

Semesterarbeit Teil 3b: Splines

Aufgabenstellung

Um eine möglichst glatte Kurve durch eine bestimmte Anzahl von Punkten zu zeichnen, können Splines verwendet werden. Splines sind Kurvenstücke aus kubischen Kurven, die so "zusammengeklebt" werden, dass an den Klebestellen die ersten beiden Ableitungen übereinstimmen.

Lesen Sie zuerst <http://docs.scipy.org/doc/scipy/reference/tutorial/interpolate.html> (nur bis zum Abschnitt über eindimensionale Splines).

Approximieren Sie eine Kurve mit Splines. Zeichnen Sie die Stützpunkte, die Splines und die exakte Kurve in einer Graphik mit Matplotlib. Splines können mit <https://docs.scipy.org/doc/scipy/reference/tutorial/interpolate.html> erstellt werden.

Implementation

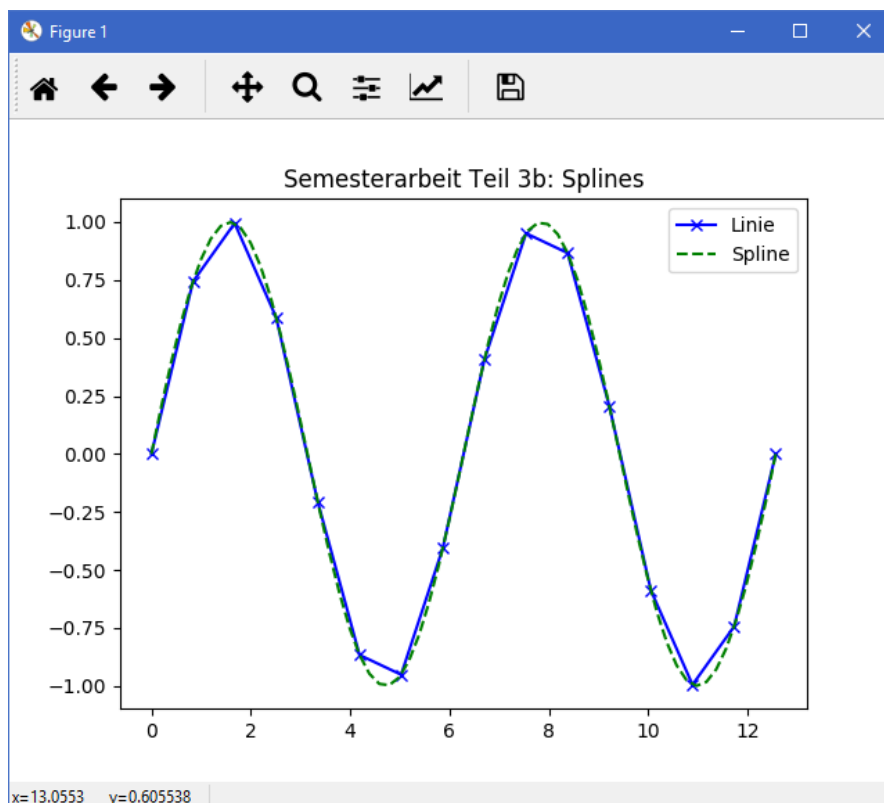
Struktur

Für diese Arbeit verwende ich folgende Scripts:

- «test.py»: Dieses Script enthält meine Implementationen.

Test-Script

Das Resultat sieht wie folgt aus:



Die Splines sind grün gestrichelt. Die bisherigen Linien sind blau. Wie man auf dem Bild erkennen kann, sind die Splines schön abgerundet, während die bisherigen Linien zackig sind.

Als erstes importiere ich die numpy-, pyplot- und interpolate-Library für die mathematischen Funktionen:

```
import numpy as np
import matplotlib.pyplot as plt

from scipy.interpolate import interp1d
```

Dann kommt die Konfiguration:

```
waveCount = 2

xMin = 0
xMax = (waveCount * 2) * np.pi

xStepCount = 4

splineAccuracy = 4
```

- Die Variable `waveCount` definiert, wieviel Sinus-Kurven gezeichnet werden.
- Die Variable `xMin` definiert den linken, minimalen X-Wert.
- Die Variable `xMax` definiert den rechten, maximalen X-Wert.
- Der Y-Wert wird durch die Sinus-Funktion bestimmt. Hier könnte eine beliebig andere mathematische Funktion eingesetzt werden.
- Die Variable `xStepCount` definiert die Schrittzahl für die X-Werte. Diese Schritte werden mit PI angepasst, damit sie mit der Sinus-Kurve harmonisieren.
- Die Variable `splineAccuracy` definiert die Genauigkeit für die Splines. Falls die Splines zu ungenau sind, kann dieser Wert erhöht werden.

Nun zum Code:

```
x = np.linspace(xMin, xMax, num=(xStepCount * (waveCount * 2)))
y = np.sin(x)
```

Die `np.linspace`-Funktion gibt `xStepCount * (waveCount * 2)` Werte von `xMin` bis `xMax` zurück.

Die `np.sin`-Funktion gibt die Sinus-Y-Werte für `x` zurück.

```
f = interp1d(x, y, kind='cubic')
```

Die `interp1d`-Funktion interpoliert eine eindimensionale Funktion.

```
xnew = np.linspace(xMin, xMax, num=(xStepCount * (waveCount * 2) * splineAccuracy))
```

Wie oben erklärt, zusätzlich wird noch mit `splineAccuracy` multipliziert.

Der letzte Teil ist für die Darstellung zuständig:

```
plt.plot(x, y, 'bx-', xnew, f(xnew), 'g--')

plt.legend(['Linie', 'Spline'], loc='best')

plt.title('Semesterarbeit Teil 3b: Splines')

plt.show()
```

Auf diesen Teil gehe ich nicht weiter ein, da ich diesen bereits in den vorherigen Arbeiten verwendet und vorgestellt habe.