Institut für Theoretische Physik
der Universität zu Köln

Prof. Dr. Simon Trebst
Michael Becker, Peter Bröcker

Computerphysik

Übungsblatt 2

SS 2013

Website: http://www.thp.uni-koeln.de/trebst/Lectures/2013-CompPhys.html
Abgabedatum: 30. April 2013

Eine häufig wiederkehrende Aufgabe ist die **Bestimmung von Nullstellen** einer beliebigen Funktion. Um dies zu tun, haben Sie in der Vorlesung einige iterative Verfahren kennengelernt, die wir auf diesem Übungszettel implementieren und vergleichen wollen. In der Bonusaufgabe sehen Sie außerdem, wie die simple Suche nach Nullstellen ohne viel Aufwand auf fraktale Strukturen führen kann.

9. Divide and Conquer *5 Punkte*

Als erste Methode soll eine **Grid Search** implementiert werden. Dazu wird ein Stück des Definitionsbereich um die Nullstelle herum in kleinere Bereiche unterteilt. Als nächstes wird der Bereich identifiziert, in dem die Nullstelle liegt und dieser wieder unterteilt. Wenn die gewünschte Präzision erreicht ist, wird das Verfahren abgebrochen.

Alternativ dazu kann ein Verfahren angewendet werden, das Sie bereits in der Mathematik als Intervallschachtelung kennengelernt haben und auch als **Binary Search** bekannt ist. Nachdem zwei Endpunkte gewählt wurden, wird der Bereich so halbiert, dass die Nullstelle wieder im Bereich liegt. Wie zuvor wird auch dieser Vorgang dann abgebrochen, wenn die gewünschte Genauigkeit erreicht ist.

In dieser und der nächsten Aufgabe soll eine Funktion untersucht werden, die Sie in der Quantenmechanik kennenlernen werden oder bereits kennen gelernt haben. Bei der Lösung des Problems eines Teilchens in einem endlichen Potentialtopf trifft man auf die folgende Gleichung:

$$k\tan(ka) - \kappa = 0$$

Verwenden Sie die oben beschriebenen Verfahren um die Nullstelle der obigen Funktion zu bestimmen. Setzen Sie dazu $a, \kappa = 1$.

10. Newton Verfahren *5 Punkte*

In der Vorlesung haben Sie das **Newton Verfahren** kennengelernt. Dieses soll nun ebenfalls angewendet werden, um die Nullstelle der oben genannten Funktion zu bestimmen. Implementieren Sie den Newtonalgorithmus und bestimmen Sie die Nullstelle für Startpunkte im Intervall $[0, 100]$. Tragen Sie dann die Werte der gefundenen Nullstellen gegen die Startpunkte auf.

11. Newton-Fraktale

optionale Aufgabe - 6 Punkte

Das Newtonverfahren lässt sich problemlos auch auf **komplexe Funktionen** erweitern, die eine vielfältige Struktur an Nullstellen aufweisen können. Insbesondere kann es Punkte geben, an denen der Algorithmus nicht konvergiert, ganz gleich wie die Parameter gewählt werden.

Betrachten Sie eine komplexes Polynom Ihrer Wahl und bestimmen Sie die Nullstellen. Speichern Sie die Anzahl der benötigten Iterationsschritte für Punkte aus der Menge $[-2,2] \times [-2\mathrm{i}, 2\mathrm{i}]$ in einer *numpy*-Matrix und stellen Sie diese anschließend mit *imshow* dar.

12. Mandelbrot- und Julia-Mengen

Zum Anschauen

Fraktale sind faszinierende Objekte der Mathematik, die oft, aber nicht immer, selbst-ähnliche Struktur besitzen. Zwei der bekanntesten Fraktale sind die **Mandelbrot-Menge**, benannt nach Benoît Mandelbrot, der zugleich auch den Begriff Fraktal prägte, und die **Julia -Mengen**.

Die Mandelbrot-Menge wird erzeugt indem man die komplexe Funktion $f(z) = z^2 + c$ iteriert, d.h. $z_{n+1} = z_n^2 + c$ berechnet. Die Zahl c gehört zur Mandelbrotmenge, wenn die Funktion mit $z_0 = 0$ nicht divergiert. Grafisch darstellen kann man die Menge, indem man die Anzahl der Schritte berechnet, während derer die Funktion unterhalb einer bestimmten Zahl bleibt und die Schrittzahl dann mit einem bestimmten Farbwert assoziiert.

Julia-Mengen lassen sich für eine größere Klasse komplexer Funktionen definieren und zeigen nicht immer fraktale Eigenschaften. Iteriert man eine solche komplexe Funktion, dann gibt die Julia-Menge Auskunft über die Fixpunktstruktur, das heißt Punkte zu denen die Iteration bei unendlich vielen Schritten konvergiert. Interessant ist dies besonders für die Bereiche in der komplexen Zahlenebene, in denen dieser "Fluss" zu einem Fixpunkt stark von den Anfangsbedingungen abhängt, was dann auf fraktale Strukturen führen kann. Die Gleichung $f(z) = z^2$ besitzt zum Beispiel die Fixpunkte $\{0, \infty\}$. Wählt man ein z_0 auf dem Einheitskreis, so konvergiert die Gleichung nicht. Für ein z_0, dessen Betrag aber nur minimal kleiner oder größer als 1 ist, konvergiert die Iteration gegen einen der beiden Fixpunkte.

In den Beispielprogrammen mandelbrot.py und julia.py untersuchen wir die Funktion $f(z) = z^2 + c$. Die Mandelbrot-Menge wird mit fester Auflösung berechnet, die mit der Variable *resolution* eingestellt werden kann. Die Julia-Menge hingegen wird iterativ für ein immer feineres Raster berechnet. Die Variable *resolution* kontrolliert in diesem Falle die Zielauflösung. Anhand der Programme können Sie noch einmal sehr schön sehen, wie einfach der Umgang mit komplexen Zahlen und simplen Animationen in Python ist.

Ein guter Startpunkt für weitere Informationen sind die Wikipedia-Einträge sowohl auf Deutsch als auch auf Englisch zu Fraktalen (de, en), Mandelbrot-Menge (de, en) und Julia-Mengen (de, en).