

Kantonsschule Baden

Simulationen auf der Kugel: Anordnungen von Punkten und Energieberechnungen

Maturarbeit

von

Lucas Brönnimann und Martin Lanter

ausgeführt unter der Leitung von

H.R. Schneebeil

und

M. Speck

an der Kantonsschule Baden

Baden, März 2007

Inhaltsverzeichnis

1	Einleitung	2
2	Methoden	3
2.1	Aufbau der Simulationen	3
2.2	Deterministische Simulation	3
2.3	Stochastische Simulation	3
2.4	Stärken und Schwächen der Programme	4
2.5	Visualisierung	5
2.6	GUI	7
3	Ergebnisse	8
3.1	Testproblem: 8 Punkte	8
3.2	Falsifizierung der Behauptung	9
3.3	Weitere Resultate	9
3.4	Geschwindigkeit der Programme	10
3.4.1	Geschwindigkeit der deterministischen Simulation	11
3.4.2	Geschwindigkeit der stochastischen Simulation	12
4	Diskussion	14
4.1	Federenergie aller Abstände	14
4.2	Federenergie der echten Kanten	14
4.2.1	Idee	14
4.2.2	Ergebnis	14
4.2.3	Energie durch Kanten	15
4.3	Coulomb-Gesetz	16
4.4	Oberfläche und Volumen	17
4.5	Sonstige Invarianten	17
4.6	das N^2 Gesetz	18
5	Noch offene Fragen	20
5.1	Was passiert wenn n sehr gross wird?	20
5.2	Anzahl Rechenschritte für n Punkte	20
5.3	Konstante optimieren	20
6	Schlusswort	21
7	Quellen	22

Zusammenfassung

Die allgemeine Frage lautet: Wie sind n Punkte auf der Einheitskugel anzuordnen, so dass die Punkte untereinander den grösstmöglichen Abstand haben? Es werden zwei verschiedene Simulationen vorgestellt, um diese Frage zu klären:

- -deterministisch, die Punkte stossen sich gegenseitig nach einem physikalischen Gesetz ab. Die Simulation sucht einen Gleichgewichtszustand.
- -stochastisch, eine Anfangskonfiguration wird durch „Mutationen“ gestört und nach einem darwinschen Schema auf die gewünschte Eigenschaft hin selektioniert.

Mit beiden Methoden gelingt es, die Behauptung zu widerlegen, dass die Eckenanordnung der platonischen Körper auf der Einheitskugel im Sinne der Frage optimal sind. Die Behauptung stimmt für die platonischen Körper, die durch Dreiecke berandet werden.

Die Methoden werden bezüglich Effizienz und Genauigkeit verglichen. Zudem enthält die Arbeit mehrere Eigenentwicklungen zur algorithmischen Geometrie (Berechnung der konvexen Hülle) und Betrachtungen zu Invarianten von Punktekongfigurationen mit dem Ziel, die „Güte“ von Punktekongfigurationen durch *eine* Zahl auszudrücken, um globale Aussagen treffen zu können.

1 Einleitung

Wie ordnet man Punkte auf einer Kugel optimal an? Diese Frage beinhaltet wesentlich mehr, als man ihr beim ersten Blick ansieht. Insbesondere stellt sich aber die Frage, ob die Eckpunkte der platonischen Körper immer eine optimale Punktekongfiguration liefern. Wie findet man heraus, welche von zwei Figuren mit n Eckpunkten auf einer Kugeloberfläche günstiger ist? Was für Möglichkeiten gibt es, um optimale Verteilungen zu finden und welche davon bewährt sich am besten?

Um Antworten auf diese Fragen zu finden, wurden zwei Matlab-Simulationen programmiert, welche die Punkte auf der Kugel möglichst optimal anordnen. Obwohl beide in den meisten Fällen die gleichen Resultate liefern, basieren sie auf sehr unterschiedlichen Prinzipien. Während der Arbeit an diesem Projekt und der Programmierung der Simulationen ergaben sich auch weitere Fragestellungen, aber auch Erkenntnisse, welche in direktem Zusammenhang mit der Beantwortung der Fragestellung stehen.

In diesem Bericht werden die Simulationen genauer erklärt und die Vor- und Nachteile der einzelnen Arbeitsweisen der Simulationen eruiert. Ebenso werden die Resultate präsentiert und auf verschiedene Eigenschaften überprüft, insbesondere geht es darum, wie man die Energie der entstanden Körper berechnen kann.

Fragestellung Eine Anzahl Punkte n wird mit einer Simulation auf einer Kugeloberfläche so verteilt, dass die Punkte untereinander den grösstmöglichen Abstand haben. These: Sind die Eckpunkte der platonischen Körper Musterbeispiele für die Verteilung der Punkte?

2 Methoden

Nach dem Vorbild aus der Biologie und der Physik bieten sich zwei prinzipiell verschiedene Optimierungsmethoden an:

- Nach dem Muster der darwinschen Evolution eine stochastische Simulation.
- Nach dem Muster von physikalischen Systemen, die in einem stabilen Gleichgewicht ein Energieminimum anstreben, eine deterministische Simulation.

2.1 Aufbau der Simulationen

Als Input gibt man bei jeder Simulation die Anzahl gesuchter Eckpunkte $n \leq 20$ der Figur, sowie die Anzahl Optimierungsschritte S vor. Wenn keine Startkonfiguration vorgegeben wird, generiert die Simulation selber eine zufällige Anordnung von Punkten auf der Einheitskugel und wählt diese als Startkonfiguration.

Im Algorithmus kommt auch eine Konstante k vor, welche bestimmt, wie stark die Punkte während eines Optimierungsschrittes verschoben werden können. So wird sichergestellt, dass sich die Punkte nicht zu stark bewegen können, aber auch nicht zu kleine Bewegungen ausführen. Auf die Wahl der Konstante wird später noch genauer eingegangen.

Nach S Schritten hört die Simulation auf und um die gewonnenen Daten zu vereinheitlichen, wird die Punktekonfiguration so gedreht, dass ein Punkt auf dem Nordpol der Kugel zu stehen kommt und ein weiterer Punkt auf der x-z-Ebene liegt.

Mit dem Hilfsprogramm Kugel2.m und der Matlabfunktion plot3 können die resultierenden Punktekoordinaten aufgezeichnet werden und mit den Hilfsprogrammen pnet.m und cnet.m werden zusätzlich die Kanten eingezeichnet, so dass die Figur besser erkennbar wird.

2.2 Deterministische Simulation

Die deterministische Simulation basiert auf der Vektorgeometrie. Aus der Sicht der Biologie kann man auch von einer lamarckschen Evolution sprechen.

Zuerst wird der Einfluss der Punkte 2 bis n auf den Punkt 1 gemäss dem Coulomb-Prinzip $F \propto 1/r^2$ berechnet und aufsummiert. Dieser neue Vektor wird normiert, sprich der Betrag des Vektors wird zu 1 und der Punkt landet wieder auf der Einheitskugel.

Dann wird diese Methode auf Punkt 2 angewendet, dann auf Punkt 3 und so weiter mit allen n Punkten. Ein solcher Zyklus ist ein einziger Optimierungsschritt. Dieser wird S mal durchgeführt. Dabei verbessert sich der Energiezustand der Punkte mit jeder Verschiebung eines Punktes. Wenn S genügend hoch (1'000-10'000 empfohlen) gewählt ist, liegen die Punkte jetzt „optimal“ verteilt.

2.3 Stochastische Simulation

Diese Simulation imitiert einen genetischen Algorithmus; die Punkte werden mit Zufallswerten in verschiedene Richtungen verschoben und es wird geprüft, ob

der neue Zustand besser ist als der alte. Eine Art darwinsche Evolution auf der Kugel also.

