11,32)
( -2 -3  1  1  1  2);     120 mal;   Typ: (27,10,27)
( -2 -3  1  1  1  3);     120 mal;   Typ: (23,9,32)
( -2 -3  1  1  1  4);     120 mal;   Typ: (20,8,36)
( -2 -3  1  1  1  5);     120 mal;   Typ: (19,6,39)
( -2 -3  1  1  2  2);     180 mal;   Typ: (22,10,32)
( -2 -3  1  1  2  3);     360 mal;   Typ: (19,8,37)
( -2 -3  1  1  2  4);     360 mal;   Typ: (17,7,40)
( -2 -3  1  1  2  5);     360 mal;   Typ: (16,6,42)
( -2 -4  1  1  1  1);      30 mal;   Typ: (20,8,36)
( -2 -4  1  1  1  2);     120 mal;   Typ: (24,8,32)
( -2 -4  1  1  2  3);     360 mal;   Typ: (24,8,32)
( -2 -4  1  1  2  5);     360 mal;   Typ: (19,6,39)
( -2 -4  1  1  3  3);     180 mal;   Typ: (21,7,36)
( -2 -5  1  1  1  2);     120 mal;   Typ: (23,6,35)
( -2 -5  1  1  2  2);     180 mal;   Typ: (26,6,32)
( -2 -5  1  1  3  3);     180 mal;   Typ: (26,6,32)
( -2 -5  1  1  3  4);     360 mal;   Typ: (23,6,35)
( -3 -3  1  1  1  1);      15 mal;   Typ: (17,9,38)
( -3 -3  1  1  1  2);      60 mal;   Typ: (23,9,32)
( -3 -3  1  1  1  4);      60 mal;   Typ: (26,6,32)
( -3 -3  1  1  1  5);      60 mal;   Typ: (23,6,35)
( -3 -3  1  1  2  4);     180 mal;   Typ: (21,7,36)
( -3 -3  1  2  2  2);      60 mal;   Typ: (24,8,32)
( -3 -4  1  1  1  1);      30 mal;   Typ: (16,6,42)
( -3 -4  1  1  1  2);     120 mal;   Typ: (20,8,36)
( -3 -4  1  1  1  3);     120 mal;   Typ: (26,6,32)
( -3 -4  1  1  2  2);     180 mal;   Typ: (24,8,32)
( -3 -4  1  1  2  5);     360 mal;   Typ: (23,6,35)
( -3 -4  1  2  2  3);     360 mal;   Typ: (25,7,32)
( -3 -5  1  1  1  2);     120 mal;   Typ: (19,6,39)
( -3 -5  1  1  1  3);     120 mal;   Typ: (23,6,35)
( -3 -5  1  1  2  3);     360 mal;   Typ: (26,6,32)
( -3 -5  1  2  2  4);     360 mal;   Typ: (26,6,32)
( -4 -5  1  1  2  2);     180 mal;   Typ: (19,6,39)
( -4 -5  1  1  2  3);     360 mal;   Typ: (23,6,35)
( -4 -5  1  2  2  3);     360 mal;   Typ: (26,6,32)
( -1 -1 -1  1  1  1);      10 mal;   Typ: (22,20,22)
( -1 -1 -1  1  1  2);      60 mal;   Typ: (17,15,32)
( -1 -1 -1  1  1  3);      60 mal;   Typ: (16,11,37)
( -1 -1 -1  1  2  2);      60 mal;   Typ: (13,12,39)
( -1 -1 -1  1  2  3);     120 mal;   Typ: (12,9,43)
( -1 -1 -1  1  3  3);      60 mal;   Typ: (11,6,47)
( -1 -1 -1  2  2  2);      20 mal;   Typ: (10,10,44)
( -1 -1 -1  2  2  3);      60 mal;   Typ: (9,8,47)
( -1 -1 -1  2  3  3);      60 mal;   Typ: (8,6,50)
( -1 -1 -2  1  1  2);      90 mal;   Typ: (25,14,25)
( -1 -1 -2  1  1  3);     180 mal;   Typ: (21,11,32)
( -1 -1 -2  1  1  4);     180 mal;   Typ: (20,8,36)
( -1 -1 -2  1  2  2);     180 mal;   Typ: (20,12,32)
( -1 -1 -2  1  2  3);     360 mal;   Typ: (17,10,37)
( -1 -1 -2  1  2  4);     360 mal;   Typ: (16,8,40)
( -1 -1 -2  2  2  3);     180 mal;   Typ: (14,8,42)
( -1 -1 -2  2  3  3);     180 mal;   Typ: (12,7,45)
( -1 -1 -2  2  3  4);     360 mal;   Typ: (11,6,47)
( -1 -1 -3  1  2  2);     180 mal;   Typ: (27,10,27)
```

```
( -1 -1 -3  1  2  3);     360 mal;   Typ: (23,9,32)
( -1 -1 -3  1  2  4);     360 mal;   Typ: (20,8,36)
( -1 -1 -3  1  2  5);     360 mal;   Typ: (19,6,39)
( -1 -1 -3  2  2  2);      60 mal;   Typ: (22,10,32)
( -1 -1 -3  2  2  3);     180 mal;   Typ: (19,8,37)
( -1 -1 -3  2  2  4);     180 mal;   Typ: (17,7,40)
( -1 -1 -3  2  2  5);     180 mal;   Typ: (16,6,42)
( -1 -1 -4  1  2  2);     180 mal;   Typ: (24,8,32)
( -1 -1 -4  1  3  3);     180 mal;   Typ: (26,6,32)
( -1 -1 -4  2  2  3);     180 mal;   Typ: (24,8,32)
( -1 -1 -4  2  2  5);     180 mal;   Typ: (19,6,39)
( -1 -1 -4  2  3  3);     180 mal;   Typ: (21,7,36)
( -1 -1 -5  1  2  2);     180 mal;   Typ: (23,6,35)
( -1 -1 -5  2  2  2);      60 mal;   Typ: (26,6,32)
( -1 -1 -5  2  3  3);     180 mal;   Typ: (26,6,32)
( -1 -1 -5  2  3  4);     360 mal;   Typ: (23,6,35)
( -1 -2 -2  1  2  3);     360 mal;   Typ: (22,10,32)
( -1 -2 -2  1  3  3);     180 mal;   Typ: (19,8,37)
( -1 -2 -2  1  3  4);     360 mal;   Typ: (17,7,40)
( -1 -2 -2  1  3  5);     360 mal;   Typ: (16,6,42)
( -1 -2 -3  1  2  4);     720 mal;   Typ: (24,8,32)
( -1 -2 -3  1  3  4);     720 mal;   Typ: (21,7,36)
( -1 -2 -3  2  2  3);     360 mal;   Typ: (24,8,32)
( -1 -2 -4  1  3  5);     720 mal;   Typ: (23,6,35)
( -1 -2 -4  2  3  3);     360 mal;   Typ: (25,7,32)
( -1 -2 -5  1  3  3);     360 mal;   Typ: (26,6,32)
( -1 -2 -5  2  3  4);     720 mal;   Typ: (26,6,32)
( -1 -3 -3  2  2  2);      60 mal;   Typ: (24,8,32)
( -1 -3 -3  2  2  4);     180 mal;   Typ: (25,7,32)
( -1 -3 -4  2  2  3);     360 mal;   Typ: (25,7,32)
( -1 -3 -4  2  2  5);     360 mal;   Typ: (26,6,32)
( -1 -3 -5  2  2  4);     360 mal;   Typ: (26,6,32)


##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:10:29

Auswertung der schrägen Hyperebenen für Q6:

e0(6) = 234688.000


Schräge Hyperebenen in Q6:

#Pkt      #HE      #Typ

   6     123648
            64 mal Typ ( 1, 6,57)  [ 1 verschiedene OV ]
          1280 mal Typ ( 4, 6,54)  [ 2 verschiedene OV ]
          3840 mal Typ ( 8, 6,50)  [ 3 verschiedene OV ]
         12480 mal Typ (11, 6,47)  [ 6 verschiedene OV ]
         13440 mal Typ (16, 6,42)  [ 7 verschiedene OV ]
         19200 mal Typ (19, 6,39)  [ 8 verschiedene OV ]
         30720 mal Typ (23, 6,35)  [ 11 verschiedene OV ]
         42624 mal Typ (26, 6,32)  [ 17 verschiedene OV ]

   7      40320
```

```
          5760 mal Typ (12, 7,45)  [ 3 verschiedene OV ]
         11520 mal Typ (17, 7,40)  [ 4 verschiedene OV ]
         11520 mal Typ (21, 7,36)  [ 4 verschiedene OV ]
         11520 mal Typ (25, 7,32)  [ 4 verschiedene OV ]

