

**Programmierung für Naturwissenschaften 1  
Wintersemester 2019/2020  
Übungen zur Vorlesung: Ausgabe am 29.01.2020**

...und immer wieder am Mittwoch verfügbar:

<https://feedback.informatik.uni-hamburg.de/PfN1/wise2019-2020>



Die Umfrage hat ergeben, dass der 13.02.2020 um 12:00 Uhr ein favorisierter Termin für die Diskussion zur Fragenliste für die Vorbereitung zur PfN1-Klausur ist. Die Diskussion findet an diesem Termin in Hörsaal C, Chemie statt.

Alle Studierenden, die regelmäßig zur Übung erschienen sind, haben die Studienleistung für das Modul PfN1 erfüllt. Es ist daher nicht notwendig, dass Sie die Lösung zu dieser Übung abgeben, können das aber gerne bis zum 28.2.2020 tun. Sie sollten in jedem Fall die Übung am 29.1.2020 nutzen und sich mit den Übungsaufgaben beschäftigen, da Sie z.B. wichtige Konzepte lernen können, wie man einfache geometrische Objekte zeichnet.

Einige Studierende haben Interesse an den Abschnitten aus den Folien, die in der Vorlesung nicht behandelt werden konnten, gezeigt. Daher präsentiere ich am 31.3.2020 ab 10:00 Uhr solche ausgewählten Abschnitte. Alle interessierten Studierenden sind eingeladen. Die Präsentation findet voraussichtlich in Raum 16, ZBH statt.

**Aufgabe 13.1** (2 Punkte) Aus Zeitgründen können wir in der Vorlesung einen Teil des Themas „Numerische Integration“ (Folien Seite 580–585) nicht behandeln. Um diesen Abschnitt geht es in dieser Aufgabe. Dazu müssen sich einige Studierende vor der Übung anhand der Vorlesungsfolien auf den Inhalt der genannten Folien vorbereiten und das erworbene Wissen in Kleingruppen in den ersten 15-20 Minuten am Anfang der Übung an die anderen Studierenden weitergeben.

Die durch den Zufallsgenerator bestimmten Kleingruppen setzen sich wie folgt zusammen:

- |                                  |                                       |
|----------------------------------|---------------------------------------|
| 1. Ehlers, Flotow, Paulsen       | 10. Moeller, Carlsen, Gruber-Roet     |
| 2. Hauschild, Kaemmle, Jochens   | 11. Gubernator, LFranke, Gruetzmacher |
| 3. Stahl, Jegminat, Molkentin    | 12. Biegemann, Kuhn, Harkov, Eckmann  |
| 4. Schenk, Franke, Loewenberg    | 13. Witte, Kaether, Lindemann         |
| 5. Podolskiy, Jakobi, Myronovych | 14. Block, Pluemer, Rahlf             |
| 6. Grosse, Breiholz, Schuett     | 15. Scheu, Klemm, Liessmann           |
| 7. Plesch, Dao, David            | 16. Fender, Ebbing, Radtke            |
| 8. Lehmann, Breker, Tang         | 17. Paffenholz, Lohmann, Froechtling  |
| 9. Leege, Scheele, Quante, Kniep |                                       |

Die Namen der Studierenden, die sich auf das Thema vorbereiten müssen, sind jeweils am Beginn

jeder Zeile aufgeführt. Falls jemand von diesen Studierenden nicht zur Übung erscheint, verteilen sich die übrigen Mitglieder der Kleingruppe auf die anderen Gruppen.