Hierbei gibt es zwei Varianten der Simulation: Die eine Variante verschiebt jeweils nur einen Punkt auf der Kugel und berechnet danach dessen Abstand zu allen anderen Punkten auf der Kugel. Ist dieser grösser geworden, so bleibt der Punkt dort, wo er hingeschoben wurde, andernfalls nimmt er wieder seine alte Position ein. Danach wird der nächste Punkt zufällig verschoben, bis alle Punkte einmal durchgerechnet worden sind. Insgesamt wird dieser Zyklus S mal durchgegangen, wobei die effektive Veränderung der Anordnung nur vom Zufall abhängig ist. Für die weiteren Berechnungen wurde diese Variante verwendet.

Bei der anderen Variante werden S mal alle Punkte gleichzeitig ein klein wenig verschoben und erst dann wird überprüft, ob die neue Situation besser geworden ist. Hierzu werden alle Abstände sämtlicher Punkte zu allen anderen auf der Kugel zusammengerechnet. Ist die neue Situation die nun durch Zufall entstanden ist energetisch günstiger als die alte, so wird die neue abgespeichert und die Punktekonfiguration nähert sich ebenfalls einem Optimum an.

2.4 Stärken und Schwächen der Programme

Im folgenden Abschnitt werden die beiden Methoden zur Berechnung der optimalsten Punktekonfigurationen verglichen und Vor- sowie Nachteile erfasst. Die deterministische Simulation schiebt die Punkte auf einem absolut voraus-sagbaren Weg an ihr Ziel. Wiederholt man die Simulation mit derselben Start-konfiguration mehrmals, so werden die Punkte jedes Mal genau denselben Weg zurücklegen. Bei der stochastischen Methode ist dies nicht der Fall, hier diffundieren die Punkte zufällig über die Kugel und werden, je nachdem ob sie die Lage dabei verbessern oder verschlechtern, wieder auf die alte Position zurückgeholt. Man kann dies mit einer Brown'schen Bewegung eines Teilchens vergleichen, mit dem Unterschied, dass die Simulation den Weg der Teilchen mehr oder weniger zu kontrollieren vermag. Die unterschiedlichen Wege der Punkte sind in Abb. 1 erkennbar.

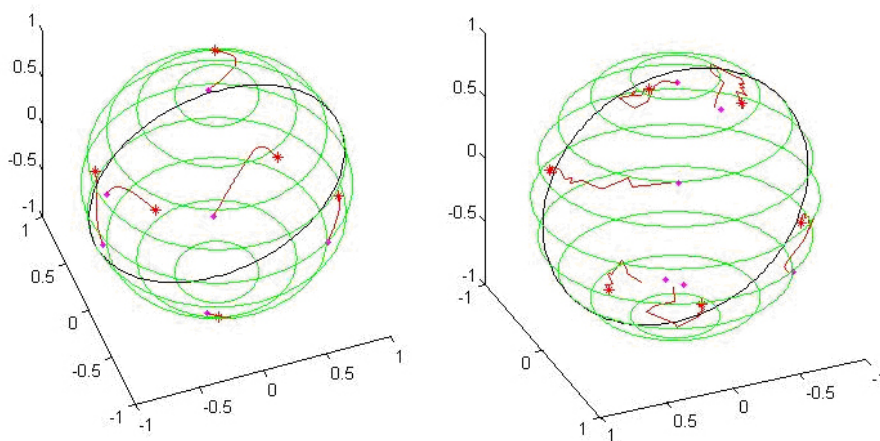


Abbildung 1: Vergleich der Punktwege: deterministisch (links), stochastisch (rechts)

Kriterium	Deterministisch	Stochastisch
Geschwindigkeit	Sehr schnell	Eher langsam
Approximation	Nach wenigen Schritten extrem genau	Nach vielen Schritten wird es immer unwahrscheinlicher, dass ein Punkt gefunden wird, der noch genauer ist.
Lokale Minima	Schafft es nur bei einer grossen Konstante aus dem lokalen Minimum Würfel auszubrechen	Sehr unwahrscheinlich, dass jemals zufällig eine Konfiguration gefunden wird, die besser als der Würfel ist, um aus dem Minimum auszubrechen
Symmetrische) Figuren (Doppelpyramide, alle Punkte auf Nord- und Südpol	Kann nicht aus einer Doppelpyramide ausbrechen, sofern die Konstante nicht künstlich erhöht wird	Es ist unwahrscheinlich aus einer Doppelpyramide ausbrechen zu können, aber wenn alle Punkte die gleichen sind oder sich an Nord- und Südpol gegenüber liegen, bricht diese Simulation aus.
Varianten (Dimensionen, anderes Gebiet)	Für mehrere Dimensionen anwendbar, wobei jedoch das Problem besteht, wie man kontrollieren soll, dass der Punkt nicht aus dem erlaubten Gebiet flieht.	Für beliebig viele Dimensionen und Gebiete anwendbar, die man mit Gleichungen abgrenzen kann. Ebenso ist es möglich, nach einem anderen Kriterium, als dem Abstand zwischen den Punkten zu optimieren, beispielsweise nach einem möglichst grossen Volumen.

Tabelle 1: Unterschiede der Methoden

In Tabelle 1 sind die Unterschiede übersichtlich aufgelistet. Eine Simulation mit einem deterministischen sowie einem stochastischen Aspekt ist ein „mächtiges“ Instrument. Wenn eine Konfiguration mit der deterministischen Methode berechnet werden soll, vorher oder nach einigen Schritten aber noch (stochastisch) durchgeschüttelt wird, schafft es die Simulation sowohl die Vorteile der deterministischen Simulation auszunutzen, als auch das negative Kriterium zu vermeiden, dass die Punkte unter Umständen in ein lokales Minimum fallen und nicht mehr herauskommen (was allerdings während des Projektes sowieso nie vorgekommen ist). Dies bedeutet, dass man in jedem Fall aus einer beliebigen Konfiguration, wie beispielsweise dem Würfel, die optimale Verteilung für die vorgegebene Anzahl Punkte finden kann, ohne dabei Qualitätseinbussen zu erleiden.

2.5 Visualisierung

Die Simulationen alleine geben zwar die Koordinaten der Ergebnisse aus, doch noch lässt sich damit nicht viel anfangen, es sei denn man hätte andere Koordinaten zum Vergleich. Mit Hilfe des plot-Befehls in Matlab können die Punkte jedoch aufgezeichnet werden und dazu zeichnet ein eigens dafür programmiertes Programm eine Kugel auf. Diese Kugel wird dargestellt, indem 8 Breitengrade, der Äquator und der 0° Meridian erstellt werden. Dies führt dann zu einem Bild wie in Abbildung 2 gezeigt.

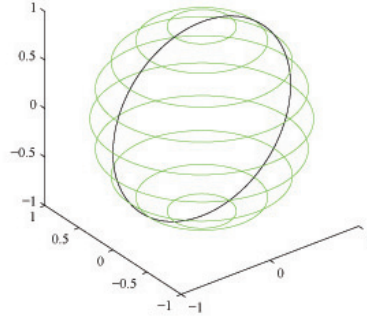


Abbildung 2: Darstellung der Kugel

Um eine bessere Übersicht über die Punkte zu bekommen, wird die Punktekongfiguration um die z- und die y-Achse gedreht, damit ein Punkt auf $[0|0|1]$ liegt. Dann wird die Kongfiguration um die z-Achse gedreht, damit ein zweiter Punkt auf der x-z-Ebene liegt. So sollten die Koordinaten gleicher Resultate möglichst dasselbe Bild zeigen. Dies geschieht mit dem Hilfsprogramm rot3.m.