  8      47040
           960 mal Typ ( 5, 8,51)  [ 2 verschiedene OV ]
          3840 mal Typ ( 9, 8,47)  [ 3 verschiedene OV ]
          3840 mal Typ (14, 8,42)  [ 3 verschiedene OV ]
          3840 mal Typ (16, 8,40)  [ 3 verschiedene OV ]
          5760 mal Typ (19, 8,37)  [ 3 verschiedene OV ]
          9600 mal Typ (20, 8,36)  [ 7 verschiedene OV ]
         19200 mal Typ (24, 8,32)  [ 10 verschiedene OV ]

  9       6720
          1920 mal Typ (12, 9,43)  [ 3 verschiedene OV ]
           960 mal Typ (17, 9,38)  [ 2 verschiedene OV ]
          3840 mal Typ (23, 9,32)  [ 3 verschiedene OV ]

 10      11072
          1280 mal Typ (10,10,44)  [ 2 verschiedene OV ]
          3840 mal Typ (17,10,37)  [ 3 verschiedene OV ]
          3840 mal Typ (22,10,32)  [ 3 verschiedene OV ]
          2112 mal Typ (27,10,27)  [ 3 verschiedene OV ]

 11       2688
           384 mal Typ ( 6,11,47)  [ 2 verschiedene OV ]
           384 mal Typ (16,11,37)  [ 2 verschiedene OV ]
          1920 mal Typ (21,11,32)  [ 3 verschiedene OV ]

 12       2240
           960 mal Typ (13,12,39)  [ 2 verschiedene OV ]
          1280 mal Typ (20,12,32)  [ 2 verschiedene OV ]

 14        480
           480 mal Typ (25,14,25)  [ 2 verschiedene OV ]

 15        448
            64 mal Typ ( 7,15,42)  [ 1 verschiedene OV ]
           384 mal Typ (17,15,32)  [ 2 verschiedene OV ]

 20         32
            32 mal Typ (22,20,22)  [ 1 verschiedene OV ]

Größtes aufgetretenes Vektorelement: 5  (kmax=6)

   - - - -   E N D E   - - - -
```

Dimension 7

(Ab Dimension 7 wird die vollständige Liste der Orthogonalvektoren nicht mehr angegeben, da z.B. alleine die Datei mit allen Ergebnissen aus Q_8 mehr als 8 MB groß ist.)

```
##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-01    11:12:30

Auswertung der schrägen Hyperebenen für Q7:

e0(7) = 69640192.000


Schräge Hyperebenen in Q7:

#Pkt      #HE      #Typ

   7   23963648
         128 mal Typ ( 1, 7,120)  [ 1 verschiedene OV ]
        4480 mal Typ ( 4, 7,117)  [ 2 verschiedene OV ]
       17920 mal Typ ( 8, 7,113)  [ 3 verschiedene OV ]
       85120 mal Typ (11, 7,110)  [ 6 verschiedene OV ]
       94080 mal Typ (16, 7,105)  [ 7 verschiedene OV ]
      215040 mal Typ (19, 7,102)  [ 9 verschiedene OV ]
       17920 mal Typ (20, 7,101)  [ 3 verschiedene OV ]
      403200 mal Typ (23, 7,98)  [ 13 verschiedene OV ]
      840448 mal Typ (26, 7,95)  [ 32 verschiedene OV ]
      322560 mal Typ (27, 7,94)  [ 6 verschiedene OV ]
      806400 mal Typ (30, 7,91)  [ 12 verschiedene OV ]
      758016 mal Typ (32, 7,89)  [ 27 verschiedene OV ]
      322560 mal Typ (33, 7,88)  [ 6 verschiedene OV ]
     1451520 mal Typ (35, 7,86)  [ 35 verschiedene OV ]
      215040 mal Typ (36, 7,85)  [ 9 verschiedene OV ]
     1532160 mal Typ (39, 7,82)  [ 40 verschiedene OV ]
      698880 mal Typ (41, 7,80)  [ 16 verschiedene OV ]
     1066240 mal Typ (42, 7,79)  [ 33 verschiedene OV ]
     1142400 mal Typ (43, 7,78)  [ 21 verschiedene OV ]
      241920 mal Typ (45, 7,76)  [ 9 verschiedene OV ]
      967680 mal Typ (46, 7,75)  [ 13 verschiedene OV ]
     1478400 mal Typ (47, 7,74)  [ 43 verschiedene OV ]
      179200 mal Typ (48, 7,73)  [ 8 verschiedene OV ]
      645120 mal Typ (49, 7,72)  [ 12 verschiedene OV ]
     1344000 mal Typ (50, 7,71)  [ 39 verschiedene OV ]
      698880 mal Typ (51, 7,70)  [ 11 verschiedene OV ]
      161280 mal Typ (52, 7,69)  [ 5 verschiedene OV ]
      645120 mal Typ (53, 7,68)  [ 7 verschiedene OV ]
     3046400 mal Typ (54, 7,67)  [ 70 verschiedene OV ]
     4561536 mal Typ (57, 7,64)  [ 111 verschiedene OV ]

   8   18278400
        80640 mal Typ (12, 8,108)  [ 4 verschiedene OV ]
       161280 mal Typ (17, 8,103)  [ 5 verschiedene OV ]
        80640 mal Typ (20, 8,100)  [ 4 verschiedene OV ]
       161280 mal Typ (21, 8,99)  [ 5 verschiedene OV ]
```

```
      322560 mal Typ (24, 8,96)  [ 6 verschiedene OV ]
      161280 mal Typ (25, 8,95)  [ 5 verschiedene OV ]
      322560 mal Typ (26, 8,94)  [ 10 verschiedene OV ]
      188160 mal Typ (27, 8,93)  [ 8 verschiedene OV ]
      322560 mal Typ (28, 8,92)  [ 6 verschiedene OV ]
      403200 mal Typ (29, 8,91)  [ 10 verschiedene OV ]
      161280 mal Typ (32, 8,88)  [ 5 verschiedene OV ]
      456960 mal Typ (33, 8,87)  [ 18 verschiedene OV ]
      322560 mal Typ (34, 8,86)  [ 6 verschiedene OV ]
      483840 mal Typ (35, 8,85)  [ 12 verschiedene OV ]
      483840 mal Typ (36, 8,84)  [ 11 verschiedene OV ]
      161280 mal Typ (37, 8,83)  [ 5 verschiedene OV ]
      483840 mal Typ (38, 8,82)  [ 12 verschiedene OV ]
      322560 mal Typ (39, 8,81)  [ 6 verschiedene OV ]
      967680 mal Typ (40, 8,80)  [ 22 verschiedene OV ]
      161280 mal Typ (42, 8,78)  [ 5 verschiedene OV ]
      806400 mal Typ (43, 8,77)  [ 25 verschiedene OV ]
      967680 mal Typ (44, 8,76)  [ 14 verschiedene OV ]
      483840 mal Typ (45, 8,75)  [ 14 verschiedene OV ]
      322560 mal Typ (46, 8,74)  [ 10 verschiedene OV ]
      806400 mal Typ (47, 8,73)  [ 18 verschiedene OV ]
     1612800 mal Typ (48, 8,72)  [ 35 verschiedene OV ]
      322560 mal Typ (49, 8,71)  [ 6 verschiedene OV ]
      564480 mal Typ (51, 8,69)  [ 15 verschiedene OV ]
     2741760 mal Typ (52, 8,68)  [ 60 verschiedene OV ]
     3440640 mal Typ (56, 8,64)  [ 75 verschiedene OV ]

