Numerik II Hausaufgaben 1

Robert Müller

Aufgabe 1

Gegeben sei das nichtlineare Gleichungssystem

$$0 = F((x,y)) = \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \end{pmatrix}$$

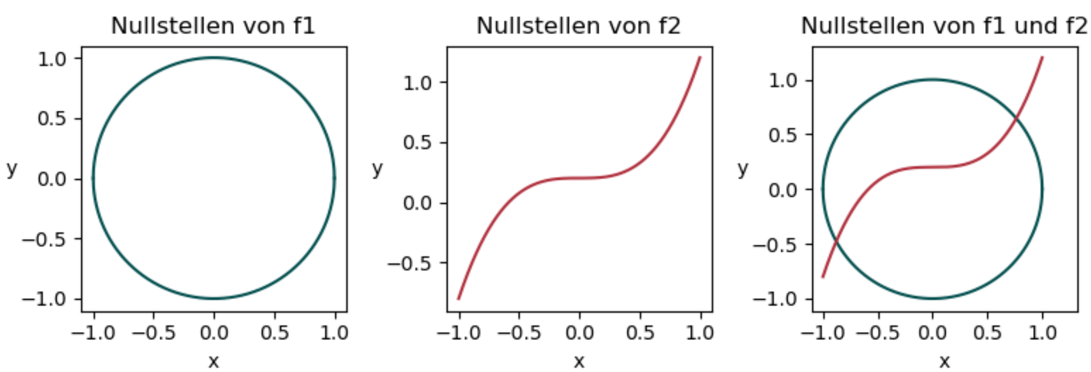
mit $f_1(x,y) = x^2 + y^2 - 1$ und $f_2(x,y) = x^3 - y + 0,2$.

a) Stellen Sie die Kurven Nullstellen der Kurven $f_1 = 0$ und $f_2 = 0$ in einem kartesischen x-y-Koordinatensystem dar! Bestimmen Sie Anzahl der Lösungen von $F(x,y) = 0$. Lesen Sie aus der grafischen Darstellung einen Näherungswert x_0 für eine der Lösungen ab (auf eine Dezimalstelle genau)!

b) Führen Sie mit x_0 drei Schritte des zweidimensionalen Newtonverfahren aus! Notieren Sie alle relevanten Zwischenwerte!

Lösung

a) Offensichtlich schneiden sich die Nullmengen von f_1 und f_2 in zwei Punkten. Daher hat $F((x,y)) = 0$ zwei Lösungen. Den Quellcode für den Plot findet man nach der letzten Aufgabe.



Wähle als Startpunkt der Iteration: $x_0 = \begin{pmatrix} 0.6 \\ 0.6 \end{pmatrix}$

b) Die Lösungen wurden automatisch durch ein Python Programm generiert. Dieses findet man ebenfalls nach der letzten Aufgabe.

In der 0. Iteration ergibt sich:

$$x_0 = \begin{pmatrix} 0.6 \\ 0.6 \end{pmatrix}, \quad JF(x_0) = \begin{pmatrix} 1.20000 & 1.20000 \\ 1.08000 & -1.00000 \end{pmatrix}, \quad F(x_0) = \begin{pmatrix} -0.28000 \\ -0.18400 \end{pmatrix}$$

Dann ergibt sich für x_1:

$$x_1 = x_0 - JF(x_0)^{-1}F(x_0) = \begin{pmatrix} 0.800641025641026 \\ 0.632692307692308 \end{pmatrix}$$

In der 1. Iteration ergibt sich:

$$x_1 = \begin{pmatrix} 0.800641025641026 \\ 0.632692307692308 \end{pmatrix}, \quad JF(x_1) = \begin{pmatrix} 1.60128 & 1.26538 \\ 1.92308 & -1.00000 \end{pmatrix}, \quad F(x_1) = \begin{pmatrix} 0.04133 \\ 0.08054 \end{pmatrix}$$