Ein weiteres Programm sollte nun die Punkte so miteinander verbinden, dass die Kanten der Figur sichtbar werden. Dies ist nicht allzu einfach, denn man muss für jeden Punkt bestimmen, welche anderen Punkte mit ihm verbunden sind. Bei diversen Körpern ist jeder Punkt immer mit einer gleichen Anzahl anderer Punkte verbunden, was das ganze stark vereinfachen würde. Doch sobald dies bei mindestens einem Punkt nicht mehr stimmt, wie beispielsweise beim Resultat mit 7 oder 9 Punkten, könnte man sich nicht mehr darauf beziehen.

Das Hilfsprogramm pnet.m Um die Resultate analysieren und vergleichen zu können, wird eine Inzidenzmatrix benötigt, die aussagt, welcher Punkt mit welchen anderen Punkten verbunden ist. Eine 1 an der Stelle (i,j) steht für eine Verbindung zwischen Punkt P_i und P_j , eine 0 für keine Verbindung. Alle Verbindungen zusammen ergeben die konvexe Hülle der Figur, wie sie auf der

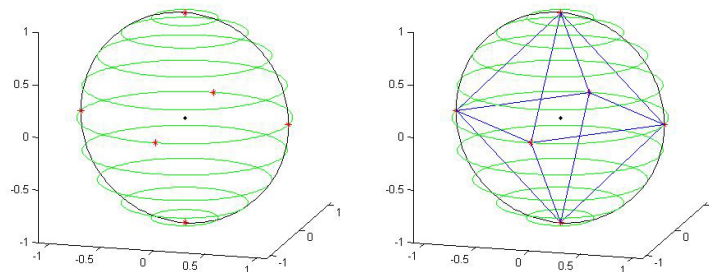


Abbildung 3: Vergleich der Figuren vor und nach dem Einsatz von pnet.m

Abbildung 3 auf der rechten Seite zu sehen ist.

Diese Inzidenzmatrix wird von pnet.m erstellt. Dies geschieht mit Hilfe der Determinanten von je 3 Richtungsvektoren zwischen 4 Punkten jeder möglichen Kombination. Die Determinante berechnet das orientierte Volumen dieser 3 Vektoren und anhand dieser Volumen kann ermittelt werden, ob eine Verbindung zwischen zwei Punkten zur konvexen Hülle gehört ($=1$), oder eine Diagonale ($=0$) darstellt.

2.6 GUI

Die Abbildung 4 zeigt die graphische Benutzeroberfläche des Endprodukts: Ein Programm, welches beide Simulationen vereint und in der Lage ist aus jeder beliebigen Startkonfiguration mindestens für $n \leq 20$ die optimale Konfiguration mit beliebiger Genauigkeit zu finden und diese aufzuzeichnen. Ebenso können einige Invarianten berechnet werden, auf diese wird dann im Kapitel 4: Diskussion genauer eingegangen. Zur Benutzung des Programms wird wie für alle Simulationen Matlab benötigt.

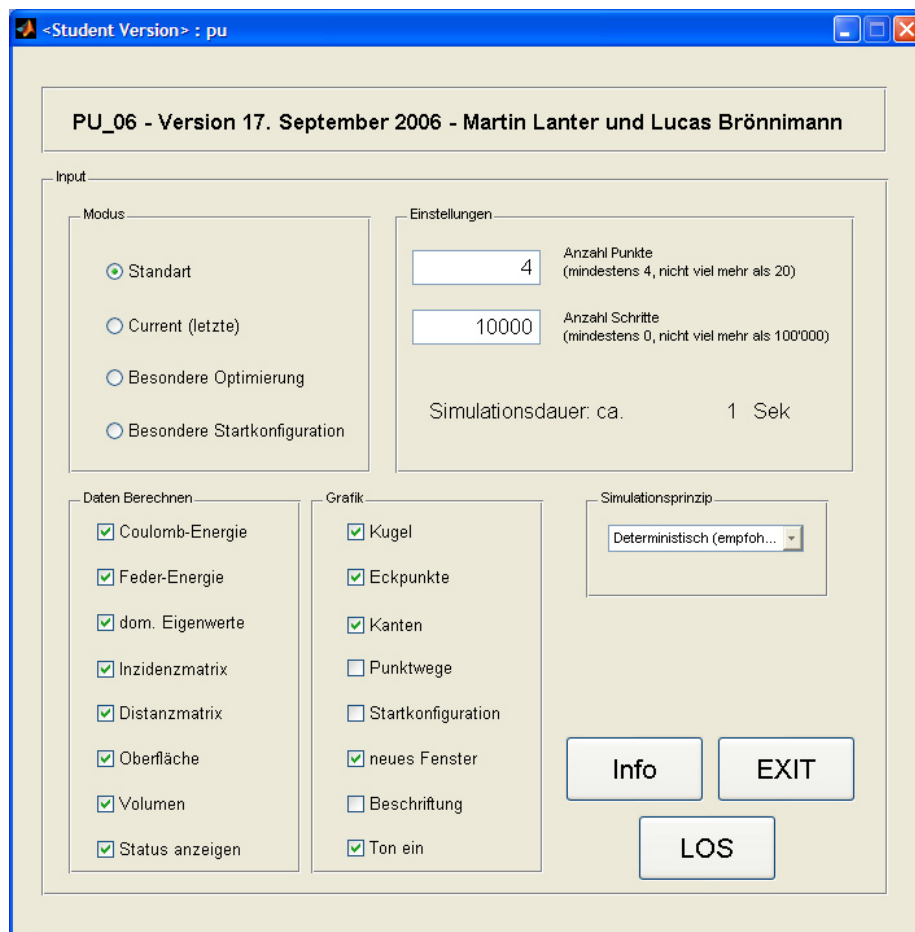


Abbildung 4: Graphische Benutzeroberfläche der Simulationen

3 Ergebnisse

3.1 Testproblem: 8 Punkte

Um die beiden Simulationen in Bezug auf die Beantwortung der Fragestellung zu testen, kann man ein Testproblem lösen. Im folgenden Abschnitt wird beschrieben, wie sich die beiden Situation verhalten, wenn $n = 8$ gewählt wird.

Zunächst einmal muss die Ausgangssituation erstellt werden. Hierbei handelt es sich im Normalfall um eine völlig zufällige Verteilung der Punkte auf der Kugel. Nun verbessert man die Positionen der einzelnen Punkte der Reihe nach, wobei bei jedem Optimierungsschritt die einzelnen Abstände zwischen den Punkten etwas grösser werden. Der zugehörige platonische Körper zu 8 Punkten ist ein Würfel und wenn die Behauptung stimmt, sollte nun ein solcher zu erwarten sein. Doch zeichnet man die Punkte auf und verbindet man die Kanten miteinander, sieht man sofort, dass der entstandene Körper kein Würfel sein kann. Entstanden ist ein verdrehter Würfel, siehe Abbildung 5. Der verdrehte Würfel ist nichts anderes als ein Würfel, bei welchem eine Seite um $\pi/4$ (45°) verdreht worden ist, wodurch ein Körper mit 8 Ecken, 10 Flächen und 16 Kanten entsteht. Es handelt sich hierbei um denselben Körper wie er auch schon in vergangenen Arbeiten (H. Rutishauser, 1945) genauer beschrieben wurde.

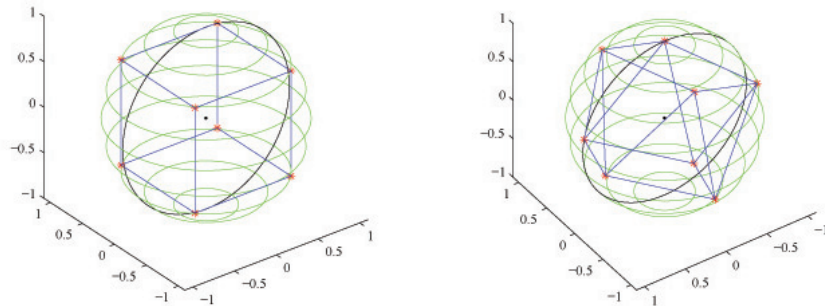


Abbildung 5: Normaler Würfel und verdrehter Würfel im Vergleich

Anders sieht die Sache aus, wenn man als Vorgabe direkt mit einem Würfel startet. Hier schafft es weder die deterministische noch die stochastische Simulation aus diesem lokalen Energieminimum auszubrechen. Bei der deterministischen Simulation besteht das Problem, dass sich die Einflüsse der anderen Punkte auf jeden einzelnen gerade aufheben, so dass der entstehende Verschiebungsvektor gerade dem Nullvektor entspricht. Ähnlich sieht es bei der stochastischen Simulation aus, egal in welche Richtung ein einzelner Punkt des Würfels verschoben wird, während der Abstand zu einer Hälfte der Punkte etwas grösser wird, wird er zu allen anderen etwas geringer.

