

## Aufgabenblatt 5

mpgi4@cg.tu-berlin.de

WiSe 2013/2014

### Allgemeine Hinweise:

- Die Aufgaben sollen in Gruppen bestehend aus zwei bis drei Personen bearbeitet werden. Ausnahmen müssen mit dem jeweiligen Tutor abgesprochen werden.
- Bitte reichen Sie Ihre Lösungen in Form einer ZIP Datei bis **Sonntag, den 02.02.2014, um 23:55 Uhr** auf der ISIS Webseite der Vorlesung ein. Wählen Sie dazu für Ihre Gruppe eine/n Sprecher/in, welche/r die Abgabe durchführt, und tragen Sie in jede abgegebene Datei Name und Email-Adresse aller Gruppenmitglieder ein.
- Wenn eine Aufgabe die Abgabe einer Grafik verlangt, dann muss ein vollständig funktionsfähiges Programm in der Lösung enthalten sein, welches bei der Ausführungen die Grafik erstellt. Wenn zum Erstellen einer Grafik bereits eine Funktion gegeben ist, dann darf diese nicht verändert werden.
- Tragen Sie in die Datei README.txt Name und Email-Adresse aller Gruppenmitglieder sowie einige kurze Kommentare zu Ihrer Abgabe ein. Vermerken Sie insbesondere Fehler oder Unvollständigkeiten in Ihrer Abgabe.

### Aufgabe 1: Raycasting von impliziten Funktionen (6.0 Punkte)

In der Vorlesung wurde vorgestellt, wie implizite Oberflächen, welche als Nullstellen von Funktionen beschrieben sind, mittels Strahlverfolgung visualisiert werden können. Im Beispielcode zu dieser Aufgabe ist eine einfache Variante dieses Verfahrens implementiert (siehe `raycaster.py`). Die Nullstellberechnung erfolgt in `surface.py` mit der Methode `scipy.optimize.newton`:

```
def find_root(func, dfunc):  
    try:  
        return scipy.optimize.newton(func, 1.0, dfunc)  
    except RuntimeError:  
        return None
```

Als Parameter erhält die Funktion `find_root` in `func` eine 1-dimensionale Funktion und in `dfunc` deren erste Ableitung übergeben.

- a) Ersetzen Sie die Funktion `find_root` durch eine eigene Implementierung, welche das Newton-Verfahren verwendet.
- b) Erweitern Sie ihre Implementierung der Funktion `find_root`, so dass immer die größte Nullstelle von `func` im Intervall  $[-1, 1]$  gefunden wird (sofern eine solche existiert). Geben Sie das Ergebnis in Form einer PDF Datei ab.

### Aufgabe 2: Bonus: Darstellung komplexer Szenen (3.0 Punkte)

Erweitern Sie die Strahlenverfolgung, so dass "komplexe" Szenen dargestellt werden können. Demonstrieren Sie Ihre Implementierung mit einer Szene mit mindestens drei Objekten. Reichen Sie ein Bild der Szene, welche mit Ihrer Implementierung erzeugt wurde, ein.

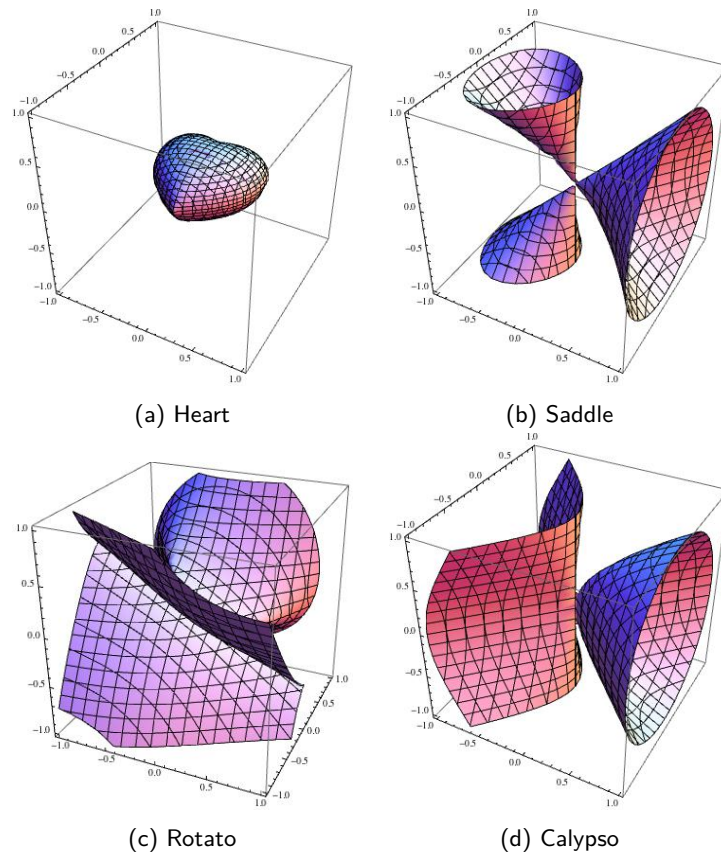


Abbildung 1: Visualisierung der Oberflächen berechnet mit Mathematica.

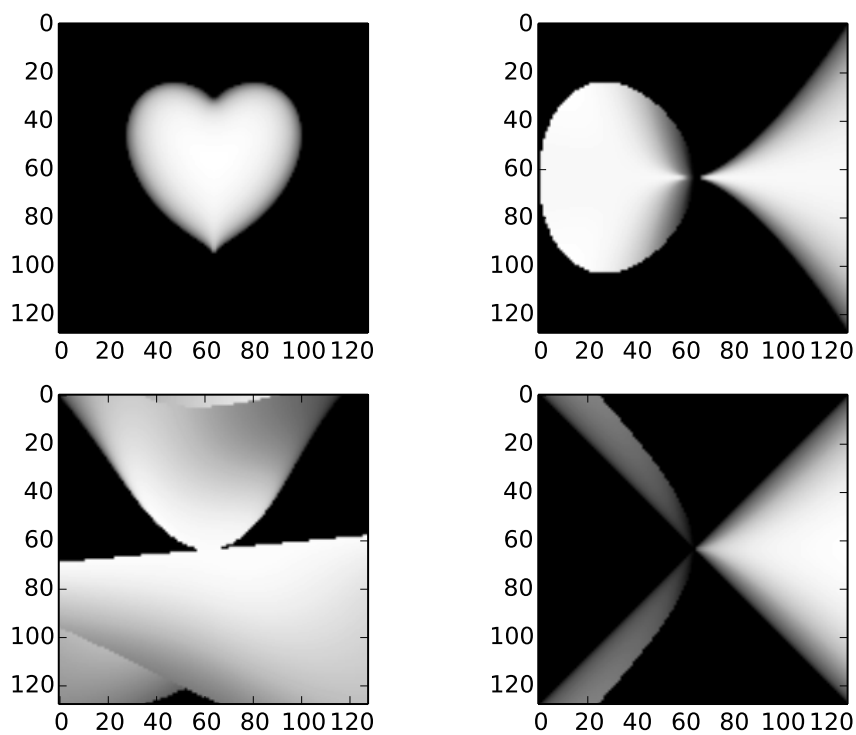


Abbildung 2: Von raycaster.py erzeugte Ausgabe.