Dann ergibt sich für x_1:

$$x_2 = x_1 - JF(x_1)^{-1}F(x_1) = \begin{pmatrix} 0.765139393788455 \\ 0.644959342975880 \end{pmatrix}$$

In der 2. Iteration ergibt sich:

$$x_2 = \begin{pmatrix} 0.765139393788455 \\ 0.644959342975880 \end{pmatrix}, \quad JF(x_2) = \begin{pmatrix} 1.53028 & 1.28992 \\ 1.75631 & -1.00000 \end{pmatrix}, \quad F(x_2) = \begin{pmatrix} 0.00141 \\ 0.00298 \end{pmatrix}$$

Dann ergibt sich für x_2:

$$x_3 = x_2 - JF(x_2)^{-1}F(x_2) = \begin{pmatrix} 0.763754145227816 \\ 0.645508967131841 \end{pmatrix}$$

Aufgabe 2

Gegeben sie die Funktion $P(x) = 3x^3 - 5x^2 - x + 1$. Führen Sie sechs Schritte des Newtonverfahrens mit

a) $x_0 = 2$

b) $x_0 = 0$

aus. Erklären Sie das Resultat!

Lösung

Auch hier wurde ein Python-Programm zur Hilfe genommen. Der Quellcode ist dem Anhang nach der letzten Aufgabe zu entnehmen.

a) In der 0. Iteration ergibt sich:

$$x_0 = 2.000000000000000, \quad f^{'}(x_0) = 15.000000000000000, \quad f(x_0) = 3.000000000000000$$

Dann ergibt sich für x_1:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = \underline{1.800000000000000}$$

In der 1. Iteration ergibt sich:

$$x_1 = 1.800000000000000, \quad f^{'}(x_1) = 10.160000000000004, \quad f(x_1) = 0.495999999999999$$

Dann ergibt sich für x_2:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = \underline{1.751181102362205}$$

In der 2. Iteration ergibt sich:

$$x_2 = 1.751181102362205, \quad f^{'}(x_2) = 9.087906255812516, \quad f(x_2) = 0.026343741380396$$

Dann ergibt sich für x_3:

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = \underline{1.748282333312563}$$

In der 3. Iteration ergibt sich:

$$x_3 = 1.748282333312563, \quad f^{'}(x_3) = 9.025596719629746, \quad f(x_3) = 0.000090347014428$$

Dann ergibt sich für x_4:

$$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)} = \underline{1.748272323224894}$$

In der 4. Iteration ergibt sich:

$$x_4 = 1.748272323224894, \quad f^{'}(x_4) = 9.025381813138576, \quad f(x_4) = 0.000000001075618$$

Dann ergibt sich für x_5:

$$x_5 = x_4 - \frac{f(x_4)}{f'(x_4)} = \underline{1.748272323105717}$$

In der 5. Iteration ergibt sich:

$$x_5 = 1.748272323105717, \quad f^{'}(x_5) = 9.025381810579976, \quad f(x_5) = 0.000000000000001$$

Dann ergibt sich für x_6:

$$x_6 = x_5 - \frac{f(x_5)}{f'(x_5)} = \underline{1.748272323105717}$$

In der 6. Iteration ergibt sich:

$$x_6 = 1.748272323105717, \quad f^{'}(x_6) = 9.025381810579976, \quad f(x_6) = -0.000000000000004$$