Dieses Problem kann nur eine Kombination beider Simulationen lösen: Zunächst einmal wird die Punktekongfiguration leicht durchgeschüttelt so dass jeder Punkt an einer leicht veränderten Position zu stehen kommt. Selbst die von Matlab kleinstmögliche Veränderung reicht, damit die deterministische Simulation nach

einigen Schritten aus dem lokalen Minimum herauskommt und schliesslich ebenfalls den verdrehten Würfel als optimale Verteilung der Punkte findet.

3.2 Falsifizierung der Behauptung

Die Behauptung ist eindeutig widerlegt! Nicht alle platonischen Körper sind Musterbeispiele der Simulationen. Dies lässt sich anhand von zwei Gegenbeispielen, nämlich bei $n = 8$ und $n = 20$, beweisen. Bei $n = 8$ Punkten erhält man wie vorher beschrieben nicht einen platonischen Körper, sprich einen Würfel, sondern einen verdrehten Würfel.

Das zweite Gegenbeispiel ist der Dodekaeder. Beide Simulationen finden bei 20 Punkten als Ergebnis nicht den zugehörigen platonischen Körper, sondern ein Polyeder, bestehend aus lauter gleichschenkligen Dreiecken. Siehe auch Abbildung 6. Dieser Körper ist besonders in Bezug auf die energetische Situation sehr interessant. Mit der in Kapitel 4 beschriebenen Energiemessmethode Coulomb-Energie pro Kante stellt man fest, dass der Körper gegenüber dem Dodekaeder eine 30 % bessere Energie aufweist. Ebenso weist er ein wesentlich grösseres Volumen und auch eine grössere Oberfläche als der Dodekaeder auf.

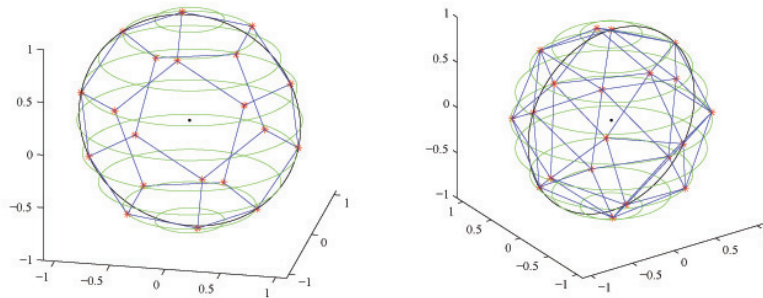


Abbildung 6: Dodekaeder und Ergebniss mit 20 Punkten im Vergleich

3.3 Weitere Resultate

Interessant ist, dass sich als Ergebnisse fast nur Körper ergeben, welche aus Dreiecken aufgebaut sind. So zum Beispiel sind auch die drei platonischen Körper Tetraeder, Oktaeder und Ikosaeder Ergebnisse der beiden Simulationen. Ein Vergleich dieser Körper findet man bei den Abbildungen 7 und 8.

Nicht zu vergessen sind auch noch die Ergebnisse bei einer ungeraden Anzahl Eckpunkten. Diese Resultate wurden der Vollständigkeit halber ebenfalls berechnet, allerdings nicht besonders analysiert. Zu sehen sind diese Körper bei den Abbildungen 9, 10 und 11. Auch diese Figuren sind aus Dreiecken aufgebaut und weisen teilweise interessante Symmetrien auf, wie zum Beispiel der Körper mit 7 Eckpunkten der achsensymmetrisch aufgebaut ist.

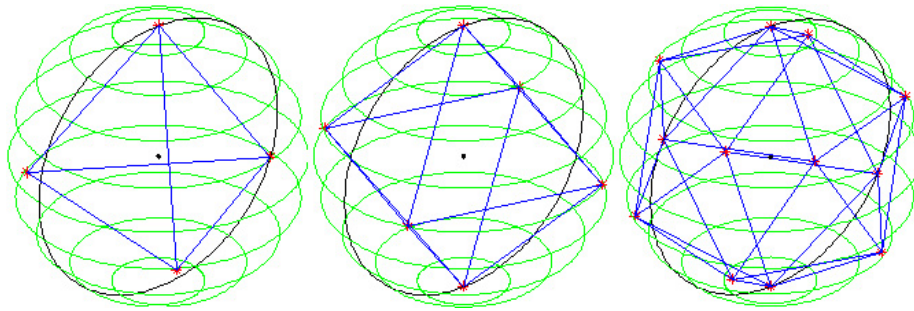


Abbildung 7: Durch Simulation entstanden: Tetraeder, Oktaeder und Ikosaeder

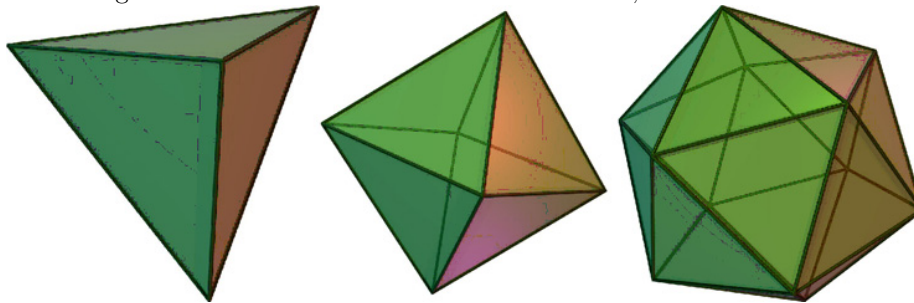


Abbildung 8: Zum Vergleich: Tetraeder, Oktaeder und Ikosaeder

3.4 Geschwindigkeit der Programme

Zu Beginn wurde erwähnt, dass bei beiden Simulationen durch eine Konstante k bestimmt werden kann, wie stark sich die Punkte während eines Optimierungsschrittes bewegen können. In diesem Kapitel soll analysiert werden, wie schnell die beiden Simulationen eigentlich an ihr Ziel, sprich an ein möglichst genaues Resultat, kommen und wie diese Geschwindigkeit von der Grösse der Konstante abhängt.