9   15554560
        4480 mal Typ ( 5, 9,114)  [ 2 verschiedene OV ]
       26880 mal Typ ( 9, 9,110)  [ 3 verschiedene OV ]
       26880 mal Typ (13, 9,106)  [ 3 verschiedene OV ]
       53760 mal Typ (14, 9,105)  [ 4 verschiedene OV ]
       26880 mal Typ (16, 9,103)  [ 3 verschiedene OV ]
       80640 mal Typ (18, 9,101)  [ 4 verschiedene OV ]
       80640 mal Typ (19, 9,100)  [ 4 verschiedene OV ]
      120960 mal Typ (20, 9,99)  [ 8 verschiedene OV ]
      241920 mal Typ (22, 9,97)  [ 9 verschiedene OV ]
      376320 mal Typ (24, 9,95)  [ 20 verschiedene OV ]
       53760 mal Typ (25, 9,94)  [ 4 verschiedene OV ]
      215040 mal Typ (27, 9,92)  [ 9 verschiedene OV ]
      161280 mal Typ (28, 9,91)  [ 5 verschiedene OV ]
      161280 mal Typ (29, 9,90)  [ 5 verschiedene OV ]
      206080 mal Typ (30, 9,89)  [ 11 verschiedene OV ]
       26880 mal Typ (31, 9,88)  [ 3 verschiedene OV ]
      268800 mal Typ (32, 9,87)  [ 13 verschiedene OV ]
      286720 mal Typ (33, 9,86)  [ 17 verschiedene OV ]
      161280 mal Typ (34, 9,85)  [ 5 verschiedene OV ]
      322560 mal Typ (35, 9,84)  [ 10 verschiedene OV ]
      618240 mal Typ (36, 9,83)  [ 28 verschiedene OV ]
      322560 mal Typ (37, 9,82)  [ 13 verschiedene OV ]
       53760 mal Typ (38, 9,81)  [ 4 verschiedene OV ]
      403200 mal Typ (39, 9,80)  [ 18 verschiedene OV ]
      806400 mal Typ (40, 9,79)  [ 26 verschiedene OV ]
      860160 mal Typ (41, 9,78)  [ 25 verschiedene OV ]
      241920 mal Typ (42, 9,77)  [ 13 verschiedene OV ]
      698880 mal Typ (44, 9,75)  [ 28 verschiedene OV ]
      806400 mal Typ (45, 9,74)  [ 27 verschiedene OV ]
      806400 mal Typ (46, 9,73)  [ 21 verschiedene OV ]
```

```
       376320 mal Typ (47, 9,72)  [ 18 verschiedene OV ]
       295680 mal Typ (49, 9,70)  [ 13 verschiedene OV ]
      1666560 mal Typ (50, 9,69)  [ 52 verschiedene OV ]
      1155840 mal Typ (51, 9,68)  [ 42 verschiedene OV ]
      3539200 mal Typ (55, 9,64)  [ 124 verschiedene OV ]

10    3655680
        13440 mal Typ (12,10,106)  [ 3 verschiedene OV ]
        26880 mal Typ (15,10,103)  [ 3 verschiedene OV ]
        13440 mal Typ (17,10,101)  [ 3 verschiedene OV ]
        80640 mal Typ (20,10,98)  [ 4 verschiedene OV ]
        53760 mal Typ (23,10,95)  [ 4 verschiedene OV ]
       161280 mal Typ (25,10,93)  [ 5 verschiedene OV ]
        53760 mal Typ (29,10,89)  [ 5 verschiedene OV ]
       161280 mal Typ (31,10,87)  [ 6 verschiedene OV ]
        53760 mal Typ (32,10,86)  [ 4 verschiedene OV ]
       161280 mal Typ (33,10,85)  [ 5 verschiedene OV ]
        26880 mal Typ (35,10,83)  [ 3 verschiedene OV ]
        80640 mal Typ (37,10,81)  [ 4 verschiedene OV ]
       255360 mal Typ (38,10,80)  [ 12 verschiedene OV ]
       537600 mal Typ (43,10,75)  [ 29 verschiedene OV ]
       161280 mal Typ (45,10,73)  [ 5 verschiedene OV ]
       309120 mal Typ (48,10,70)  [ 17 verschiedene OV ]
       403200 mal Typ (49,10,69)  [ 11 verschiedene OV ]
        80640 mal Typ (50,10,68)  [ 4 verschiedene OV ]
      1021440 mal Typ (54,10,64)  [ 40 verschiedene OV ]

11    4465664
        17920 mal Typ (10,11,107)  [ 3 verschiedene OV ]
        53760 mal Typ (17,11,100)  [ 4 verschiedene OV ]
        53760 mal Typ (21,11,96)  [ 4 verschiedene OV ]
        80640 mal Typ (22,11,95)  [ 7 verschiedene OV ]
        26880 mal Typ (25,11,92)  [ 3 verschiedene OV ]
       107520 mal Typ (26,11,91)  [ 8 verschiedene OV ]
       139776 mal Typ (27,11,90)  [ 12 verschiedene OV ]
        80640 mal Typ (29,11,88)  [ 4 verschiedene OV ]
        53760 mal Typ (32,11,85)  [ 4 verschiedene OV ]
       110208 mal Typ (33,11,84)  [ 10 verschiedene OV ]
        53760 mal Typ (34,11,83)  [ 4 verschiedene OV ]
       215040 mal Typ (35,11,82)  [ 10 verschiedene OV ]
       161280 mal Typ (36,11,81)  [ 6 verschiedene OV ]
       161280 mal Typ (37,11,80)  [ 9 verschiedene OV ]
        26880 mal Typ (39,11,78)  [ 3 verschiedene OV ]
       161280 mal Typ (40,11,77)  [ 10 verschiedene OV ]
       376320 mal Typ (41,11,76)  [ 19 verschiedene OV ]
       241920 mal Typ (42,11,75)  [ 10 verschiedene OV ]
        71680 mal Typ (44,11,73)  [ 7 verschiedene OV ]
       107520 mal Typ (46,11,71)  [ 8 verschiedene OV ]
       376320 mal Typ (47,11,70)  [ 16 verschiedene OV ]
       645120 mal Typ (48,11,69)  [ 32 verschiedene OV ]
      1142400 mal Typ (53,11,64)  [ 58 verschiedene OV ]

12    1349376
         2688 mal Typ ( 6,12,110)  [ 2 verschiedene OV ]
         2688 mal Typ (16,12,100)  [ 2 verschiedene OV ]
        13440 mal Typ (17,12,99)  [ 3 verschiedene OV ]
        26880 mal Typ (21,12,95)  [ 4 verschiedene OV ]
        53760 mal Typ (23,12,93)  [ 4 verschiedene OV ]
```

```
         53760 mal Typ (24,12,92)  [ 4 verschiedene OV ]
         26880 mal Typ (32,12,84)  [ 4 verschiedene OV ]
        107520 mal Typ (33,12,83)  [ 8 verschiedene OV ]
         32256 mal Typ (37,12,79)  [ 7 verschiedene OV ]
         67200 mal Typ (39,12,77)  [ 7 verschiedene OV ]
        107520 mal Typ (40,12,76)  [ 6 verschiedene OV ]
        107520 mal Typ (45,12,71)  [ 9 verschiedene OV ]
        161280 mal Typ (46,12,70)  [ 6 verschiedene OV ]
         32256 mal Typ (47,12,69)  [ 7 verschiedene OV ]
        430080 mal Typ (52,12,64)  [ 28 verschiedene OV ]
        123648 mal Typ (58,12,58)  [ 12 verschiedene OV ]

13     1397760
          4480 mal Typ (11,13,104)  [ 2 verschiedene OV ]
         13440 mal Typ (13,13,102)  [ 3 verschiedene OV ]
         31360 mal Typ (18,13,97)  [ 5 verschiedene OV ]
         17920 mal Typ (20,13,95)  [ 3 verschiedene OV ]
         17920 mal Typ (27,13,88)  [ 3 verschiedene OV ]
         53760 mal Typ (28,13,87)  [ 5 verschiedene OV ]
         17920 mal Typ (29,13,86)  [ 3 verschiedene OV ]
         80640 mal Typ (30,13,85)  [ 5 verschiedene OV ]
         17920 mal Typ (32,13,83)  [ 3 verschiedene OV ]
         13440 mal Typ (34,13,81)  [ 3 verschiedene OV ]
         80640 mal Typ (36,13,79)  [ 8 verschiedene OV ]
         53760 mal Typ (37,13,78)  [ 5 verschiedene OV ]
        134400 mal Typ (38,13,77)  [ 8 verschiedene OV ]
         26880 mal Typ (39,13,76)  [ 6 verschiedene OV ]
         35840 mal Typ (42,13,73)  [ 6 verschiedene OV ]
        188160 mal Typ (44,13,71)  [ 17 verschiedene OV ]
        152320 mal Typ (45,13,70)  [ 12 verschiedene OV ]
        456960 mal Typ (51,13,64)  [ 38 verschiedene OV ]

14      255360
         26880 mal Typ (25,14,89)  [ 3 verschiedene OV ]
         53760 mal Typ (35,14,79)  [ 5 verschiedene OV ]
         80640 mal Typ (43,14,71)  [ 5 verschiedene OV ]
         53760 mal Typ (50,14,64)  [ 4 verschiedene OV ]
         40320 mal Typ (57,14,57)  [ 3 verschiedene OV ]