Dann ergibt sich für x_7:

$$x_7 = x_6 - \frac{f(x_6)}{f'(x_6)} = \underline{1.748272323105717}$$

b) In der 0. Iteration ergibt sich:

$$x_0 = 0.00000, \quad f^{'}(x_0) = -1.00000, \quad f(x_0) = 1.00000$$

Dann ergibt sich für x_1:

$$x_1 = x_0 - \frac{f(x_0)}{f^{'}(x_0)} = \underline{1.00000}$$

In der 1. Iteration ergibt sich:

$$x_1 = 1.00000, \quad f^{'}(x_1) = -2.00000, \quad f(x_1) = -2.00000$$

Dann ergibt sich für x_2:

$$x_2 = x_1 - \frac{f(x_1)}{f^{'}(x_1)} = \underline{0.00000}$$

In der 2. Iteration ergibt sich:

$$x_2 = 0.00000, \quad f^{'}(x_2) = -1.00000, \quad f(x_2) = 1.00000$$

Dann ergibt sich für x_3:

$$x_3 = x_2 - \frac{f(x_2)}{f^{'}(x_2)} = \underline{1.00000}$$

In der 3. Iteration ergibt sich:

$$x_3 = 1.00000, \quad f^{'}(x_3) = -2.00000, \quad f(x_3) = -2.00000$$

Dann ergibt sich für x_4:

$$x_4 = x_3 - \frac{f(x_3)}{f^{'}(x_3)} = \underline{0.00000}$$

In der 4. Iteration ergibt sich:

$$x_4 = 0.00000, \quad f^{'}(x_4) = -1.00000, \quad f(x_4) = 1.00000$$

Dann ergibt sich für x_5:

$$x_5 = x_4 - \frac{f(x_4)}{f^{'}(x_4)} = \underline{1.00000}$$

In der 5. Iteration ergibt sich:

$$x_5 = 1.00000, \quad f^{'}(x_5) = -2.00000, \quad f(x_5) = -2.00000$$

Dann ergibt sich für x_6:

$$x_6 = x_5 - \frac{f(x_5)}{f^{'}(x_5)} = \underline{0.00000}$$

In der 6. Iteration ergibt sich:

$$x_6 = 0.00000, \quad f^{'}(x_6) = -1.00000, \quad f(x_6) = 1.00000$$

Dann ergibt sich für x_7:

$$x_7 = x_6 - \frac{f(x_6)}{f^{'}(x_6)} = \underline{1.00000}$$

Die Werte 1.0 und 0.0 wechseln sich immer ab. Das Newton-Verfahren konvergiert mit dem Startwert 0.0 nicht.

Aufgabe 3

Gegeben sei das Polynom $P(x) = 254520x^5 - 580044x^4 + 444746x^3 - 77171x^2 - 77910x + 38955$

a) Berechnen Sie ausgehend von $x_0 = 0$ vier Glieder der Newton-Iterationsfolge! Formulieren Sie eine Vermutung, wie sich die Iterationsfolge fortsetzt!

b) Berechnen Sie das nächste Folgenglied! Korrigieren Sie gegebenenfalls Ihre Vermutung aus a)!

Lösung

Die Schritte wurden mithilfe eines Taschenrechners berechnet.

a) In der 0. Iteration ergibt sich:

$$x_0 = 0.00000, \quad P(x_0) = 38955, \quad P^{'}(x_0) = 77910$$

Dann ergibt sich für x_1:

$$x_1 = x_0 - \frac{P(x_0)}{P'(x_0)} = 0 - \frac{38955}{77910} = \frac{1}{2}$$

In der 1. Iteration ergibt sich:

$$x_1 = \frac{1}{2}, \quad P(x_1) = 8001.5, \quad P^{'}(x_0) = -32006$$

Dann ergibt sich für x_2:

$$x_2 = x_1 - \frac{P(x_1)}{P'(x_1)} = \frac{1}{2} - \frac{8001.5}{-32006} = \frac{3}{4}$$

In der 2. Iteration ergibt sich:

$$x_2 = \frac{3}{4}, \quad P(x_2) = 206115 \cdot 32, \quad P^{'}(x_2) = -618345 \cdot 128$$

Dann ergibt sich für x_3:

$$x_3 = x_2 - \frac{P(x_2)}{P'(x_2)} = \frac{3}{4} - \frac{206115 \cdot 32}{-618345 \cdot 128} = \frac{5}{6}$$

