Übungsserie 3

Erstellen Sie für ihre manuelle Lösung für die Aufgabe 1 eine PDF-Datei *Name_S3_Aufg1.pdf* und fassen Sie diese mit Ihren Python-Skripts zu den Aufgaben 2 und 3 in einer ZIP-Datei *Name_S3.zip* zusammen. Laden Sie dieses File vor der Übungsstunde nächste Woche auf Moodle hoch.

Aufgabe 1 (ca. 40 Minuten):

Berechnen Sie für das nichtlineare Gleichungssystem $f(x_1, x_2) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = 0$ mit

$$\begin{aligned} f_1(x_1, x_2) &= 20 - 18x_1 - 2x_2^2 \\ f_2(x_1, x_2) &= -4x_2 \cdot (x_1 - x_2^2) \end{aligned}$$

die ersten zwei Schritte mit dem Newton-Verfahren für den Startvektor $x^{(0)} = (1.1, 0.9)^T$. Geben Sie jeweils für jeden Iterationsschritt k zusätzlich noch $\| f(x^{(k)}) \|_2$ und $\| x^{(k)} - x^{(k-1)} \|_2$ an. Schreiben Sie den kompletten Lösungsweg mit Zwischenschritten und den Resultaten pro Iteration auf und scannen Sie Ihre Lösung in die Datei *Name_S3_Aufg1.pdf*. Die Berechnungen führen Sie aber natürlich mit Python durch.

Aufgabe 2 (ca. 40 Minuten):

LORAN steht für LOng RAnge Navigation und ist ein erdgebundenes Funknavigationssystem, welches in der Luft- und Seefahrt zur Positionsbestimmung zum Einsatz kommt (siehe z.B. http://de.wikipedia.org/wiki/LORAN), als unterdessen etwas überholte Alternative zum satellitengestützten GPS. Die Sender emittieren nach einem festen Schema Impulsgruppen. Aus den Laufzeitdifferenzen der bei einem Empfänger eintreffenden Signale ist es möglich, hyperbelförmige Ortskurven und aus deren Schnittpunkten die Position des Empfängers zu berechnen. Die folgenden zwei impliziten Funktionsgleichungen sind ein solches Beispiel zur Bestimmung der $(x, y)-$Koordinaten eines Empfängers:

$$\begin{aligned} f_1(x, y) = \frac{x^2}{186^2} - \frac{y^2}{300^2 - 186^2} - 1 &= 0 \\ f_2(x, y) = \frac{(y - 500)^2}{279^2} - \frac{(x - 300)^2}{500^2 - 279^2} - 1 &= 0 \end{aligned}$$

Schreiben Sie ein Skript *Name_S3_Aufg2.py* welches Ihnen die beiden folgenden Aufgaben löst:

a) Lösen Sie das Gleichungssystem zuerst grafisch, indem Sie mit Python und dem Befehl `sympy.plot_implicit()` (siehe Online-Dokumentation https://docs.sympy.org/latest/modules/plotting.html) die beiden durch f_1 und f_2 definierten Hyperbeln zusammen in ein Grafikfenster plotten und anschliessend von Auge Näherungswerte für die Koordinaten der vier Schnittpunkte zwischen den beiden Hyperbeln bestimmen. Das erreichen Sie durch

```
import sympy as sp
x, y = sp.symbols('x y')
f1 = x**2/186**2 - y**2/(300**2-186**2) - 1
f2 = (y-500)**2/279**2 - (x-300)**2/(500**2 - 279**2) - 1
p1=sp.plot_implicit(sp.Eq(f1,0),(x,-2000,2000),(y,-2000,2000))
p2=sp.plot_implicit(sp.Eq(f2,0),(x,-2000,2000),(y,-2000,2000))
p1.append(p2[0])
p1.show()
```

b) Benutzen Sie Ihre unter a) bestimmten Näherungsvektoren und bestimmen Sie mit dem Newton-Verfahren die vier Lösungen mit einer Genauigkeit $\| f(x^{(k)} \|_2 < 10^{-5}$.

Aufgabe 3 (ca. 40 Minuten):

Berechnen Sie für das nitchlineare Gleichungssystem

$$f(x_1,x_2,x_3) = \begin{pmatrix} f_1(x_1,x_2,x_3) \\ f_2(x_1,x_2,x_3) \\ f_3(x_1,x_2,x_3) \end{pmatrix} = \begin{pmatrix} x_1 + x_2^2 - x_3^2 - 13 \\ \ln \frac{x_2}{4} + \mathrm{e}^{0.5x_3-1} - 1 \\ (x_2-3)^2 - x_3^3 + 7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

ausgehend vom Startvektor $x^{(0)} = (1.5, 3, 2.5)^T$ mit dem gedämpften Newton-Verfahren die Lösung mit einer Genauigkeit $\| f(x^{(k)} \|_2 < 10^{-5}$ in einem Skript *Name_S3_Aufg3.py*.