15      383488
         17920 mal Typ (21,15,92)  [ 3 verschiedene OV ]
         13440 mal Typ (22,15,91)  [ 3 verschiedene OV ]
         13440 mal Typ (25,15,88)  [ 4 verschiedene OV ]
          2688 mal Typ (28,15,85)  [ 2 verschiedene OV ]
         26880 mal Typ (31,15,82)  [ 4 verschiedene OV ]
         44800 mal Typ (33,15,80)  [ 7 verschiedene OV ]
         17920 mal Typ (40,15,73)  [ 3 verschiedene OV ]
         67200 mal Typ (41,15,72)  [ 8 verschiedene OV ]
         13440 mal Typ (43,15,70)  [ 3 verschiedene OV ]
        165760 mal Typ (49,15,64)  [ 18 verschiedene OV ]

16      133056
           896 mal Typ ( 7,16,105)  [ 2 verschiedene OV ]
          4480 mal Typ (14,16,98)  [ 2 verschiedene OV ]
          5376 mal Typ (17,16,95)  [ 3 verschiedene OV ]
          5376 mal Typ (32,16,80)  [ 3 verschiedene OV ]
          5376 mal Typ (38,16,74)  [ 3 verschiedene OV ]
         13440 mal Typ (40,16,72)  [ 3 verschiedene OV ]
```

```
          896 mal Typ (42,16,70)  [ 2 verschiedene OV ]
        50176 mal Typ (48,16,64)  [ 12 verschiedene OV ]
        47040 mal Typ (56,16,56)  [ 7 verschiedene OV ]

  17    53760
        13440 mal Typ (29,17,82)  [ 3 verschiedene OV ]
        26880 mal Typ (39,17,72)  [ 4 verschiedene OV ]
        13440 mal Typ (47,17,64)  [ 3 verschiedene OV ]

  18    96320
         4480 mal Typ (24,18,86)  [ 2 verschiedene OV ]
        13440 mal Typ (26,18,84)  [ 4 verschiedene OV ]
        17920 mal Typ (36,18,74)  [ 4 verschiedene OV ]
        13440 mal Typ (37,18,73)  [ 3 verschiedene OV ]
        40320 mal Typ (46,18,64)  [ 7 verschiedene OV ]
         6720 mal Typ (55,18,55)  [ 2 verschiedene OV ]

  20    30464
         2688 mal Typ (18,20,90)  [ 2 verschiedene OV ]
         5376 mal Typ (33,20,75)  [ 3 verschiedene OV ]
        13440 mal Typ (44,20,64)  [ 4 verschiedene OV ]
         8960 mal Typ (54,20,54)  [ 2 verschiedene OV ]

  21     4608
          128 mal Typ ( 8,21,99)  [ 1 verschiedene OV ]
          896 mal Typ (22,21,85)  [ 2 verschiedene OV ]
         3584 mal Typ (43,21,64)  [ 4 verschiedene OV ]

  22    11648
         4480 mal Typ (30,22,76)  [ 3 verschiedene OV ]
         4480 mal Typ (42,22,64)  [ 2 verschiedene OV ]
         2688 mal Typ (53,22,53)  [ 2 verschiedene OV ]

  24     2240
         2240 mal Typ (52,24,52)  [ 1 verschiedene OV ]

  26     3584
          896 mal Typ (23,26,79)  [ 2 verschiedene OV ]
         2688 mal Typ (38,26,64)  [ 3 verschiedene OV ]

  30      448
          448 mal Typ (49,30,49)  [ 1 verschiedene OV ]

  35      128
          128 mal Typ (29,35,64)  [ 1 verschiedene OV ]

Größtes aufgetretenes Vektorelement: 9  (kmax=14)

   - - - -   E N D E   - - - -
```

Dimension 8

```
##       HYPER ORTH V 2.0   (C)1992 by OSSISOFT     ##

1992-05-27    06:33:20

Auswertung der schrägen Hyperebenen für Q8:

e0(8) = 85682904704.000

Schräge Hyperebenen in Q8:

#Pkt      #HE      #Typ

   8  25806963456
          256 mal Typ ( 1, 8,247)  [ 1 verschiedene OV ]
        14336 mal Typ ( 4, 8,244)  [ 2 verschiedene OV ]
        71680 mal Typ ( 8, 8,240)  [ 3 verschiedene OV ]
       448000 mal Typ (11, 8,237)  [ 6 verschiedene OV ]
       501760 mal Typ (16, 8,232)  [ 7 verschiedene OV ]
      1576960 mal Typ (19, 8,229)  [ 9 verschiedene OV ]
       286720 mal Typ (20, 8,228)  [ 4 verschiedene OV ]
      3153920 mal Typ (23, 8,225)  [ 13 verschiedene OV ]
      8687616 mal Typ (26, 8,222)  [ 35 verschiedene OV ]
      5160960 mal Typ (27, 8,221)  [ 7 verschiedene OV ]
      2580480 mal Typ (28, 8,220)  [ 6 verschiedene OV ]
     12902400 mal Typ (30, 8,218)  [ 14 verschiedene OV ]
     10321920 mal Typ (31, 8,217)  [ 8 verschiedene OV ]
     17676288 mal Typ (32, 8,216)  [ 36 verschiedene OV ]
      7741440 mal Typ (33, 8,215)  [ 13 verschiedene OV ]
      3010560 mal Typ (34, 8,214)  [ 10 verschiedene OV ]
     19783680 mal Typ (35, 8,213)  [ 39 verschiedene OV ]
     18923520 mal Typ (36, 8,212)  [ 26 verschiedene OV ]
     13762560 mal Typ (37, 8,211)  [ 19 verschiedene OV ]
      2580480 mal Typ (38, 8,210)  [ 6 verschiedene OV ]
     22364160 mal Typ (39, 8,209)  [ 46 verschiedene OV ]
     15482880 mal Typ (40, 8,208)  [ 15 verschiedene OV ]
     21504000 mal Typ (41, 8,207)  [ 27 verschiedene OV ]
     37058560 mal Typ (42, 8,206)  [ 76 verschiedene OV ]
     25016320 mal Typ (43, 8,205)  [ 41 verschiedene OV ]
     30965760 mal Typ (44, 8,204)  [ 24 verschiedene OV ]
     29245440 mal Typ (45, 8,203)  [ 43 verschiedene OV ]
     37416960 mal Typ (46, 8,202)  [ 47 verschiedene OV ]
     42900480 mal Typ (47, 8,201)  [ 72 verschiedene OV ]
     46735360 mal Typ (48, 8,200)  [ 54 verschiedene OV ]
     20643840 mal Typ (49, 8,199)  [ 22 verschiedene OV ]
     64225280 mal Typ (50, 8,198)  [ 113 verschiedene OV ]
     61071360 mal Typ (51, 8,197)  [ 69 verschiedene OV ]
     38707200 mal Typ (52, 8,196)  [ 42 verschiedene OV ]
     75694080 mal Typ (53, 8,195)  [ 84 verschiedene OV ]
    129454080 mal Typ (54, 8,194)  [ 163 verschiedene OV ]
     64512000 mal Typ (55, 8,193)  [ 54 verschiedene OV ]
     86016000 mal Typ (56, 8,192)  [ 107 verschiedene OV ]
    182647808 mal Typ (57, 8,191)  [ 300 verschiedene OV ]
    173322240 mal Typ (58, 8,190)  [ 186 verschiedene OV ]
     91607040 mal Typ (59, 8,189)  [ 108 verschiedene OV ]
    130314240 mal Typ (60, 8,188)  [ 147 verschiedene OV ]
```

137625600 mal Typ (61, 8,187) [143 verschiedene OV]
144076800 mal Typ (62, 8,186) [162 verschiedene OV]
116121600 mal Typ (63, 8,185) [117 verschiedene OV]
191686656 mal Typ (64, 8,184) [241 verschiedene OV]
232243200 mal Typ (65, 8,183) [262 verschiedene OV]
165150720 mal Typ (66, 8,182) [157 verschiedene OV]
220989440 mal Typ (67, 8,181) [293 verschiedene OV]
108380160 mal Typ (68, 8,180) [87 verschiedene OV]
228587520 mal Typ (69, 8,179) [233 verschiedene OV]
184074240 mal Typ (70, 8,178) [193 verschiedene OV]
279838720 mal Typ (71, 8,177) [360 verschiedene OV]
206438400 mal Typ (72, 8,176) [195 verschiedene OV]
217047040 mal Typ (73, 8,175) [215 verschiedene OV]
411299840 mal Typ (74, 8,174) [516 verschiedene OV]
209018880 mal Typ (75, 8,173) [249 verschiedene OV]
145797120 mal Typ (76, 8,172) [145 verschiedene OV]
270090240 mal Typ (77, 8,171) [276 verschiedene OV]
266004480 mal Typ (78, 8,170) [268 verschiedene OV]
373811200 mal Typ (79, 8,169) [452 verschiedene OV]
295895040 mal Typ (80, 8,168) [303 verschiedene OV]
178913280 mal Typ (81, 8,167) [207 verschiedene OV]
471797760 mal Typ (82, 8,166) [515 verschiedene OV]
380047360 mal Typ (83, 8,165) [412 verschiedene OV]
246435840 mal Typ (84, 8,164) [208 verschiedene OV]
159559680 mal Typ (85, 8,163) [181 verschiedene OV]
429219840 mal Typ (86, 8,162) [536 verschiedene OV]
438466560 mal Typ (87, 8,161) [422 verschiedene OV]
387072000 mal Typ (88, 8,160) [394 verschiedene OV]
337842176 mal Typ (89, 8,159) [446 verschiedene OV]
415887360 mal Typ (90, 8,158) [446 verschiedene OV]
444917760 mal Typ (91, 8,157) [432 verschiedene OV]
442552320 mal Typ (92, 8,156) [409 verschiedene OV]
330301440 mal Typ (93, 8,155) [313 verschiedene OV]
556523520 mal Typ (94, 8,154) [562 verschiedene OV]
462493696 mal Typ (95, 8,153) [575 verschiedene OV]
469073920 mal Typ (96, 8,152) [462 verschiedene OV]
445132800 mal Typ (97, 8,151) [439 verschiedene OV]
488570880 mal Typ (98, 8,150) [579 verschiedene OV]
434667520 mal Typ (99, 8,149) [454 verschiedene OV]
458035200 mal Typ (100, 8,148) [415 verschiedene OV]
344350720 mal Typ (101, 8,147) [333 verschiedene OV]
560107520 mal Typ (102, 8,146) [658 verschiedene OV]
443412480 mal Typ (103, 8,145) [473 verschiedene OV]
555663360 mal Typ (104, 8,144) [524 verschiedene OV]
441548800 mal Typ (105, 8,143) [498 verschiedene OV]
607918080 mal Typ (106, 8,142) [623 verschiedene OV]
463626240 mal Typ (107, 8,141) [446 verschiedene OV]
630604800 mal Typ (108, 8,140) [615 verschiedene OV]
547061760 mal Typ (109, 8,139) [494 verschiedene OV]
476241920 mal Typ (110, 8,138) [617 verschiedene OV]
487137280 mal Typ (111, 8,137) [553 verschiedene OV]
694149120 mal Typ (112, 8,136) [664 verschiedene OV]
520110080 mal Typ (113, 8,135) [587 verschiedene OV]
378900480 mal Typ (114, 8,134) [375 verschiedene OV]
390942720 mal Typ (115, 8,133) [401 verschiedene OV]
1012838400 mal Typ (116, 8,132) [865 verschiedene OV]
903024640 mal Typ (117, 8,131) [1044 verschiedene OV]
2411546624 mal Typ (120, 8,128) [2686 verschiedene OV]

Für Hyperebenen mit 9 bis 22 Ecken werden die verschiedenen Typen hier nicht mehr angegeben, da diese Liste über 1000 Zeilen hätte.

