

Konvexe Hüllen mit “qhull”

Dokumentation zur vierten Aufgabe im Modul
„Computational Geometry“

Michael Röder, Markus Hüttner

Inhaltsverzeichnis

1. Beschreibung der Aufgabe.....	3
2. Beschreibung "qhull".....	3
3. Erzeugung der Testdaten.....	3
4 Berechnung mit "qhull".....	4
5 Ergebnis und Auswertung.....	4

1. Beschreibung der Aufgabe

In vierten Praktikum soll mit dem Programm “qhull” gearbeitet werden. Es sollen Punktemengen für 2-8 Dimensionen erzeugt werden. Zudem soll für jede Dimension verschiedene Mengen an Punkten erzeugt werden. Anschließend wird zu den erzeugten Punktemengen eine konvexe Hülle berechnet werden und die benötigte Zeit ausgewertet.

2. Beschreibung “qhull”

Das Programm qhull bringt bereits ein Werkzeug zum Erzeugen von Punktemengen mit, “rbox”. Mit Hilfe von rbox können Punktemengen in verschiedenen Dimensionen erzeugt werden.

Zur Berechnung der konvexen Hülle kann das Programm “qhull” selbst benutzt werden. Als Ausgabe werden einige Parameter zur berechneten konvexen Hülle geliefert, insbesondere die Berechnungszeit.

qhull verwendet zur Berechnung der konvexen Hülle den Algorithmus “Quickhull”. Die Komplexität dieses Algorithmus verhält sich ähnlich der von QuickSort. Im besten Fall besitzt der Algorithmus eine Komplexität von $O(n * \log(n))$.

3. Erzeugung der Testdaten

Das Programm rbox kann mit Hilfe einiger Eingabe Parameter gesteuert werden. Folgende Parameter sind für das Praktikum von Relevanz:

- Anzahl der zu erstellenden Punkte. Wird als Integer auf der Kommandozeile übergeben.
- Dimension der Punktemenge. Wird in der Form Dn auf der Kommandozeile übergeben. Beispielsweise für die Dimension 4: D4
- Angabe der Bounding Box der Punktemenge. Beschreibt die maximalen Werte, positiv als auch negativ, für die erzeugten Punkte. Als Standardwert ist von rbox 0.5 vorgegeben. Die Übergabe an rbox erfolgt in der Form Bn auf der Kommandozeile. Beispielsweise für Bounding Box von 10: B10

Für alle erzeugten Punktemengen wurde eine Bounding Box von 10 gewählt.

Als Mengen für die Testdaten wurden folgende Werte gewählt (in 10^3):

- 1.
- 10
- 100
- 1000

Beispielhaft ist ein Aufruf von rbox für die Dimension 4 mit 10000 Punkten und der Bounding Box 10 dargestellt. Die erzeugte Punktemenge wird direkt in die Datei “d4_10k.dat” geschrieben.

`./qhull/rbox 10000 D4 B10 > ./testdata/d4_10k.dat`

Aus den beschriebenen Parametern, Dimension 2-8, 4 Mengen pro Dimension ergeben sich letztlich 28 Punktemengen zu denen eine konvexe Hülle berechnet wird.

4 Berechnung mit “qhull”

qhull benötigt als Eingabe Parameter die Dimension, die Anzahl der Punkte und anschließend die Punkt Koordinaten. Da das Programm rbox bereits ein Ausgabe Format erzeugt, dass qhull einlesen kann, können die erzeugten Testdateien direkt verwendet werden.

Beispielhaft wird ein Aufruf von qhull für die in Abschnitt 3 erzeugten Testdaten dargestellt.

```
./qhull/qconvex < ./testdata/d4_10k.dat >> ./results/results_4d.txt
```

Das Ergebnis der Berechnung wird in die Datei „results_4d.txt“ geschrieben.

Die Ausgabe sieht wie folgend aus:

```
Convex hull of 10000 points in 4-d:

Number of vertices: 377
Number of facets: 2089

Statistics for: rbox 10000 D4 B10 | qconvex

Number of points processed: 523
Number of hyperplanes created: 9434
Number of distance tests for qhull: 365529
CPU seconds to compute hull (after input): 0.008012
```

Neben einiger Daten zum Ergebnis der Berechnung sind vor allem die benötigten CPU Sekunden angegeben. Dieser Wert wird für die folgende Auswertung verwendet.