In der 3. Iteration ergibt sich:

$$x_3 = \frac{5}{6}, \quad P(x_3) = 6715 \cdot 3, \quad P^{'}(x_3) = -26860 \cdot 18$$

Dann ergibt sich für x_4:

$$x_4 = x_3 - \frac{P(x_3)}{P'(x_3)} = \frac{5}{6} - \frac{6715 \cdot 3}{-26860 \cdot 18} = \frac{7}{8}$$

Vermutung:
Das Newton-Verfahren konvergiert vermutlich gegen einen Wert in der Nähr von 1.

b) In der 3. Iteration ergibt sich:

$$x_4 = \frac{7}{8}, \quad P(x_4) = 734069 \cdot 512, \quad P^{'}(x_4) = 104867 \cdot 4069$$

Dann ergibt sich für x_5:

$$x_5 = x_4 - \frac{P(x_4)}{P'(x_4)} = \frac{7}{8} - \frac{734069 \cdot 512}{104867 \cdot 4069} = \frac{7}{8} - \frac{7}{8} = 0$$

Folglich konvergiert das Newton-Verfahren für $x_0 = 0$ <u>nicht</u> und wechselt periodisch zwischen den Werten:

$$0, \quad \frac{1}{2}, \quad \frac{3}{4}, \quad \frac{5}{6}, \quad \frac{7}{8}$$

Aufgabe 4

Klassifizieren Sie die folgenden Differentialgleichungen unter Verwendung der Begriffe, linear/nichtlinear, homogen/inhomogen, konstante/nichtkonstante Koeffizienten!

a) $x^2y' = y^2$

b) $y^{'''} - 4y' = 2\sin(\omega x + \varphi)$

c) $y^{(4)} = y + x^2 \exp(-x)$

d) $x^2y'' + xy' + (x^2 - 1)y = 0$

Lösung

Differentialgleichung	linear	homogen	konstante Koeffizienten
a) $x^2y' = y^2$	nein	ja	nein
b) $y''' - 4y' = 2\sin(\omega x + \varphi)$	ja	nein	ja
c) $y^{(4)} = y + x^2 \exp(-x)$	ja	nein	ja
d) $x^2y'' + xy' + (x^2 - 1)y = 0$	ja	ja	nein

Aufgabe 5

Lösen Sie folgende Differntialgleichungen erster Ordnung

a) $y' = (1 - y)^2$

b) $y'(x^2 + 1) = xy$

c) $y' + \cos(x)y = 0$ mit $y(\frac{\pi}{2}) = 2\pi$

Lösung

a) Setze $u = 1 - y$. Dann gilt:

$$\frac{du}{dx} = \frac{du}{dy}\frac{dy}{dx} = -\frac{dy}{dx} = -y' \implies u' = -u^2$$

Dann löst $u(x) = \frac{1}{x}$ die u-abhängige Differentialgleichung, also ist

$$y = \frac{x - 1}{x}$$

b) $y\prime(x^2+1) = xy \implies y\prime = \frac{x}{1+x^2}y$,da $1+x^2 \neq 0$

Also gilt:

$$\begin{aligned} y(x) = \exp\left(\int_0^x \frac{t}{1+t^2}dt\right) &\overset{u=1+t^2}{=} \exp\left(\frac{1}{2}\int_1^{1+x^2}\frac{1}{u}du\right) \\ &= \exp(\ln\sqrt{1+x^2}) = \sqrt{1+x^2} \end{aligned}$$

c) Wir betrachten das Anfangswertproblem: $y\prime + cos(x)y = 0$ mit $y(\frac{\pi}{2}) = 2\pi$.

$$y(x) = 2\pi\exp\left(\int_{\pi/2}^x -\cos\tau d\tau\right) = 2\pi\exp(1-\sin(x))$$

Aufgabe 6

Wir betrachten die folgenden Anfangswertprobleme:

i) $e^y\frac{dy}{dx} = e^{-x} - e^y, \quad y(0) = 0$

ii) $y\prime = 2\sqrt{|y|}, \quad y(0) = 0$

a) Besitzen die AWP Lösungen und sind diese eindeutig?

b) Finden Sie die Lösung des AWP's i) mithilfe der Substitution $u = e^y$.