```
 9  23857182720

10  16009297920

11  8515261440

12  5245578240

13  2524612608

14  1599346688

15  843171840

16  604692480

17  216857088

18  189865984

19  104832000

20   57523200

21   35546112

22   28966400

23   13289472
       71680 mal Typ (30,23,203)  [ 4 verschiedene OV ]
       71680 mal Typ (42,23,191)  [ 3 verschiedene OV ]
      215040 mal Typ (49,23,184)  [ 5 verschiedene OV ]
      430080 mal Typ (50,23,183)  [ 6 verschiedene OV ]
      587776 mal Typ (53,23,180)  [ 18 verschiedene OV ]
      286720 mal Typ (55,23,178)  [ 5 verschiedene OV ]
       71680 mal Typ (64,23,169)  [ 3 verschiedene OV ]
      215040 mal Typ (67,23,166)  [ 5 verschiedene OV ]
      430080 mal Typ (68,23,165)  [ 6 verschiedene OV ]
      430080 mal Typ (69,23,164)  [ 7 verschiedene OV ]
      215040 mal Typ (70,23,163)  [ 5 verschiedene OV ]
       71680 mal Typ (71,23,162)  [ 3 verschiedene OV ]
       43008 mal Typ (75,23,158)  [ 3 verschiedene OV ]
       71680 mal Typ (76,23,157)  [ 4 verschiedene OV ]
      358400 mal Typ (80,23,153)  [ 8 verschiedene OV ]
      860160 mal Typ (81,23,152)  [ 8 verschiedene OV ]
      430080 mal Typ (82,23,151)  [ 7 verschiedene OV ]
      501760 mal Typ (84,23,149)  [ 10 verschiedene OV ]
      215040 mal Typ (91,23,142)  [ 5 verschiedene OV ]
     2222080 mal Typ (93,23,140)  [ 28 verschiedene OV ]
     1290240 mal Typ (94,23,139)  [ 12 verschiedene OV ]
       43008 mal Typ (95,23,138)  [ 3 verschiedene OV ]
     4157440 mal Typ (105,23,128)  [ 50 verschiedene OV ]

24   10257408
      143360 mal Typ (40,24,192)  [ 4 verschiedene OV ]
```

```
 215040 mal Typ (45,24,187)  [ 5 verschiedene OV ]
 107520 mal Typ (61,24,171)  [ 4 verschiedene OV ]
 107520 mal Typ (62,24,170)  [ 3 verschiedene OV ]
 430080 mal Typ (63,24,169)  [ 6 verschiedene OV ]
 645120 mal Typ (65,24,167)  [ 11 verschiedene OV ]
 286720 mal Typ (66,24,166)  [ 4 verschiedene OV ]
 645120 mal Typ (77,24,155)  [ 11 verschiedene OV ]
 716800 mal Typ (78,24,154)  [ 12 verschiedene OV ]
 573440 mal Typ (80,24,152)  [ 8 verschiedene OV ]
 215040 mal Typ (90,24,142)  [ 4 verschiedene OV ]
1075200 mal Typ (91,24,141)  [ 13 verschiedene OV ]
 788480 mal Typ (92,24,140)  [ 14 verschiedene OV ]
 430080 mal Typ (93,24,139)  [ 5 verschiedene OV ]
2652160 mal Typ (104,24,128)  [ 35 verschiedene OV ]
1225728 mal Typ (116,24,116)  [ 19 verschiedene OV ]

25    6612480
  14336 mal Typ (19,25,212)  [ 2 verschiedene OV ]
  14336 mal Typ (28,25,203)  [ 2 verschiedene OV ]
  43008 mal Typ (34,25,197)  [ 3 verschiedene OV ]
  71680 mal Typ (42,25,189)  [ 3 verschiedene OV ]
  71680 mal Typ (43,25,188)  [ 3 verschiedene OV ]
 143360 mal Typ (46,25,185)  [ 4 verschiedene OV ]
  71680 mal Typ (52,25,179)  [ 3 verschiedene OV ]
  71680 mal Typ (59,25,172)  [ 3 verschiedene OV ]
 215040 mal Typ (60,25,171)  [ 6 verschiedene OV ]
 286720 mal Typ (63,25,168)  [ 5 verschiedene OV ]
  89600 mal Typ (65,25,166)  [ 5 verschiedene OV ]
 516096 mal Typ (76,25,155)  [ 14 verschiedene OV ]
 716800 mal Typ (78,25,153)  [ 14 verschiedene OV ]
 430080 mal Typ (79,25,152)  [ 6 verschiedene OV ]
 107520 mal Typ (87,25,144)  [ 4 verschiedene OV ]
 107520 mal Typ (89,25,142)  [ 3 verschiedene OV ]
 716800 mal Typ (90,25,141)  [ 17 verschiedene OV ]
 157696 mal Typ (91,25,140)  [ 9 verschiedene OV ]
 430080 mal Typ (92,25,139)  [ 6 verschiedene OV ]
 143360 mal Typ (94,25,137)  [ 4 verschiedene OV ]
2193408 mal Typ (103,25,128)  [ 46 verschiedene OV ]