5 Ergebnis und Auswertung

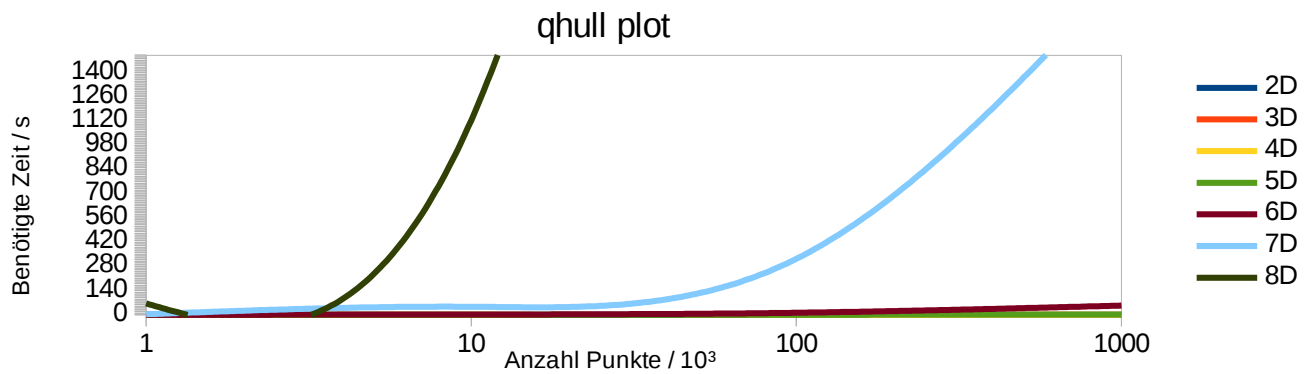
In der folgende Tabelle sind die Ergebnisse der Berechnung dargestellt.

Dimensionen	Anzahl Punkte in 10 ³			
	1	10	100	1000
2D	0,000498	0,003721	0,007211	0,07139
3D	0,001642	0,001983	0,01447	0,146
4D	0,002244	0,008012	0,04712	0,8728
5D	0,05247	0,1049	0,5466	2,528
6D	0,2577	1,99	12,1	53,41
7D	3,274	46,65	323,5	1956
8D	65,96	1123	8573	

Berechnungsdauer in Sekunden

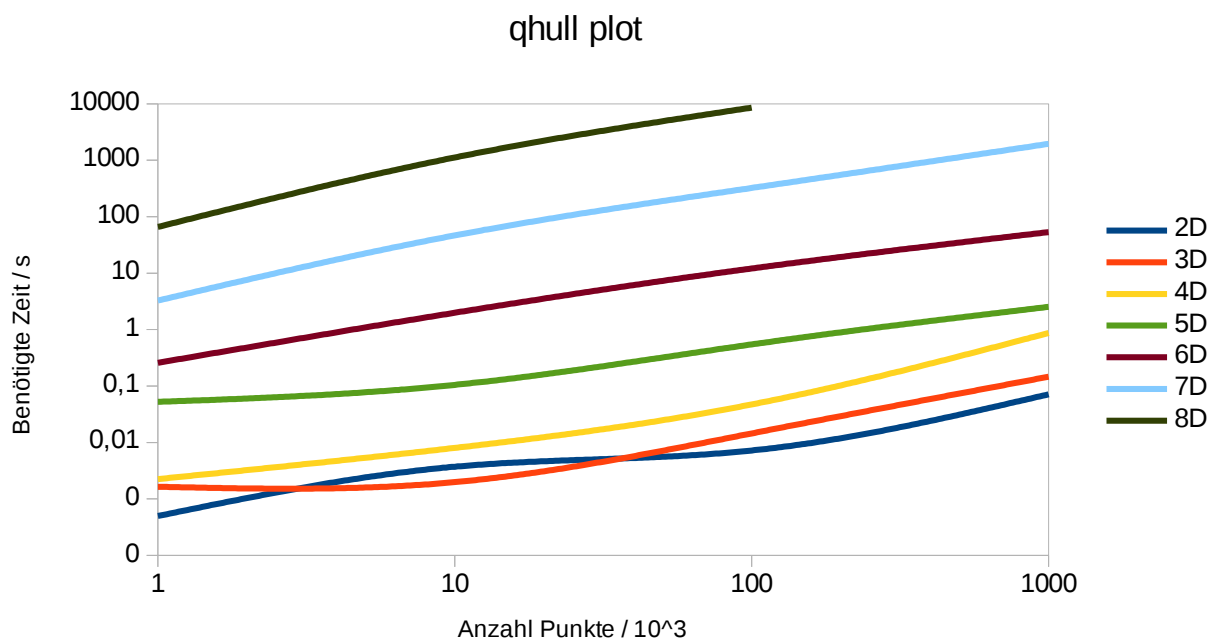
Der Wert für D8 und 1000*10³ Punkte ist nicht angegeben, da die Berechnung zu lange dauert und in der Größe für die Auswertung keine zusätzliche Aussagekraft mehr bringt.

Im folgenden Diagramm sind die jeweiligen Ergebnisse grafisch dargestellt.



Wie in diesem Diagramm zu erkennen ist, steigt mit steigender Dimension und steigender Punkteanzahl die Berechnungsdauer deutlich an. Anzumerken ist hier allerdings, dass die Kurven nicht das Zeitverhalten Widerspiegeln, da die X-Achse in logarithmischer Darstellung vorliegt, während die Y-Achse in linearer Darstellung vorliegt.

Um das Zeitverhalten noch einmal besser grafisch zu sehen, ist im folgenden Diagramm die Y-Achse ebenso in logarithmischer Darstellung dargestellt.



Aus diesem Diagramm ist die Komplexität des QuickHull Algorithmus zu erkennen. Mit steigender Dimension steigt grundsätzlich der Berechnungsaufwand, da Pro Punkt mehr Koordinaten Werte betrachtet werden müssen. Auf der logarithmischen Darstellung ergibt sich Näherungsweise eine gerade für die jeweilige Dimension, was der Komplexität $O(n * \log(n))$ entspricht.