Um hierauf eine Antwort zu geben, wurden je fünf zufällige Punkteverteilungen zu 4, 6 und 12 Punkten erstellt und mit verschiedenen Konstanten und beiden Methoden so lange optimiert, bis jeder Abstand eine vorgegebene Genauigkeit erreicht hatte. Als Genauigkeit ist die Abweichung der Abstände von den Abständen des theoretischen Optimums gemeint. Gezählt wurden dabei die

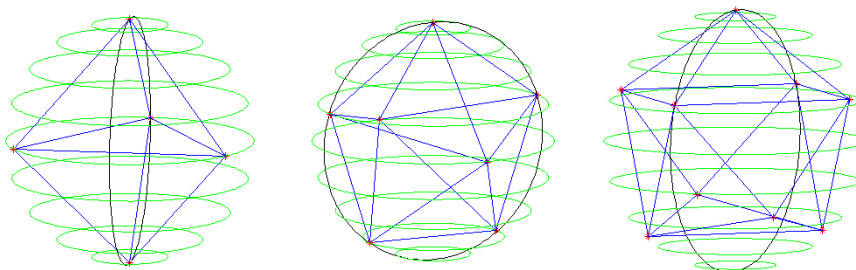


Abbildung 9: Die Ergebnisse mit $n = 5$, $n = 7$ und $n = 9$

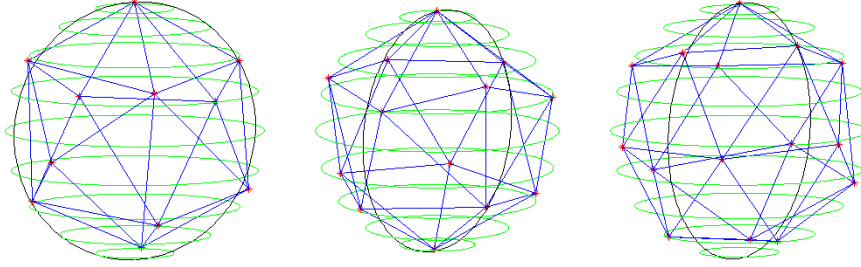


Abbildung 10: Die Ergebnisse mit $n = 11$, $n = 13$ und $n = 15$

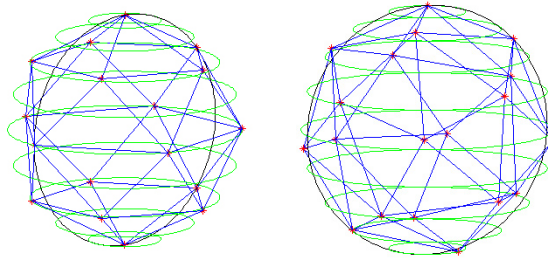


Abbildung 11: Die Ergebnisse mit $n = 17$ und $n = 19$

benötigte Anzahl Optimierungsschritte bis die vorgegebene Genauigkeit erreicht wurde.

Für mehr als 12 Punkte wäre die Berechnung sehr zeitaufwändig und komplizierter, weil die Körper unter Umständen mehrere verschiedene Abstände aufweisen könnten, weshalb bei diesem Problem nur für wenige Beispiele ein Resultat vorliegen.

3.4.1 Geschwindigkeit der deterministischen Simulation

Für die deterministische Simulation hat sich gezeigt, dass eine möglichst grosse Konstante schneller zu einem genauen Resultat führt. Tabelle 2 zeigt die durchschnittlich benötigte Anzahl von Schritten der deterministischen Simulation bei 4 Punkten für einige verschiedene Konstanten. Im Diagramm 12 sind diese Daten dargestellt, so dass die lineare Zunahme der benötigten Anzahl Schritte gut zu sehen ist. Beachten sollte man hier, dass das Diagramm bei 100 Schritten abgeschnitten ist; wie man bei der Tabelle sehen kann, werden bei einer Konstante von $1/25$ wesentlich mehr Schritte benötigt.

Mit einer Konstanten von 1000 erreicht die deterministische Simulation bereits nach durchschnittlich 6 Optimierungsschritten, nachdem also jeder Punkt 6 Mal verschoben wurde, eine Genauigkeit von 1%. Bereits nach zwei weiteren Schritten hat sich die Genauigkeit verzehnfacht und nach 15 Schritten ist das Ergebnis bereits auf 6 Stellen hinter dem Komma absolut exakt. Für eine solch grosse Konstante ist die deterministische Methode also extrem effizient.

Auch mit Konstanten grösser als 1000 wurde experimentiert und für $K = 10^6$ oder 10^{10} ergab sich dieselbe Anzahl benötigter Schritte, wie für $K = 1000$. Für

Konstante	Geschwindigkeit				
	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
$K = 1000$	6	8	10	12	15
$K = 25$	7	9	11	13	15
$K = 5$	8	10	13	16	19
$K = 1$	14	19	25	30	35
$K = 1/5$	46	64	83	102	120
$K = 1/25$	203	287	372	457	543

Tabelle 2: Anzahl Optimierungsschritte der deterministischen Simulation von 4 Punkten für verschiedene Genauigkeiten für verschiedene Konstanten

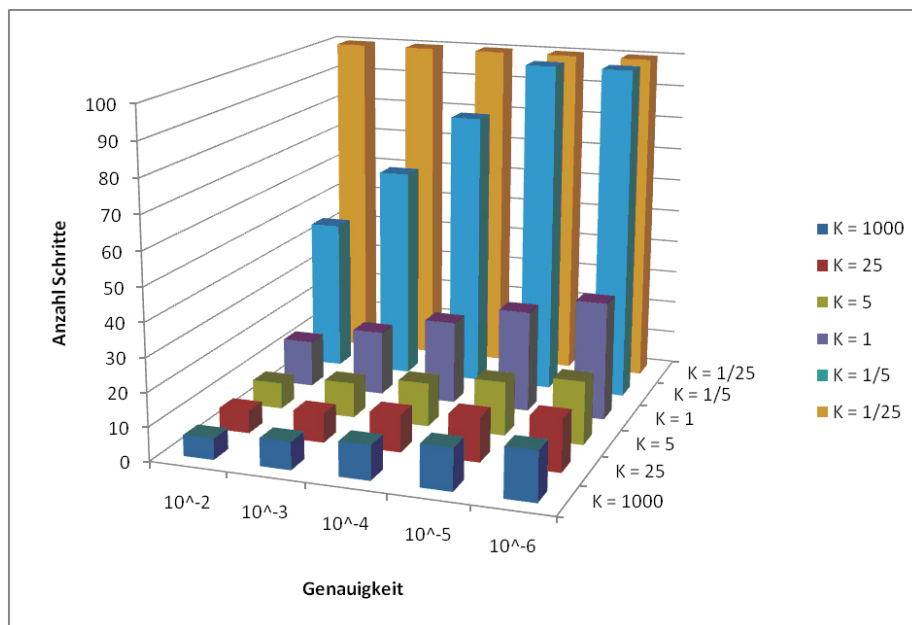


Abbildung 12: Die Daten aus Tabelle 2 graphisch dargestellt

viele Punkte sollte die Konstante aber gesenkt werden, da sonst die Bewegungen durch die extrem grossen Abstossungskräfte der Punkte zu riesig wären.

Für 12 Punkte benötigt die deterministische Simulation aber etwas länger. Tabelle 3 zeigt die benötigten Schritte für die Genauigkeiten von 10^{-2} bis 10^{-6} .

Hierbei ist eine Zunahme der benötigten Schritte zu sehen, wobei die Rechenzeit immer noch extrem gering ist. Sowohl in Tabelle 2 als auch in Tabelle 3 ist zu sehen, dass die Anzahl Optimierungsschritte proportional zum Logarithmus der Zunahme der Genauigkeit scheint. Diese Eigenschaft ist natürlich sehr nützlich, weil man dadurch für eine viel grössere Genauigkeit nicht wesentlich mehr Rechenzeit benötigt.

3.4.2 Geschwindigkeit der stochastischen Simulation

Bei der stochastischen Simulation ist die Wahl der Konstante entscheidender als bei der deterministischen. Tabelle 4 zeigt die Anzahl benötigter Optimierungs-

Genauigkeit	Genauigkeit				
	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
$K = 1000$	20	25	31	36	41

Tabelle 3: Anzahl benötigter Optimierungsschritte der deterministischen Simulation für 12 Punkte

Konstante	Geschwindigkeit				
	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
$K = 1$	647	XXX	XXX	XXX	XXX
$K = 1/5$	90	XXX	XXX	XXX	XXX
$K = 1/25$	216	344	17578	XXX	XXX
$K = 1/125$	969	1009	1604	28292	XXX
$K = 1/625$	4759	4851	4939	6604	XXX

Tabelle 4: Anzahl benötigter Optimierungsschritte der stochastischen Simulation für 4 Punkte bei unterschiedlichen Konstanten

schritte mit 4 Punkten und verschiedene Konstanten für die entsprechenden Genauigkeiten.