26    4429824
  71680 mal Typ (31,26,199)  [ 4 verschiedene OV ]
  43008 mal Typ (37,26,193)  [ 3 verschiedene OV ]
  43008 mal Typ (44,26,186)  [ 3 verschiedene OV ]
 286720 mal Typ (58,26,172)  [ 6 verschiedene OV ]
  43008 mal Typ (65,26,165)  [ 3 verschiedene OV ]
 143360 mal Typ (71,26,159)  [ 4 verschiedene OV ]
 143360 mal Typ (74,26,156)  [ 4 verschiedene OV ]
 114688 mal Typ (77,26,153)  [ 7 verschiedene OV ]
  86016 mal Typ (80,26,150)  [ 4 verschiedene OV ]
 107520 mal Typ (86,26,144)  [ 3 verschiedene OV ]
 645120 mal Typ (89,26,141)  [ 12 verschiedene OV ]
 215040 mal Typ (90,26,140)  [ 5 verschiedene OV ]
1089536 mal Typ (102,26,128)  [ 24 verschiedene OV ]
1397760 mal Typ (115,26,115)  [ 28 verschiedene OV ]

27    3225600
  14336 mal Typ (23,27,206)  [ 3 verschiedene OV ]
  43008 mal Typ (38,27,191)  [ 4 verschiedene OV ]
```

```
        7168 mal Typ (44,27,185)  [ 2 verschiedene OV ]
      193536 mal Typ (54,27,175)  [ 9 verschiedene OV ]
      143360 mal Typ (55,27,174)  [ 4 verschiedene OV ]
       43008 mal Typ (64,27,165)  [ 4 verschiedene OV ]
       14336 mal Typ (65,27,164)  [ 3 verschiedene OV ]
       43008 mal Typ (70,27,159)  [ 3 verschiedene OV ]
      215040 mal Typ (72,27,157)  [ 5 verschiedene OV ]
      430080 mal Typ (73,27,156)  [ 7 verschiedene OV ]
       14336 mal Typ (79,27,150)  [ 3 verschiedene OV ]
      100352 mal Typ (86,27,143)  [ 8 verschiedene OV ]
      430080 mal Typ (87,27,142)  [ 7 verschiedene OV ]
      286720 mal Typ (88,27,141)  [ 5 verschiedene OV ]
       43008 mal Typ (90,27,139)  [ 3 verschiedene OV ]
     1204224 mal Typ (101,27,128)  [ 31 verschiedene OV ]

28    1668864
         256 mal Typ ( 9,28,219)  [ 1 verschiedene OV ]
      107520 mal Typ (49,28,179)  [ 4 verschiedene OV ]
       71680 mal Typ (53,28,175)  [ 3 verschiedene OV ]
       71680 mal Typ (67,28,161)  [ 4 verschiedene OV ]
      215040 mal Typ (69,28,159)  [ 6 verschiedene OV ]
       17920 mal Typ (82,28,146)  [ 2 verschiedene OV ]
      322560 mal Typ (85,28,143)  [ 9 verschiedene OV ]
      647168 mal Typ (100,28,128)  [ 16 verschiedene OV ]
      215040 mal Typ (114,28,114)  [ 3 verschiedene OV ]

29     645120
       71680 mal Typ (46,29,181)  [ 4 verschiedene OV ]
       71680 mal Typ (68,29,159)  [ 4 verschiedene OV ]
      286720 mal Typ (84,29,143)  [ 7 verschiedene OV ]
      215040 mal Typ (99,29,128)  [ 6 verschiedene OV ]

30     577024
       43008 mal Typ (63,30,163)  [ 3 verschiedene OV ]
      107520 mal Typ (81,30,145)  [ 4 verschiedene OV ]
       43008 mal Typ (98,30,128)  [ 3 verschiedene OV ]
      383488 mal Typ (113,30,113)  [ 12 verschiedene OV ]

31    1300992
       17920 mal Typ (35,31,190)  [ 3 verschiedene OV ]
       43008 mal Typ (39,31,186)  [ 4 verschiedene OV ]
       28672 mal Typ (49,31,176)  [ 5 verschiedene OV ]
      143360 mal Typ (64,31,161)  [ 5 verschiedene OV ]
       43008 mal Typ (65,31,160)  [ 4 verschiedene OV ]
       43008 mal Typ (67,31,158)  [ 3 verschiedene OV ]
       43008 mal Typ (70,31,155)  [ 3 verschiedene OV ]
        7168 mal Typ (77,31,148)  [ 2 verschiedene OV ]
      157696 mal Typ (81,31,144)  [ 10 verschiedene OV ]
      143360 mal Typ (82,31,143)  [ 4 verschiedene OV ]
       71680 mal Typ (84,31,141)  [ 3 verschiedene OV ]
      559104 mal Typ (97,31,128)  [ 23 verschiedene OV ]

32     150528
        7168 mal Typ (24,32,200)  [ 2 verschiedene OV ]
       14336 mal Typ (80,32,144)  [ 3 verschiedene OV ]
       43008 mal Typ (96,32,128)  [ 4 verschiedene OV ]
       86016 mal Typ (112,32,112)  [ 7 verschiedene OV ]
```

```
 33     451584
         71680 mal Typ (57,33,166)  [ 4 verschiedene OV ]
         43008 mal Typ (59,33,164)  [ 4 verschiedene OV ]
        107520 mal Typ (77,33,146)  [ 5 verschiedene OV ]
         14336 mal Typ (80,33,143)  [ 2 verschiedene OV ]
        215040 mal Typ (95,33,128)  [ 8 verschiedene OV ]

 34      53760
         53760 mal Typ (111,34,111)  [ 3 verschiedene OV ]

 35     129024
         14336 mal Typ (53,35,168)  [ 2 verschiedene OV ]
         71680 mal Typ (75,35,146)  [ 4 verschiedene OV ]
         43008 mal Typ (93,35,128)  [ 3 verschiedene OV ]

 36     172544
          2048 mal Typ (29,36,191)  [ 2 verschiedene OV ]
         14336 mal Typ (50,36,170)  [ 3 verschiedene OV ]
          2048 mal Typ (64,36,156)  [ 2 verschiedene OV ]
          7168 mal Typ (71,36,149)  [ 2 verschiedene OV ]
         57344 mal Typ (92,36,128)  [ 7 verschiedene OV ]
         89600 mal Typ (110,36,110)  [ 5 verschiedene OV ]

 38     129024
         14336 mal Typ (44,38,174)  [ 3 verschiedene OV ]
         43008 mal Typ (70,38,148)  [ 5 verschiedene OV ]
         71680 mal Typ (90,38,128)  [ 5 verschiedene OV ]

 40      53760
         17920 mal Typ (66,40,150)  [ 2 verschiedene OV ]
         14336 mal Typ (88,40,128)  [ 2 verschiedene OV ]
         21504 mal Typ (108,40,108)  [ 2 verschiedene OV ]

 42      23040
          2048 mal Typ (30,42,184)  [ 2 verschiedene OV ]
          2048 mal Typ (65,42,149)  [ 2 verschiedene OV ]
         14336 mal Typ (86,42,128)  [ 4 verschiedene OV ]
          4608 mal Typ (107,42,107)  [ 3 verschiedene OV ]

 44       8960
          8960 mal Typ (106,44,106)  [ 2 verschiedene OV ]

 46      21504
          7168 mal Typ (56,46,154)  [ 3 verschiedene OV ]
         14336 mal Typ (82,46,128)  [ 3 verschiedene OV ]

 52       3584
          3584 mal Typ (102,52,102)  [ 2 verschiedene OV ]

 56       2304
           256 mal Typ (37,56,163)  [ 1 verschiedene OV ]
          2048 mal Typ (72,56,128)  [ 2 verschiedene OV ]

 70        128
           128 mal Typ (93,70,93)  [ 1 verschiedene OV ]