Lösung

i) a) und b):
Setze $u = e^y$, dann folgt:

$$\begin{aligned} u\frac{dy}{dx} &= e^{-x} - u \\ u\prime = \frac{du}{dx} &= \frac{du}{dy}\frac{dy}{dx} = u\frac{dy}{dx} = e^{-x} - u, \quad u(0) = 1 \end{aligned}$$

Löse nun das AWP für u:

$$\begin{aligned} u_0(x) &= \exp(-x) \\ u(x) &= u_0(x)\left(1 + \int_0^x \frac{\exp(-t)}{\exp(-t)}dt\right) = \exp(-x)(1+x) \end{aligned}$$

Folglich ist

$$y(x) = \ln(u(x)) = ln(1+x) - x$$

eine Lösung der Differentialgleichung. Diese Lösung ist sogar eindeutig,da die Funktion $f(x,u) = e^{-x} - u(x) \overset{!}{=} u\prime$ ein lokale Lipschitzbedingung erfüllt. Da der *logarithmus naturalis* auch stetig ist erfüllt auch i) eine lokale Lipschitzbedingung.

ii) a) und b):
Da sowohl $y \equiv 0$ als auch $x|x|$ Lösungen des AWP's sind, ist dies offenbar nicht eindeutig lösbar. Das folgt daraus, dass die Wurzelfunktion bei 0 nicht lokal Lipschitzstetig ist. Weiterhin sind für $a \leq 0 \leq b \in [-\infty, \infty]$:

$$y(x) = -\mathbb{1}_{(-\infty,a)}(x)(x-a)^2 + \mathbb{1}_{(b,\infty)}(x)(x-b)^2$$

weitere Lösungen.
(Mit der arithmetischen Definition: $0 \cdot \infty := 0$)

Quellcode von Python-Programmen

Hier werden nun die Quellcodes der jeweiligen Aufgaben angegeben. Da sowohl Python als andere Rechenmittel (z.B. Taschenrechner) nur begrenzt genau arbeiten sind die so entstandenen Lösung mit Vorsicht zu genießen (wie Scans von Xerox-Workcenter). ;)

Quellcode Aufgabe 1a):

```
import matplotlib.pyplot as plt
import numpy as np
from sympy.solvers import solve
from sympy import Symbol


f1 = lambda x,y: x**2+y**2-1.0
f2 = lambda x,y: x**3-y+0.2

x=Symbol('x')
y=Symbol('y')
z=Symbol('z')
data1=solve(f1(x,y),x,y)
data2=solve(f2(x,y),x,y)


g1 = lambda x: -np.sqrt(1.0 - x**2)
g2 = lambda x: np.sqrt(1.0 - x**2)
g3 = lambda x: x**3+0.2

dt=0.0025
u = np.arange(-1.0,1.0,dt)
t = np.append(u,1)

color1=np.random.rand(3)
color2=np.random.rand(3)

fig, axs = plt.subplots(1,3,num='Yeet',figsize=(8,3))

axs[0].plot(t,g1(t),color=color1)
axs[0].plot(t,g2(t),color=color1)
axs[0].set_title('Nullstellen von f1')

axs[1].plot(t,g3(t),color=color2)
axs[1].set_title('Nullstellen von f2')

axs[2].plot(t,g1(t),t,g2(t),color=color1)
axs[2].plot(t,g3(t),color=color2)
axs[2].set_title('Nullstellen von f1 und f2')

for i in range(0,3):
    axs[i].set_xticks([-1,-0.5,0,0.5,1])
    axs[i].set_yticks([-1,-0.5,0,0.5,1])
    axs[i].set_xlabel('x')
    axs[i].set_ylabel('y',rotation=0)
```

```
    axs[i].axis('square')

fig.tight_layout()
plt.show()
```