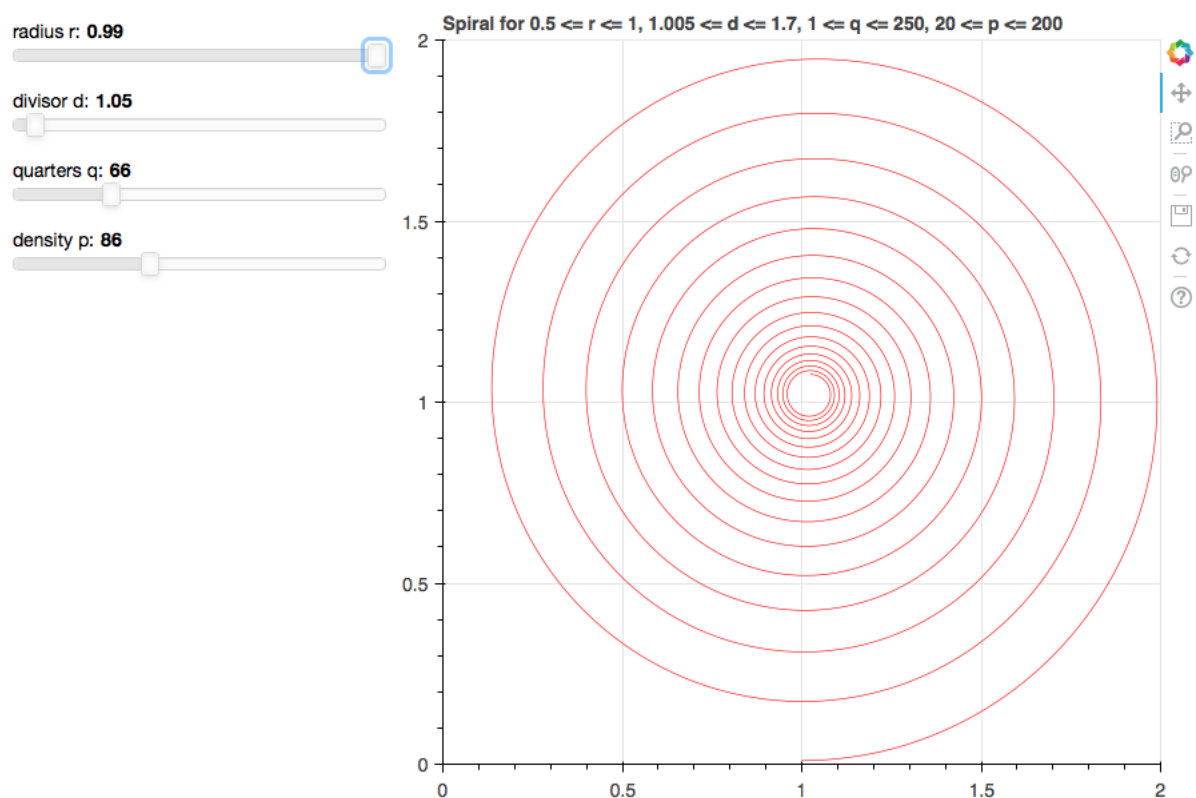
Nach der Übung dokumentiert jede Kleingruppe in einer E-mail an `kurtz@zbh.uni-hamburg.de` das Vorgehen bei der Erarbeitung des Themas und ggf. noch bestehende Verständnisfragen oder Hinweise zu Unklarheiten in den Folien. Willkommen sind natürlich auch Bemerkungen zur Lehrform selbst und Fragen zum Inhalt. Es soll aber nicht der Inhalt der Folien wiedergegeben werden. Die E-mail soll die folgenden Eigenschaften haben:

- abgesendet bis zum 31.1.2020
- maximal 10 Zeilen mit maximal 80 Zeichen pro Zeile,
- Angabe der Nachnamen aller Mitglieder der Kleingruppe am Anfang der Email in der Notation:

Kleingruppe: Name1, Name2, Name3

Die E-mail soll von einer/einem Studierenden erstellt werden, die/der sich nicht auf das Thema vorbereitet hat.

**Aufgabe 132** (13 Punkte) In dieser Aufgabe geht es darum, ein interaktives Python-Programm zu analysieren und zu dokumentieren. Das Programm benutzt Bokeh und zeichnet im Browser entsprechend der Einstellung von vier Schiebereglern interaktiv eine Spirale. Hier ist ein Screenshot:



Die grundlegenden Techniken für das interaktive Plotten mit Bokeh haben Sie in der Vorlesung bereits kennengelernt. Diese Techniken werden hier in leicht verallgemeinerter Form wiederverwendet, um eine Spirale statt z.B. eine linearen Funktion, zu plotten. Im Folgenden finden Sie eine Beschreibung der Teilaufgaben.

# 1 Ausprobieren der Bokeh-App

In den Materialien zu dieser Aufgabe finden Sie das Programm `ispiral.py`. Damit Sie es verwenden können, brauchen Sie eine Installation von Bokeh, die z.B. in der `anaconda3`-Distribution von Python3 enthalten ist. Diese finden Sie auf Rechnern im ZBH-Pool im Verzeichnis `/usr/local/zbhtools/anaconda/bin`. Sie müssen also in der Datei `.bashrc` in Ihrem Home-Verzeichnis eine Zeile

```
export PATH="/usr/local/zbhtools/anaconda/bin:${PATH}"
```

einfügen und dann im Terminal den Befehl `source .bashrc` ausführen, also Punkt Leerzeichen Punkt `bashrc`. Wenn Sie auf Ihrem eigenen Rechner Bokeh nicht verfügbar haben, dann können Sie es durch den folgenden Befehl installieren: `pip3 install bokeh`. Sie brauchen dazu aber Administrator-Rechte.