„XXX“ heisst, dass innerhalb von 10^5 Schritten kein genügend genaues Resultat gefunden wurde. Für 4 Punkte scheint eine Konstante in der Nähe von $1/25$ am besten geeignet um schnell eine Genauigkeit von 1% zu erreichen, mehr wäre für das Auge ohnehin kaum unterscheidbar. Sucht man jedoch eine genauere Lösung muss eine kleinere Konstante gewählt werden. Am besten wäre eine stetige Anpassung der Konstante an die Genauigkeit der momentanen Abstände zwischen den Punkten. Allgemein ist zu sehen, dass die benötigte Anzahl Optimierungsschritte, um eine bestimmte Genauigkeit zu erreichen hochgradig exponentiell verläuft. Weiter ist zu bemerken, dass die stochastische Simulation zufallsgesteuert ist und dadurch eine gewisse Streuung entsteht. Die Tabelle 5 zeigt nun die Anzahl der benötigten Optimierungsschritte bei 12 Punkten.

Interessant ist die Tatsache, dass für eine Konstante von $1/625$ und einer geringen Genauigkeit weniger Schritte benötigt wurden, als bei nur 4 Punkten. Drei Mal mehr Punkte benötigen also trotzdem etwa gleich viele Schritte. Die stochastische Simulation mag zwar einiges langsamer arbeiten als die deterministische, doch könnte sich das bei vielen Punkten und einer hohen Genauigkeit irgendwann mal ändern und die stochastische Simulation wäre effizienter. Um dies festzustellen wären aber weitere Forschungen und sehr viel Rechenkapazität notwendig.

Konstante	Geschwindigkeit				
	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
$K = 1/5$	11369	XXX	XXX	XXX	XXX
$K = 1/25$	968	56470	XXX	XXX	XXX
$K = 1/125$	1076	4071	XXX	XXX	XXX
$K = 1/625$	4637	4846	15867	XXX	XXX

Tabelle 5: Benötigte Schritte der stochastischen Simulation für 12 Punkte

4 Diskussion

Um unsere Figuren zu vergleichen ist ein gutes Kriterium notwendig. Dieses Kriterium ist ein Wert, der von der Punkteverteilung abhängt und aussagt, wie gut diese liegt, eine sogenannte Invariante.

4.1 Federenergie aller Abstände

Man nimmt an, dass zwischen allen Eckpunkten der Figur Zugfedern aufgespannt sind. Je stärker diese Federn gespannt sind, sprich je grösser die Abstände der einzelnen Punkte voneinander sind, umso grösser wird die Energie. Alle Abstände werden ins Quadrat gesetzt und aufsummiert.

Erstaunlicherweise führt diese Idee in eine Sackgasse, weil man für absolut verschiedene Figuren denselben Wert erhält. Sowohl der Wert des normalen, als auch der Wert des verdrehten Würfels und der Doppelpyramide mit 8 Punkten ergibt den Wert 64. Das ist gerade das Quadrat der Anzahl Punkte. Genau dasselbe ist auch bei anderen Körpern zu beobachten. Diese Eigenschaft nützt uns in der Gütebestimmung einer Figur also nichts, wir werden aber in Kapitel 4.6 nochmals darauf zurückkommen.

4.2 Federenergie der echten Kanten

4.2.1 Idee

Wenn man annimmt, dass die Federn nur zwischen echten Kanten der Figur gespannt werden, müssen zuerst die überflüssigen Abstände (Diagonalen) ausgeschnitten werden. Je weiter die Punkte auseinander liegen, desto angespannter sind die Federn und umso grösser die Energiewerte. Für die Berechnung erstellt das Programm `pnet.m` eine Inzidenzmatrix. Diese Matrix besteht aus 0 und 1 und beinhaltet die Codierung, welche Punkte miteinander verbunden sind (1) und welche nicht (0).

Diese Matrix wird mit der Abstandsmatrix elementweise multipliziert und es entsteht eine Matrix der Abstände der echten Kanten. Diese Matrix wird mit ihrer Transponierten multipliziert und davon die Hälfte der Spur ausgerechnet. Dieser Wert kann als Qualitätswert der zugehörigen Figur betrachtet werden. Im Grunde ist dies nichts anderes, als alle echten Kanten ins Quadrat zu setzen und aufzusummieren. Dies geschieht mit folgender Formel in Matlab, wobei `pnet(w)` die Inzidenzmatrix ist:

$$\text{trace}(\text{pnet}(w) \cdot \text{dmat}(w) \cdot (\text{pnet}(w) \cdot \text{dmat}(w))')/2$$

4.2.2 Ergebnis

Wenn die Anzahl Punkte gleichmässig steigt, steigt die genannte Energie aber sehr unregelmässig. Grund ist, dass auch die Anzahl Kanten unregelmässig zunimmt. Bei vielen Kanten werden auch viele Abstände einbezogen. Z.B. hat die Figur mit 7 Punkten die Energie 26.9 und 15 Kanten. Der verdrehte Würfel mit 16 Kanten aber nur 24.2 und die Figur mit 9 Punkten und 21 Kanten wieder mehr, nämlich 30.5.

Es fällt auf, dass der exakte Würfel mit 12 Kanten nur den Wert 18.67 hat, welcher ähnlich der Energie eines Tetraeder mit nur 4 Punkten ist, obwohl jener aus doppelt so vielen Punkten besteht.

Bei gleicher Anzahl Punkten sollte der Energiewert möglichst gross sein, denn grosse Abstände bewirken noch grössere Quadrate, also grosse Energie. Bei zunehmender Anzahl Punkten steigen die Energiewerte auch an, obwohl mehrere Punkte (Kleine Abstände $\Rightarrow$ grosse Abstossungskräfte) eigentlich in einer schlechteren Situation vorliegen, als wenn es nur wenige sind. Diese Beobachtung ist aber auf die schnell wachsende Kantenanzahl zurückzuführen. Siehe auch Abbildung 13.

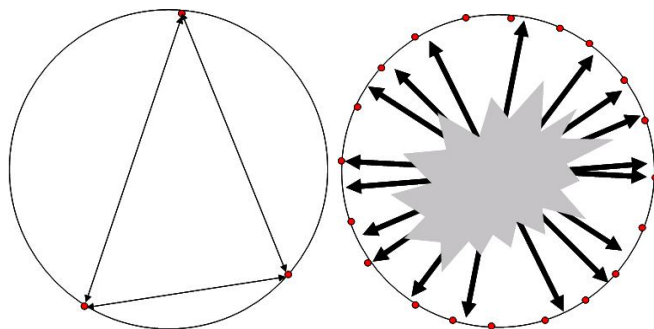


Abbildung 13: mehr Punkte $\Leftrightarrow$ kleiner Abstand $\Leftrightarrow$ grosse Abstossungskraft

4.2.3 Energie durch Kanten

Um den Einfluss der Kantenanzahl zu beseitigen kann man die Energiewerte durch die Anzahl Kanten der zugehörigen Figur teilen. Damit hat man den Wert pro Kante. Nun sieht man, dass die Energie mit zunehmender Anzahl Punkten immer kleiner wird, da die Abstände und damit ihre Quadrate kleiner werden. Ebenso wird der Wert für den exakten Würfel kleiner als der Wert des verdrehten Würfels wegen den kürzeren Kanten. Diese neuen Werte sind ein besseres Kriterium als der alte Wert. Ein paar Beispiele hierfür sieht man auf Tabelle 6.