Größtes aufgetretenes Vektorelement: 18  (kmax=32)
   - - - -   E N D E   - - - -
```

7. Tabellen

Das durch die Berechnungen gewonnene Datenmaterial ermöglicht einen guten Einblick in die Strukturen des Einheitswürfels in höheren Dimensionen. Der Vorteil dieser Daten ist es, eine rasche "Erstüberprüfung" von intuitiv gefundenen Zusammenhängen zu ermöglichen.

Auf den folgenden Seiten sind daher die wichtigsten Zahlenwerte in Tabellenform zusammengefaßt. Dabei handelt es sich zum Großteil um durch das Programm enumerativ ermittelte Werte. Wurden zusätzliche Werte berechnet, so ist die verwendete Formel in der Tabelle angegeben.

In Tabelle 18 wird die Anzahl aller Hyperebenen im Q_d nach deren Parallelität angegeben. Dabei gilt der in Satz 1 gezeigte Zusammenhang zwischen schrägen ($e_0(d)$) und k-parallelen ($e_k(d)$) Hyperebenen. Die Anzahl aller möglichen Hyperebenen im Q_d ist dann die Zeilensumme über alle Arten schräger und paralleler Hyperebenen.

Tabelle 19 ermöglicht einen Vergleich der tatsächlichen Anzahl von Hyperebenen im Q_d mit der trivialen Obergrenze von $\binom{2^d}{d}$. Weiters wird hier der maximale Betrag des Koordinatenwerts von nicht kürzbaren Orthogonalvektoren für schräge Hyperebenen im Vergleich zur theoretischen Schranke und zur Determinante von 0-1-Matrizen dargestellt. Bis zur Dimension acht ist dabei der maximale Koordinatenwert bekannt, für die Dimensionen neun und zehn wird der mögliche Bereich angegeben. Die obere Schranke ist dabei durch Satz 5 (Seite 26) festgelegt, die untere Schranke wird durch die Existenz von bekannten Orthogonalvektoren gebildet. Beispiele für solche Vektoren sind in der Tabelle ebenfalls angeführt.

Tabelle 20 enthält verschiedene Angaben über die Anzahl von Eckpunkten aus Q_d, die gleichzeitig auf einer Hyperebene liegen können. Für die minimale und maximale Anzahl von Ecken (d bzw. 2^{d-1}) ist zusätzlich die Anzahl der dadurch erzeugten Hyperebenen angegeben. Ergänzend wird für schräge Hyperebenen noch die Anzahl unterschiedlicher Typen sowie die Anzahl unterschiedlicher Orthogonalvektor-Arten aufgelistet.

Tabelle 21 enthält den maximalen Betrag des Koordinatenwertes von Orthogonalvektoren schräger Hyperebenen. Angegeben ist dieser in Abhängigkeit von der Dimension und Eckanzahl der Hyperebenen. Die Angabe eines Bereichs (z.B. "2..7") bedeutet dabei, daß es sowohl Orthogonalvektoren mit einem größten Koordinatenwert der unteren (z.B.: 2) als auch der oberen (z.B.: 7) Bereichsgrenzen gibt. Natürlich kann es auch Vektoren mit einem größten Koordinatenwert innerhalb des Bereichs geben. Ist nur ein Wert angegeben, so ist dies für alle mögliche Orthogonalvektoren genau der größte Koordinatenwert.

Dimension d	Anzahl Hyper-ebenen im Q_d	$e_0(d)$	$e_1(d)$	$e_2(d)$	$e_3(d)$	$e_4(d)$	$e_5(d)$	$e_6(d)$	$e_7(d)$	$e_8(d)$	$e_9(d)$
1	2	2									
2	6	2	4								
3	20	8	6	6							
4	140	88	32	12	8						
5	3254	2704	440	80	20	10					
6	252434	234688	16224	1320	160	30	12				
7	71343208	69640192	1642816	56784	3080	280	42	14			
8	86246755608	85682904704	557121536	6571264	151424	6160	448	56	16		
9	?	?	771146142336	2507046912	19713792	340704	11088	672	72	18	
10	?	?	?	3855730711680	8356823040	49284480	681408	18480	960	90	20

Tabelle 18: Anzahl k-paralleler Hyperebenen im Q_d

Dimension d	Anzahl Hyperebenen im Q_d	triviale Obergrenze $\binom{2^d}{d}$	% von $\binom{2^d}{d}$	Maximaler Koordinatenwert von teilerfremden Orthogonalvektoren *)	$2\left(\frac{d}{4}\right)^{\frac{d}{2}}$	Beispielvektor mit maximalem Koordinatenwert	max $\lvert\det(A_d)\rvert$ für $A_d \in \{0,1\}^{dxd}$
1	2	2	100.00000	1	1.00	(1)	1
2	6	6	100.00000	1	1.00	(-1 1)	1
3	20	56	35.71429	1	1.30	(-1 1 1)	2
4	140	1820	7.69231	2	2.00	(-2 1 1 1)	3
5	3254	201376	1.61588	3	3.49	(-3 1 1 1 1)	5
6	252434	74974368	0.33669	5	6.75	(-5 1 1 1 2 2)	9
7	71343208	94525795200	0.07547	9	14.18	(-9 1 1 2 2 3 4)	32
8	86246755608	409663695276000	0.02105	18	32.00	(-18 1 2 3 4 5 6 7)	≥ 42, ≤ 76
9	?	$6.2081\ 10^{18}$	?	≥42 , ≤ 76	76.89	(-3 -10 -22 -42 1 4 8 13 24)	≥ 96, ≤ 195
10	?	$3.3427\ 10^{23}$	?	≥96 , ≤ 195	195.31	(-11 -34 -52 -96 4 6 21 37 39 66)	≥ 193, ≤ 521

*) Weitere Ergebnisse für Q_{11} bis Q_{14} sind: Q_{11} : ≥193 : (-11 -34 -52 -96 4 6 21 37 39 66 193) , Q_{12} : ≥366 : (-11 -34 -52 -96 -366 4 6 21 37 39 66 193)

Q_{13} : ≥632 : (-34 -37 -65 -179 -317 7 10 11 18 73 97 101 632) Q_{14} : ≥1144 : (-13 -23 -29 -131 -132 -203 -613 35 40 44 55 215 224 1144)

Tabelle 19: Anzahl von Hyperebenen im Q_d sowie Länge des Orthogonalvektors einer Hyperebene

Dimension d	Anzahl Hyperebenen im Q_d	Anzahl von Ecken in Hyperebenen	Anzahl verschiedener Eckanzahlen in schrägen Hyperebenen	Anz. von Hyperebenen mit genau d Ecken (für d ≥ 3 schräg !)	Anz. von Hyperebenen mit genau 2^{d-1} Ecken ($e_{d-1}(d)+e_{d-2}(d)$)	Anzahl verschiedener Typen von schrägen Hyperebenen im Q_d	Anzahl verschiedener (schräger) Arten von Orthogonalvektoren
1	2	1	1	2	2	1	1
2	6	2	1	6	6	1	1
3	20	3;4	1	8	12	1	1
4	140	4;6;8	2	80	20	3	4
5	3254	5;7;8;10;12;16	4	2112	30	8	17
6	252434	6 ... 32	10	123648	42	35	137
7	71343208	7 ... 64	19	23963648	56	219	2483
8	86246755608	8 ... 128	37	25806963456	72	1293	145415

Tabelle 20: Anzahl von Ecken, Typen und Orthogonalvektoren für Hyperebenen im Q_d

Ecken / Dim.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1																								
2		1																							
3			1																						
4				1..2		1																			
5					1..3		2	2		1															
6						1..5	3..4	2..4	3	2..3	2..3	2		2	1..2					1					
7							1..9	4..8	2..7	3..6	3..6	2..5	2..5	3..4	3..4	2..4	3	2..3		2..3	1..3	2..3		2	
8								1..18	4..16	2..14	3..12	3..12	2..10	3..10	4..9	2..8	2..8	3..8	3..7	2..6	3..6	2..6	4..6	3..5	2..5

D \ E	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	...	52	...	56	...	70
7	2				2					1																	
8	3..5	3..5	1..4	3..4	3..4	2..4	2..4	3	3	2..3	3..**4***		2..3		2..3		2..3		2		2		2		1..2		1

* Erstmaliges Auftreten von nicht monoton fallenden maximalen Koordinatenwerten. (HE-Typen 2x(64,36,156) und 3x(92,36,128))

Tabelle 21: Maximaler Betrag des Koordinatenwertes von Orthogonalvektoren schräger Hyperebenen in Abhängigkeit von der Dimension und Eckanzahl

8. Zusammenfassung

> »Edamer Mieze«, begann sie ein wenig stockend, denn sie war gar nicht sicher, ob ihr diese Bezeichnung wohl angenehm wäre - aber das Grinsen wurde davon nur noch etwas breiter. »Aha«, dachte Alice, »das hat ihr gefallen«, und fuhr fort: »Würdest du mir bitte sagen, wie ich von hier aus weitergehen soll?«
> »Das hängt zum großen Teil davon ab, wohin Du möchtest«, sagte die Katze.
> »Ach, wohin ist mir eigentlich gleich - «, sagte Alice.
> »Dann ist es auch egal, wie du weitergehst«, sagte die Katze.
>
> LEWIS CARROLL
> In: *Alice im Wunderland*

Im Rahmen dieser Arbeit wurden die durch Eckpunkte eines Einheitshyperwürfels erzeugten affinen Hyperebenen untersucht:

- Durch ein stark auf das Problem optimiertes Programm wurde eine vollständige enumerative Berechnung aller relevanten Werte bis einschließlich der achten Dimension durchgeführt.
- Ausgehend von den so gewonnenen Informationen wurden mehrere allgemein gültige Sätze hergeleitet.

Diese Vorgangsweise, das enumerative Ermitteln von Datenmaterial und das anschließende Herleiten von allgemein gültigen Sätzen, hat sich als sehr fruchtbar erwiesen. So wurden einige der gezeigten Sätze nach systematischem Studium der errechneten Werte gefunden, und es ist durchaus vorstellbar, daß eine weitere Überprüfung noch zusätzliche Zusammenhänge aufdeckt.

Die theoretischen Ergebnisse beziehen sich zum Großteil auf den Orthogonalvektor einer Hyperebene. Es konnten Schranken für die Größe des Orthogonalvektors sowie ein Zusammenhang seiner Koordinatenwerte mit der k-Parallelität einer Hyperebene angegeben werden. Weiters wurde dargestellt, unter welchen Bedingungen die Spur einer Hyperebene die konvexe Hülle von Eckpunkten aus dem Einheitswürfel ist.

Durch ein Gegenbeispiel wurde gezeigt, daß die Angabe des Typs einer Hyperebene durch die Anzahl von Ecken auf, oberhalb und unterhalb dieser Hyperebene, nicht eindeutig ist. D.h. es gibt verschiedene Arten von Hyperebenen die demselben Typ entsprechen.

Die Anzahl von Hyperebenen im Q_d wächst exponentiel mit der Dimension d und ist durch $\binom{2^d}{d}$ nach oben beschränkt. Aufgrund der überwiegenden Anzahl von Hyperebenen mit genau d Eckpunkten aus Q_d erweist sich diese Schranke als relativ gut. Gezeigt wurde auch, daß die Anzahl von schrägen Hyperebenen jene von k-parallelen weit übertrifft.

Wichtige Ergebnisse wurden über den Zusammenhang zwischen verschiedenen Arten von Hyperebenen gezeigt. So konnten k-parallele Hyperebenen im Q_d auf schräge Hyperebenen im Q_{d-k} sowie schräge, unsymmetrische Hyperebenen im Q_d auf schräge, symmetrische Hyperebenen im Q_{d+1} reduziert werden. Das bedeutet für weitere Untersuchungen, daß nur mehr schräge, symmetrische Hyperebenen betrachtet werden müssen, da sich alle anderen Arten von Hyperebenen darauf zurückführen lassen!

9. Literaturverzeichnis:

The next best thing to knowing something is knowing where to find it.

SAMUEL JOHNSON, 1709-1784

[ABCO] Noga Alon / E. E. Bergmann / D. Coppersmith / A. M. Odlyzko, Balancing sets of vectors, IEEE Transactions on Information Theory Vol. 34 pp 128-130, 1988

[ALFÜ] Noga Alon / Zoltán Füredi, Covering the Cube by affine Hyperplanes, to be published

[BAK] K. A. Baker, A Generalization of Sperner´s Lemma, Journal of Combinatorial Theory Vol. 6 pp 224-225, 1969

[BRON] I. N. Bronstein / K. A. Semendjajew, Taschenbuch der Mathematik, Verlag Harri Deutsch, Thun und Frankfurt/Main, 1957/87

[CLSH] K.L. Clarkson / P.W. Shor, Algorithms for diametral pairs and convex hulls that are optimal, randomized, and incremental., Proc. 4th Ann. ACM Symp. Computational Geometry , 12-17, 1988

[COX] Harald S. M. Coxeter, Regular Polytopes, Dover Publications, New York, 1963/73

[DEW1] Alexander K. Dewdney, Das Planiversum, Paul Zsolnay-Verlag, Wien/Hamburg, 1985

[DEW2] Alexander K. Dewdney, Computer Kurzweil, Spektrum der Wissenschaft Juni 1986 Seite 6 ff, Heidelberg, 1986

[EDEL] Herbert Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-Verlag, Berlin New York Heidelber, 1987

[FISCH] Gerd Fischer, Lineare Algebra, Vieweg, Braunschweig, 1975/86

[GAJO] Michael R. Garey / David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Bell Laboratories/Freeman, San Francisco, 1979

[GARD] Martin Gardner, Mathematischer Zirkus, Verlag Ullstein/ Berlin/ Frankfurt/ Wien, 1988

[GLS] Martin Grötschel / Laszlo Lovasz / Alexander Schryver, Geometric Algorithms and combinatorical optimization, Berlin / Heidelberg, Springer Verlag, 1988

[HALL] Marshall Hall Jr., Combinatorial Constructions, Studies in combinatorics (Gian-Carlo Rota ed.) MAA Studies in mathematics Vol. 17 pp 218-253, 1978

[HRL] Eberhard Herter / Walter Röcker / Wolfgang Lörcher, Nachrichtentechnik: Übertragung, Vermittlung und Verarbeitung, Carl Hanser Verlag, München / Wien, 1981

[HSIA] Wu-Yi Hsiang, Sphere Packings and Spherical Geometry - Kepler's Conjecture and Beyond, Preprint, Center for Pure and Applied Mathematics, University of California, Berkly, Juli 1991.

[JACO] Konrad Jacobs, Einführung in die Kombinatorik, deGruyter, Berlin / New York, 1983

[KNU1] Donald E. Knuth, TheArt of Computer Programming Volume 1: Fundamental Algorithms, Addison-Wesley, Menlo Park, 1968/1973

[KNU2] Donald E. Knuth, TheArt of Computer Programming Volume 2: Seminumerical Algorithms, Addison-Wesley, Menlo Park, 1969/1981

[MEHL] Kurt Mehlhorn, Datenstrukturen und effiziente Algorithmen, Band 1: Sortieren und Suchen, B. G. Teubner, Stuttgart, 1988

[MESC] Herbert Meschkowski, Mathematisches Begriffswörterbuch, B.I.-Wissenschaftsverlag, Mannheim / Wien / Zürich, 1976

[MOSH] Bernard Moret / Henry Shapiro, Algorithms from P to NP, Vol 1, Redwood City / Calif.:The Benjamin / Cummings, 1991

[NEIL] Patrick E. O´Neil, Hyperplane cuts of an n-cube, Discrete Mathemathics Vol. 1 pp193-195, 1971/72

[PRSH] Franco P. Preparata / Michael I. Shamos, Computational Geometry, Springer-Verlag, Berlin New York Heidelber, 1985

[SARN] Karl Sarnow, Der Hypercubus, c´t - Magazin für Computertechnik 2/87 Seite 124 ff , Verlag Heise, Hannover, 1987

[SEDG] Robert Sedgewick, Algorithmen, Addison-Wesley, Bonn, 1991

[SLOA] N. J. A. Sloane, Kugelpackungen im Raum, Spektrum der Wissenschaft März 1984 Seite 120 ff, Heidelberg, 1984

[STEW] Ian Stewart, Mathematische Unterhaltung, Spektrum der Wissenschaft September 1992 Seite 12 ff, Heidelberg, 1992

[VOLK] Klaus Volkert, Die Geschichte der pathologischen Funktionen - Ein Beitrag zur Entstehung der mathematischen Methodologie, Archive for History of Exact Sciences, Band 37, Heft 3, S 193-232, 1987

[WELL] David Wells, Das Lexikon der Zahlen - Nachrichten von $\sqrt{17}$ bis $3\uparrow\uparrow\uparrow 3$, Fischer Taschenbuch Verlag, Frankfurt am Main, Juli 1990

Lebenslauf

9. September 1965, 6^{15}	geboren in Spittal an der Drau, Kärnten
1972 - 1976	Besuch der Volksschule Ost in Spittal an der Drau
1976 - 1980	Besuch der Hauptschule III in Spittal an der Drau
1980 - 1985	Besuch der Höheren Technischen Bundeslehr- und Versuchsanstalt Klagenfurt, Höhere Lehranstalt für Nachrichtentechnik und Elektronik
4. Juni 1985	Reifeprüfung mit ausgezeichnetem Erfolg bestanden
1985/1986	Ableistung des Präsenzdienstes beim Jägerbataillon 26, Spittal an der Drau
2. Oktober 1986	Immatrikulation an der Technischen Universität Graz, Beginn des Studiums der Technischen Mathematik
16. Dezember 1988	Ablegen der ersten Diplomprüfung mit Auszeichnung
Jänner 1992	Beginn der Diplomarbeit am Institut für Grundlagen der Informationsverarbeitung