Falls Sie Probleme mit der Installation von Bokeh haben, dann lassen Sie diese Teilaufgabe aus.

Starten Sie `bokeh serve ispiral.py` im Terminal. Dann meldet sich der Bokeh Server mit dem Hinweis, welche Applikation gestartet wurde. Nachdem Sie in Ihrem Internetbrowser die URL `http://localhost:5006/ispiral` angegeben haben, wird der durch Bokeh aus Ihrem Python-Programm generierte JavaScript-Code im Browser ausgeführt.

Probieren Sie die drei Schieberegler aus und beschreiben Sie, welche Effekte sie auf die gezeichnete Spirale haben.

1 Punkt

## 2 Zeichnen von Spiralen: Die Theorie

Um das Programm `ispiral.py` zu verstehen, müssen Sie sich zunächst damit beschäftigen, wie man eine Spirale zeichnen kann. Informationen dazu finden Sie in der Datei `draw_spiral.pdf` in den Materialien. Arbeiten Sie die entsprechenden Folien alleine oder in Ihrer Gruppe durch. Formulieren Sie ggf. Fragen zum Inhalt.

## 3 Zeichnen von Spiralen: Die Implementierung

Das in `draw_spiral.pdf` beschriebene Verfahren ist in der Datei `spiral_coords.py` implementiert. Versuchen Sie, die einzelnen Klassen und Funktionen zu verstehen.

- Geben Sie für die Klasse `Quarter` jeweils an, welche Werte die Member-Variablen speichern, und wozu die einzelnen Methoden dienen. 2 Punkte
- Beschreiben Sie jeweils den Zweck und die Funktionsweise der einzelnen Funktionen außerhalb der Klasse `Quarter`. Die Funktionen `parse_arguments` und `main` brauchen Sie nicht zu dokumentieren. 4 Punkte
- Beantworten Sie die folgenden Fragen:
  - Warum ist der Rückgabewert der Funktion `num_points_get` abhängig vom Parameter `radius`? 1 Punkt

- Welche speziellen Notationen bzw. Eigenschaften von `numpy`-arrays werden in der Funktion `quarter_circle_add` verwendet?

1 Punkt

`spiral_coords.py` wird in `ispiral.py` verwendet, kann aber auch direkt im Terminal aufgerufen werden, so dass Sie mit temporären `print`-Anweisungen Zwischenergebnisse ausgeben können. Die Ausgabe von `spiral_coords.py` kann unabhängig zum Plotten verwendet werden. Hierfür steht Ihnen ein minimales Programm `plt_plot.py` in den Materialien zur Verfügung.

## 4 Von InteractivePlot zu InteractivePlot2

`ispiral.py` basiert auf der Klasse `InteractivePlot2`, die etwas allgemeiner ist als die Klasse `InteractivePlot`. Letztere haben Sie in der Vorlesung kennengelernt. Die Implementierungen dieser Klassen finden Sie in den Dateien `interactive_plot.py` und `interactive_plot2.py` in den Materialien. Schauen Sie sich bitte in den Vorlesungsfolien den Abschnitt über das interaktive Plotten an. Lassen Sie sich dann durch den Aufruf von

```
diff interactive_plot.py interactive_plot2.py
```

im Terminal die Unterschiede in der Implementierung anzeigen. Zeilen, die in der ersten Datei vorkommen, in der zweiten aber nicht, werden durch das Zeichen `<` am Anfang der Zeile gekennzeichnet. Zeilen, die in der zweiten Datei vorkommen, in der ersten aber nicht, werden durch das Zeichen `>` am Anfang der Zeile gekennzeichnet.

Beschreiben Sie die wesentlichen Unterschiede der beiden Klassen und erklären Sie, warum Verallgemeinerungen in `InteractivePlot2` sinnvoll sind.

3 Punkte

## 5 Funktionweise von ispiral.py

Nachdem nun die einzelnen von `ispiral.py` importierten Module dokumentiert wurden, beschreiben Sie kurz, wie diese in `ispiral.py` genutzt werden.

1 Punkt