Figur	Energie	Kanten	Energie / Kanten
4 Punkte	16	6	2.667
5 Punkte	21	9	2.333
6 Punkte	24	12	2.000
7 Punkte	26.9098	15	1.794
8 Punkte	24.2398	16	1.515
9 Punkte	30.5027	21	1.453
10 Punkte	31.9446	24	1.331
12 Punkte	33.1672	30	1.106
16 Punkte	36.2981	40	0.907
20 Punkte	38.3647	54	0.710
Würfel	16	12	1.333
Dpyr(8)	11.4853	18	0.638
Dodekaeder	15.2786	30	0.509
Dpyr(20)	29.3698	60	0.489

Tabelle 6: Federenergie der Ergebnisse

Je grösser die Werte, umso besser sind die Punkte verteilt. Das Resultat-Polyeder ist den anderen 20-Punkte-Figuren überlegen, ebenso der verdrehte Würfel dem normalen Würfel. Der Dodekaeder weist gar einen um 40 % kleineren Wert auf als das Ergebnis der Simulation für 20 Punkte.

4.3 Coulomb-Gesetz

Trotz diesen Qualitätswerten ist die angewandte Energieberechnung immer noch verbesserbar. Eigentlich wäre es logischer, wenn die Energie bei mehr Punkten ansteigen würde und nicht umgekehrt. Gerade die deterministische Simulation basiert schliesslich auf dem Prinzip, dass sich die Punkte abstossen. Wenn die Punkte also auseinanderdriften, wäre die Energie kleiner, als wenn viele Punkte auf engen Raum liegen. Man stellt sich also Druckfedern vor. Siehe Abbildung 14.

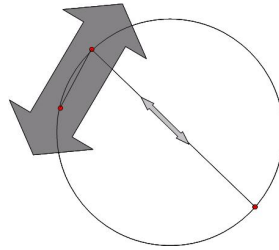


Abbildung 14: Je kleiner der Abstand der Punkte, umso stärker stossen sie sich ab.

Um einen solchen Wert zu erhalten, werden die Elemente der Abstandsmatrix der echten Kanten mit -1 potenziert. Wenn zwei Punkte nahe bei einander liegen ist der Energiewert jetzt grösser, als wenn sie weiter auseinander liegen würden. Folgende Formel berechnet in Matlab diesen Energiewert: $sum(sum(diag0(1/pnet(w) \cdot dmat(w))))$

Figur	Energie	Kanten	Energie / Kanten
4 Punkte	3.674	6	0.6124
5 Punkte	5.0751	9	0.6639
6 Punkte	8.4852	12	0.7071
7 Punkte	11.325	15	0.755
8 Punkte	13.0432	16	0.8152
9 Punkte	17.5745	21	0.8369
10 Punkte	21.0024	24	0.8751
11 Punkte	24.6996	27	0.9148
12 Punkte	28.533	30	0.9511
16 Punkte	45.549	32	1.0845
20 Punkte	64.8216	54	1.2004
Würfel	10.392	12	0.866
Dodekaeder	42.039	30	1.4013

Tabelle 7: Coulombenergie

Dieser Wert muss natürlich ebenfalls wieder durch die Anzahl Kanten der Figur geteilt werden. Die Ergebnisse sieht man auf Tabelle 7.

4.4 Oberfläche und Volumen

Auch die Werte der Oberfläche oder des Volumens eines Körpers scheinen eine geeignete Invariante zu sein. Je grösser der Wert, desto optimaler die Punkteverteilung. Nebenbei ist zu bemerken, dass sich die beiden Werte bei höheren n immer mehr der Oberfläche, bzw. des Volumens der Einheitskugel ($4 * \pi$, bzw. $4/3 * \pi$) annähern. In Tabelle 8 sind die Oberflächen und Volumina für einige Beispiele eingetragen, auch mit ihrem prozentuellen Anteil an der Einheitskugel. Der Wert der Oberfläche geteilt durch das Volumen gibt an, wieviel Oberfläche wieviel Volumen einschliesst. Für die Kugel wäre dieser Wert genau 3 und je besser sich ein Polyeder der Kugel annähert, desto weiter sinkt er gegen 3.

Punkte	Oberfläche	Oberfläche %	Volumen	Volumen %	Fläche / Volumen
4	4.6188	36.8	0.5132	12.3	9
5	7.1085	56.6	0.866	20.7	8.208
6	6.9282	55.1	1.3333	31.8	5.196
7	7.5605	60.2	1.5851	37.8	4.770
8	8.1165	64.6	1.7499	41.8	4.638
9	8.5981	68.4	2.0396	48.7	4.216
10	8.9645	71.3	2.2126	52.8	4.052
11	9.2471	73.6	2.351	56.1	3.933
12	9.5745	76.2	2.5362	60.5	3.775
16	10.259	81.6	2.8773	68.7	3.565
20	10.7007	85.2	3.101	74.0	3.451
24	11.0051	87.6	3.26	77.8	3.376
32	11.3967	90.7	3.5049	83.7	3.252
Würfel	8	63.7	1.5396	36.8	5.196
Dodekaeder	10.5146	83.7	2.7852	66.5	3.775

Tabelle 8: Oberfläche und Volumen der Körper

4.5 Sonstige Invarianten

Neben den bisher genannten Invarianten wurde auch an weiteren Ideen geforscht. Viele davon erwiesen sich als Sackgasse, so etwa die Betrachtung der Determinante verschiedener Matrixkombinationen. Während den Untersuchungen der Eigenwerte ist folgendes aufgefallen: Wenn man die Eigenwerte einer Abstandsmatrix sucht, findet man nur reelle Eigenwerte. Bis auf einen Einzigen sind sie alle negativ und öfters sind zwei bis drei davon genau gleich gross. Der letzte positive Eigenwert ist dagegen einiges grösser, also der dominante Eigenwert.

Je mehr Punkte eine Figur hat, umso grösser ist der zugehörige dominante Eigenwert. Dabei fällt auf, dass sich die Werte des Würfels und des verdrehten Würfels nicht wie bei der Energie deutlich unterscheiden, sondern extrem nahe beieinander liegen. Erst an der fünften Nachkommastelle ist der Wert des verdrehten Würfels grösser. Ebenso weichen die Werte des Dodekaeders erst an der zweiten Nachkommastelle von unserem Resultat-Polyeder ab.

Da sich die Werte mit gleich vielen Punkten nur so gering unterscheiden, ist dies kein gutes Kriterium für die Bestimmung, wie gut eine Anzahl Punkte verteilt ist. Es ist jedoch ersichtlich, dass sich der Würfel vermutlich doch in keiner extrem viel schlechteren Situation als derjenigen des verdrehten Würfel befindet.

Diese Eigenschaft wurde noch nicht gründlich untersucht, es ist möglich, dass sich noch weitere Erkenntnisse finden lassen.

4.6 das N^2 Gesetz

Die ersten Untersuchungen der Ergebnisse haben folgendes gezeigt: Wenn man sämtliche Abstände zwischen allen Punkten quadriert und aufsummiert, entspricht das Resultat der Anzahl Punkte im Quadrat. Siehe auch Kapitel 4.1.

$$\sum_{j \neq i}^n \|\vec{P}_i - \vec{P}_j\|^2 = 2n^2$$

Diese Eigenschaft gilt für alle möglichen Dimensionen, sofern der Schwerpunkt der Figur auf dem Nullpunkt liegt. Beweis:

Annahme:

$$\|\vec{Z}_k\| = 1 \quad (1)$$

$$\vec{Z}_1 = \vec{e} = [0|0|1] \quad (2)$$

$$\sum_{j=1}^n \vec{Z}_i = \vec{0} \quad (3)$$

$\vec{Z}_k$ ist der Ortsvektor zum Punkt k der Figur auf dem Einheitskreis, Punkt 1 liegt auf dem Einheitskreis und die Summe der Ortsvektoren zu den Eckpunkten der Figur ist $0 \Leftrightarrow$ Der Schwerpunkt der Figur liegt auf dem Nullpunkt.

