

Dokumentation Serie 1 – Teil 1

Arsen Hnatiuk,
Max Huneslagen

25. Oktober 2018

Inhaltsverzeichnis

1 Schnittstellendokumentation plotten.py	1
1.1 Attribute	1
2 Bedienungsanleitung zum Hauptprogramm	1
2.1 fehlerplot()	2
2.2 Beispieldurchlauf	3

1 Einleitung

Aus der Theorie der Taylorentwicklung kann man ein günstiges Verfahren zur Approximation der ersten und zweiten Ableitungen einer Funktion ableiten. Dieses Verfahren nennt man die Methode der Vorwärtsdifferenz. Dieses Programm erlaubt eine numerische Analyse dieses Verfahrens. Insbesondere dient dieses Programm zur Berechnung des absoluten Fehlers dieses Verfahrens und zur Veranschaulichung der Beziehung zwischen diesem Fehler und anderen Parametern.

2 Schnittstellendokumentation plotten.py

Diese Klasse erlaubt das Plotten von Funktionen und ihren Ableitungen. Die Ableitungen werden dabei approximiert oder können als exakte Funktionen übergeben werden.

2.1 Attribute

nicht-statische Attribute:

`plotbereich` (*pyplot.Axes-Objekt*) The operating system, that is currently run on the computer. Should be one of the values 'MAC', 'WINDOWS' or 'LINUX'

`unten` (*float*) Untere Intervallgrenze.

`oben` (*float*)
Obere Intervallgrenze.

3 Bedienungsanleitung zum Hauptprogramm

In der Datei `hauptprogramm.py` wird eine `main()` Methode implementiert. In dieser Methode werden alle Objekte erstellt, die später zur graphischen Darstellung des Fehlerplots dienen. Es werden auch die in der `differenzieren` Klasse definierten Methoden mittels der Beispielfunktion (Sinus) veranschaulicht.

Zuerst wird der Benutzer gefordert, eine Schrittweite einzugeben. Es muss eine beliebige nichtnegative `float` Zahl eingegeben werden. Die Gültigkeit des Eingabetyps wird durch `exceptions` behandelt, und danach durch eine `if` Schleife auf Positivität geprüft. Eine nicht gültige Eingabe wird nicht berücksichtigt und der Benutzer wird erneut gefordert, eine Eingabe zu machen.

Diese Kontrolle erfolgt auf die folgende Weise:

```
abbr = 1
while abbr == 1:
    try:
        h_test = float(input('Mit welcher Schrittweite wollen Sie die ' +
            'Ableitungen approximieren?\n' +
            'Schreiben Sie bitte eine echt positive Zahl, z.B. 0.1\n'))
        abbr = 0
    except ValueError:
        print('Nicht gueltiger Wert eingegeben. Versuchen Sie erneut.')
        abbr = 1
    if h_test <= 0:
        print('Nicht gueltiger Wert eingegeben. Versuchen Sie erneut.')
        abbr = 1
```

Danach wird die Beispielfunktion (die Sinus Funktion) bearbeitet. Es werden Die `subplots` und die `arrays` erzeugt, auf welchen die Abbildungen geplottet und entsprechend evaluiert und approximiert werden. Ein Objekt der `differenzieren` Klasse für die Sinus Abbildung wird auch erzeugt. Schließlich wird die Sinus Abbildung samt seinen zwei ersten Ableitungen und der Approximation (für die durch den Benutzer eingegebene Schrittweite) dieser Ableitungen mittels der in der `differenzieren` Klasse definierten Methoden graphisch dargestellt. Der absolute Fehler der Approximation jeder von den zwei Ableitungen wird dann im Terminal ausgedruckt.

Anschließend wird die Funktion `fehlerplot()` aufgerufen, um die Beziehung zwischen dem absoluten Fehler und der Schrittweite zu veranschaulichen.

3.1 fehlerplot()

In dieser Funktion wird ein Plot des absoluten Fehlers in Abhängigkeit von der Schrittweite mittels der Differenzieren Klasse gezeichnet. Diese Funktion benutzt insbesondere die `np.vectorize` Methode, um Arrays mit den Fehlerwerten zu erzeugen, und die `.loglog` Methode, um eine doppelt-logarithmisch skalierte Graphik zu erstellen.

Input

`plotbereich` (*pyplot.Axes-Objekt*) Subplot, auf dem das Plot erzeugt wird

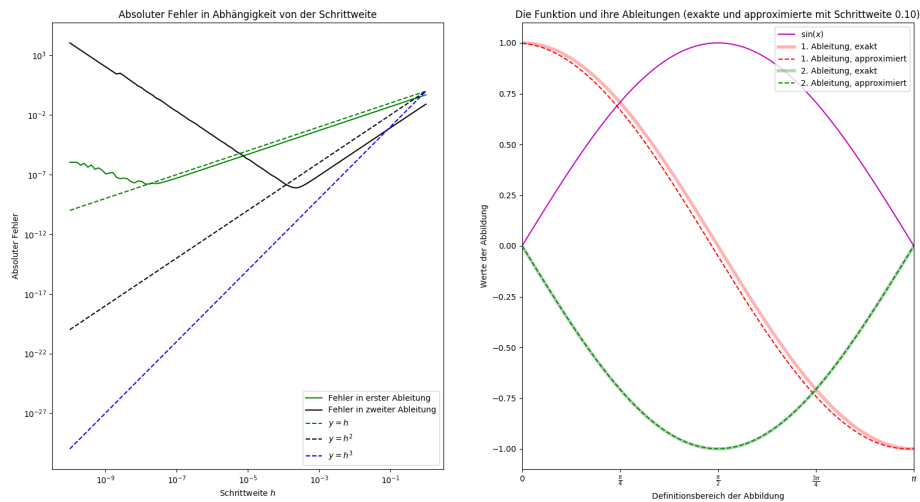


Abbildung 1: Die ausgegebene Graphik

`diff_object` (*Differenzieren-Instanz*) Differenzieren-Obejekt, das untersucht wird
`h_array` (*numpy.ndarray aus floats*) Array mit den Schrittweiten

Output Diese Funktion hat keinen Output

3.2 Beispieldurchlauf

Hier is ein Beispieldurchlauf. Der eingegebene Wert für die Schrittweite ist 0.1.

Mit welcher Schrittweite wollen Sie die Ableitungen approximieren?

Schreiben Sie bitte eine echt positive Zahl, z.B. 0.1

0.1

Der absolute Fehler in der ersten Ableitung ist 0.04999, Schrittweite 0.10

Der absolute Fehler in der zweiten Ableitung ist 0.00083, Schrittweite 0.10

Es werden dazu zwei Graphiken erzeugt (Abbildung 1). Man sieht auf der linken Seite eine Graphik mit dem absoluten Fehler als Funktion von der Schrittweite. Auf der rechten Graphik steht die Beispielfunktion mit ihren Ableitungen (exakt und approximiert).