

HOCHSCHULE MÜNCHEN
FAKULTÄT FÜR INFORMATIK UND MATHEMATIK

Praktikumsaufgabe 4

in der Vorlesung

Computational Geometry

Konvexe Hüllen mit qhull

Team:	Christopher Hinz, Tobias Gruber
Studiengruppe:	Master Informatik
Studiensemester:	1. Semester
Schwerpunkt:	Embedded Computing

19. Juni 2022

Sommersemester 2022

1 Einführung

Installieren Sie das Programm qhull, erzeugen Sie zufällige Punktmengen und berechnen Sie mit qhull konvexe Hüllen, auch in höheren Dimensionen (qhull bringt ein Werkzeug zur Erzeugung von Punktmengen mit). Plotten Sie die Zeiten für zunehmende Punktzahlen bei unterschiedlichen Dimensionen (2-8). Versuchen Sie, die Ausgaben von qhull bei "geschwätzigster" Einstellung nachzuvollziehen, zu verstehen und ggf. mit Inhalten dieser Lehrveranstaltung in Einklang zu bringen.

2 Benötigte Funktionen von qhull

Qhull bietet 6 verschiedene Programme an:

- qconvex: convex hulls
- qdelaunay: Delaunay triangulations and furthest-site Delaunay triangulations
- qhalf: halfspace intersections about a point
- qhull: all structures with additional options
- qvoronoi: Voronoi diagrams and furthest-site Voronoi diagrams
- rbox: generate point distributions for qhull

Im Zuge dieses Praktikums werden wir sowohl rbox, zur Erzeugung von Punktmengen nutzen, als auch qconvex, zur Berechnen der konvexe Hüllen, nutzen.

2.1 rbox (<http://www.qhull.org/html/rbox.htm>)

Das Programm rbox generiert zufällige oder reguläre Punkte. Standardmäßig innerhalb eines Würfels (es sei denn die Optionen 's', 'x', oder 'y' werden übergeben). Einige Beispiele zur Erzeugung von Punktmengen sind:

- rbox 10 D3: erzeugt 10 Punkte in 3D
- rbox 15 D4: erzeugt 15 Punkte in 4D
- rbox 10 D2: erzeugt 10 Punkte auf einem 2D Kreis
- rbox 100 W0: erzeugt 100 Punkte auf der Oberfläche eines Würfels

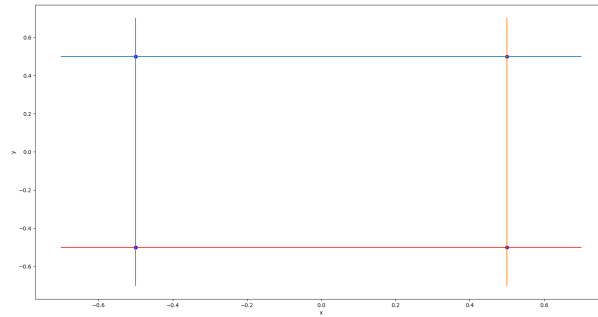
2.2 Beispiele

Bevor auf Details der Berechnungen von qconvex eingegangen wird, sollen zwei kurze Beispiele die Eingabedaten und die Ergebnisse der Berechnung visualisieren.

Zur Demonstration der Funktion von qconvex sollen selbstdefinierte Daten und deren konvexe Hülle visualisiert werden. Hierzu wurde ein Python-Skript implementiert das die Datenpunkte plottet und aus den Ergebnissen von qconvex (den sogenannten normals) die Hyperebenen zeichnet.

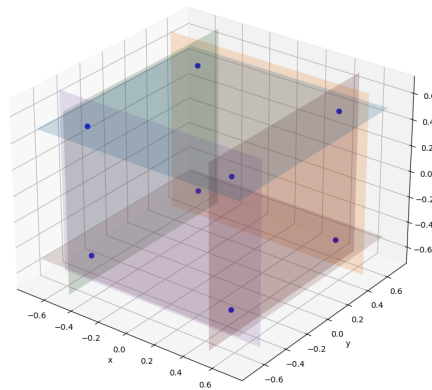
Punkte und konvexe Hülle in 2 Dimensionen

Im erste Beispiel werden die Datenpunkte eines Quadrats eingelesen und deren Hyperebenen (Geraden) mit *qconvex* berechnet. Dabei liefert *qconvex* als Ergebnis die Parameter der Geradengleichung $a * x + b * y + c = 0$ zurück. Hieraus können über die Geradengleichung die Hyperebenen berechnet und geplottet werden. Visualisiert mit *matplotlib* sieht das Ergebnis folgendermaßen aus:



Punkte und konvexe Hülle in 3 Dimensionen

Im zweiten Beispiel werden die Datenpunkte eines Würfels eingelesen und ebenfalls die Hyperebenen (Ebenen) mit *qconvex* berechnet. In diesem Fall werden die Parameter der Ebenengleichung $a * x + b * y + c * z + d = 0$ als Ergebnis zurückgeliefert. Aus diesen Parametern lässt sich die Ebenengleichung berechnen und plotten. Auch dies ist untenstehend mit *matplotlib* visualisiert:



2.3 **qconvex** (<http://www.qhull.org/html/qconvex.htm>)

Konvexe Hülle berechnen:

- `qconvex s: s = print summary`

Punktmengen erzeugen und Konvexe Hülle plotten:

- `rbox 10 D3 — qconvex s: 10 Punkte in 3D und konvexe Hülle`

TODO:

- Source code: <https://github.com/qhull/qhull>

3 Ergebnisse

4 Verifikation und Validierung