Quellcode Aufgabe 1b)

Hierfür wurde der Quellcode aus Aufgabe 2 etwas modifiziert.

```python
import numpy as np
f1 = lambda x,y: x**2+y**2-1.0
f2 = lambda x,y: x**3-y+0.2

F = lambda x,y: np.array([[x**2+y**2-1],[x**3-y+0.2]])
JF = lambda x,y: np.array([[2*x,2*y],[3*x**2,-1]])

def newton(F,JF,x0,epsilon,max_iter):
    global xn
    global n
    xn=x0
    for n in range(0,max_iter):
        print("-----------Iteration",n,"-----------")
        print()
        print( "x=%1.64f" % (xn[0][0]))
        print( "y=%1.64f" % (xn[1][0]))
        fxn = F(xn[0], xn[1])
        if abs(np.linalg.norm(fxn)) < epsilon:
            print("-----------Loesung gefunden. Nach",n,"Iterationen-----------")
            return [xn[0][0],xn[1][0]]
        print()
        print("JF(x,y)=")
        print(JF(xn[0],xn[1]))
        print()
        print( "F(x,y)=")
        print(F(xn[0],xn[1]))
        print()
        xn = xn - np.matmul(np.linalg.inv(JF(xn[0][0],xn[1][0])),F(xn[0][0],xn[1][0]))
        xn = [xn[0],xn[1]]
        print("xn+1=(%1.15f,%1.15f)" % (xn[0][0],xn[1][0]))
        print()
    print( "-----------Iterationsgrenze erreicht. Keine Loesung gefunden.-----------")
    return None



x0 = np.array([[0.6],[0.6]])
res = newton(F,JF,x0,1e-15,3)
if res==None:
    print()
    print("Nach",n,"Schritten war die kleinste Loesung:")
    print( "(x,y)=(%1.15f,%1.15f)." % (xn[0][0],xn[1][0]))
    print()
    print("Mit einer 2-Norm(F(x,y))=%1.30f" % np.linalg.norm(F(xn[0][0],xn[1][0])))
else:
    [x,y] = res
    print( "x=%1.15f" % x)
    print( "y=%1.15f" % y)
    print( "mit 2-Norm(F(x,y))= %1.30f" % np.linalg.norm(F(x,y)))
```

Quellcode Aufgabe 2

Dieser Quellcode wurde auf https://www.math.ubc.ca/ pwalls/math-python/roots-optimization/newton/ gefunden und noch etwas angepasst.

```python
def newton(f,Df,x0,eps,max_iter):
    global xn
    xn = x0
    print("x%1d = %1.15f" % (0,xn))
    for n in range(0,max_iter):
        fxn = float(f(xn))
        #if abs(fxn) < eps:
            #print("Passende Loesung nach %1d Iterationsschritten gefunden." % n)
            #return xn
        Dfxn = float(Df(xn))
        print("f'(x%1d) = %1.15f" % (n,Dfxn))
        print("f(x%1d) = %1.15f" % (n,fxn))
        if Dfxn == 0:
            print("Die Ableitung verschwindet. Keine Loesung gefunden")
            return None
        xn = xn - fxn/Dfxn
        print("x%1d=%1.15f" % (n+1,xn))
        print()
        if xn in l:
            print(n)
            print("Das Newton-Verfahren konvergiert nicht!")
            break
        l.append(xn)
    print("Maximale Anzahl an Iterationen erreicht. Keine Loesung gefunden!")
    print("Nach %1d Schritten war die kleinste Loesung: %1.15f ." % (n,xn))
    return None

x0 = 2.
l =[]
f = lambda x: 3*x**3 - 5*x**2 - x + 1
Df = lambda x: 9*x**2 - 10*x - 1
newton(f,Df,x0,1e-10,5)
```