Berechnung:

$$\sum_{k=1}^n \|\vec{Z}_k - \vec{e}\|^2 = \sum_{k=1}^n ((\vec{Z}_k - \vec{e}) \cdot (\vec{Z}_k - \vec{e})) \quad (4)$$

$$\Rightarrow \underbrace{\sum_{k=1}^n \|\vec{Z}_k\|^2}_n * \left(\underbrace{\sum_{k=1}^n (\vec{Z}_k)}_0 + \underbrace{\sum_{k=1}^n (1)}_n \right) = 2n \quad (5)$$

Die Summe der Abstände zwischen dem Nordpol ($\vec{e}$) und allen anderen $n - 1$ Punkten ergibt also $2n$. Jetzt müssen aber noch die Abstände der anderen Punkte einbezogen werden, also ist es $n * 2n = 2n^2$. Dieses Resultat muss noch durch 2 geteilt werden, weil jeder Abstand zwei Mal gerechnet wurde, nämlich von P_k zu P_i und von P_i zu P_k . Es bleibt somit n^2 .

Beweis, dass das n^2 -Gesetz genau für die Figuren stimmt, deren Schwerpunkt auf dem Nullpunkt liegen:

$$\sum_{i=1}^n \left(\sum_{k=1}^n \|\vec{Z}_k - \vec{Z}_i\|^2 \right) = 2n^2 - 2 \underbrace{\left(\sum_{i=1}^n \vec{Z}_i \right)}_{\vec{s}} \cdot \underbrace{\left(\sum_{k=1}^n \vec{Z}_k \right)}_{\vec{s}} \quad (6)$$

Zur Vereinfachung wird $\sum_{i=1}^n (\vec{Z}_i)$ mit $\vec{s}$ bezeichnet. $\|\vec{s}\|$ muss zwingend positiv sein, also schliesst man aus dem oberen Resultat:

$$2n^2 - 2\|\vec{s}\|^2 \leq 2n^2 \quad (7)$$

Das gilt für alle möglichen Figuren aus n Punkten auf der Einheitskugel. Wenn die Summe der Quadrate aller Abstände zwischen allen Punkten genau $2n^2$ ergeben soll, also $2n^2 - 2\|\vec{s}\|^2 = 2n^2$ muss s gleich 0 sein, also $\sum_{i=1}^n \vec{Z}_i = \vec{0}$

Oder, anders ausgedrückt, die Summe aller Ortsvektoren zu den Punkten muss den Nullvektor ergeben, d.h. der Schwerpunkt dieser Figur muss auf dem Nullpunkt liegen.

5 Noch offene Fragen

Die Fragestellung konnte zwar eindeutig geklärt werden, aber während dem Projekt sind laufend neue Fragen und Unklarheiten aufgetaucht und nicht alle konnten bis jetzt beantwortet werden.

5.1 Was passiert wenn n sehr gross wird?

Das Projekt befasste sich mit einem Punktespektrum von 4 bis 20, nämlich der Anzahl Eckpunkten der Platonische Körper. Theoretisch funktionieren die beiden Simulationen aber mit beliebig vielen Punkten. Eingeschränkt wird die praktische Anwendung nur durch den rasch wachsenden Zeitaufwand, um eine riesige Anzahl Punkte auf der Kugel herum zu schieben. Mit einem schnellen Computer und einigen Stunden Geduld ist eine Simulation von bis zu 1000 Punkten nach unserer Schätzung aber möglich. Ein Test mit vielen Punkten wurde bereits durchgeführt mit 60 und 120 Punkte, wobei beide nach einiger Zeit aufgezeichnet werden konnten.

5.2 Anzahl Rechenschritte für n Punkte

Noch ist es nicht gelungen die Anzahl Rechenschritte der Simulationen zu ermitteln. Natürlich kann man die Anzahl Zeilen der Programme zählen, jedoch geben diese keine wirkliche Auskunft, da jeder Matlab-Befehl ebenfalls aus weiteren Befehlen besteht, die wiederum mit einem Assembler-Code aufgebaut sind. Würde man diese Assembler-Befehle analysieren, könnte man die Rechenschritte errechnen. Damit könnte auch berechnet werden, welche der Simulationen bei hohen n schneller wäre und wieviele Rechenschritte für einen einzigen Optimierungsschritt nötig sind und wie viele Optimierungsschritte bei welcher Methode für genaue Koordinaten notwendig sind. Auf der Ebene des Assembler-Codes läge ausserdem ein grosses Optimierungspotential vor, indem man überflüssige Schritte löschen könnte.

5.3 Konstante optimieren

Sowohl in der deterministischen, als auch in der stochastischen Simulation wird eine Konstante k verwendet um die Punkteverschiebungen zu skalieren, damit die Bewegungen nicht zu stark oder zu schwach sind. Wie bereits in Kapitel 3.4 erwähnt, hat die Wahl dieser Konstante eine grosse Auswirkung auf die Geschwindigkeit der Simulationen. Für geringere oder grössere n liesse sich vermutlich eine Konstante finden, die schneller zu genaueren Koordinaten führen würde. Allgemein gesagt wäre sicher noch ein grosses Optimierungspotential in den Simulationen vorhanden, wenn sich die Konstante jeweils der gegebenen Situation anpassen könnte.

6 Schlusswort

Diese Arbeit begann am 1. März 2005 mit der Festlegung der Fragestellung, wie man Punkte auf einer Kugel optimal anordnen könnte und ob diese Resultate den Eckpunkten der Platonischen Körpern entsprechen würden. Diese Frage sollte anhand einer Simulation beantwortet werden, die wir mit dem Mathematik-Programm Matlab zu programmieren gedachten. Es erforderte einige Stunden Zeit um alle nötigen Befehle in Matlab zu finden und die Syntax zu verstehen, doch heute können wir ein Programm präsentieren, dass mehr als nur unsere Frage zu beantworten vermag. Mit einem benutzerfreundlichen GUI ist das Programm selbsterklärend und auch für einen Benutzer ohne Fachkenntnisse zu bedienen.

Was zu Beginn nach einer einfachen Fragestellung aussah, entpuppte sich schnell als ein Projekt in das man beliebig viel Zeit stecken könnte, indem man das Problem und die Lösungen beliebig weit ausbauen würde. Von einem Resultat gelangten wir zum nächsten Aspekt der weiter untersucht werden wollte und dank dem regen Gedankenaustausch mit unseren Lehrern H.R.Schneebeli und M.Speck kam unser Projekt schnell und konstant voran.

Das grösste Problem des Projektes erwies sich erst, als all die gesammelten Ergebnisse, Erfahrungen, Resultate, Matrizen und Zahlen in einem kompakten Bericht aufgelistet und erklärt werden sollten. Nebenbei musste das ganze auch noch in $\text{\LaTeX}$ -Format geschrieben werden, samt Bilder und Tabellen.

Auf jeden Fall war es ein sehr interessantes Projekt, welches unzählige neue Erfahrungen, sowohl auf mathematischer Ebene, als auch mit Matlab und anderen Programmen mit sich brachte. Wir hoffen einen guten Einblick in das Thema geboten zu haben und den mathematischen Horizont des Lesers etwas erweitert zu haben.

7 Quellen

- H. Rutishauser, „Über Punkteverteilungen auf der Kugelfläche“, Comment. Math. Helv. 1945, S. 327 - 331
- Thomas Demmig, „ $\text{\LaTeX}$ 2, Start ohne Vorwissen“, 2004 München, Markt+Technik Verlag
- Duane Hanselman und Bruce Littlefield, „The student edition of Matlab: Version 5“, 1997 New Jersey, Mathworks INT
- www.langeneggers.ch/Nuetzliches/Geometrie/dreieck.htm (Mär. 06)
- www.calc3d.com/help/gfaqu.html (Sep. 06)
- <http://mathworld.wolfram.com/SphericalCode.html> (Nov. 06)