Blatt 5: Newton-Verfahren **MNEU**

Aufgabe 1 :

Berechnen Sie mit `NLGS/NewtonIteration` die Nullstelle von

$$f(x) = x^3 - 2\,x + 2\,.$$

Verwenden Sie `TOL=1.0e-08` und $x_0 \in \{-1, 0, 1, 2\}$. Was stellen Sie fest? Erstellen Sie eine graphische Darstellung der Funktion f.

Aufgabe 2 :

Rechnen Sie folgende Beispiele einmal durch, um sich die Situationen jeweils klar zu machen:

$$(a) \qquad f(x) = x^2 + 2 \qquad x^0 = 1$$

und

$$(b) \qquad f(x) = \cos x \qquad x^0 = \frac{\pi}{4}$$
$$x^0 = \operatorname{acot}(2\,\pi)$$

Aufgabe 3 :

Zur Zeit t sei eine bestimmte Konzentration einer chemischen Substanz durch

$$g(t) = 10 \cdot \mathrm{e}^{-3t} + 2 \cdot \mathrm{e}^{-5t}$$

gegeben. Es soll der Zeitpunkt bestimmt werden, in dem die Konzentration nur noch halb so groß wie zur Zeit $t = 0$ ist.

(a) Wie lautet die Funktion, deren Nullstelle gesucht ist?

(b) Wieviele Nullstellen hat diese Funktion und warum?

(c) Berechnen Sie mit `MyNewton.m` eine Näherung der gesuchten Nullstelle(n).

Aufgabe 4 :

Leiten Sie eine Formel zur Berechnung von $\sqrt{2}$, indem Sie das Newton-Verfahren auf eine geeignete Funktion anwenden.

(a) Wie lautet die Funktion, deren Nullstelle Sie suchen?

(b) Verwenden Sie den Startwert $x^0 = 2$. Lassen Sie sich zu Iterationen $n \in \{0, 1, \ldots, 5\}$ jeweils den Fehler ausgeben.

(c) Berechnen Sie den EOC mit dem vorbereiteten MyNewtonEOC.m Versuchen Sie sich gerne zunächst an ein paar Werten aus der Tabelle und rechnen Sie diese per Hand nach:

$$\frac{\ln\left(\frac{E_2}{E_3}\right)}{\ln\left(\frac{E_1}{E_2}\right)} = \dots$$

Aufgabe 5 Baywatch:

Von Weitem sieht ein am Strand stehender Rettungsschwimmer einen Schwimmer in Not. Die Wasser-Strandgrenze befindet sich von beiden genau 5 Meter entfernt. Allerdings befindet sich der Ertrinkende nicht auf der Höhe des Rettungsschwimmers sondern 4 Meter versetzt. Der Rettungsschwimmer möchte nun möglichst schnell zum Ertrinkenden gelangen. Dazu würde er zunächst den Weg, der am kürzesten ist wählen. Allerdings ist er im Wasser (0,5 m/s) langsamer als auf dem Sand (1,4 m/s), so dass er vielleicht den Weg wählen soll, der möglichst viel, bzw. möglichst wenig im Sand verläuft. Wenn er noch lange nachdenken muss schafft er es nicht mehr rechtzeitig zum Ertrinkenden. Helfen Sie ihm!

(1) Skizzieren Sie die Situation. Fügen Sie die Varianten "kürzester Weg", "möglichst wenig Wasserweg" in Ihre Skizze.

(2) x sei die Strecke, die der Rettungsschwimmer parallel zur Wassergrenze läuft, bevor er in's Wasser springt. Erstellen Sie die Funktion $T(x)$, die die Zeit angibt, welche der Rettungsschwimmer benötigt, um den Ertrinkenden zu erreichen, je nachdem wo er entscheidet ins's Wasser zu springen.

(3) Wie lautet die Funktion `f(x)` im Programm `NLGS/MyNewton.m`, um das Baywatch-Problem zu lösen?

(4) Welches ist der Bereich sinnvoller Startwerte?

$$I =$$

(5) Wählen Sie die Parameter

$$\texttt{TOL} = 10^{-12}\,,\ \texttt{Itmax} = 1000 \quad \text{und} \quad \texttt{x0} = 0\,,$$

berechnen Sie die Lösung des Problems und erzeugen sie graphische Darstellungen der Funktionen `f(x)` und `df(x)`, sowie ihrer Lösung.

Beantworten Sie folgende Fragen und geben Sie alle Zahlen mit 6 Nachkommastellen an.

(a) Wo springt der Rettungsschwimmer in's Wasser?

(b) Wie lange braucht er, um den Ertrinkenden zu erreichen?

(c) Wieviele Iterationen waren nötig, um die Näherungslösung mit der gegebenen Toleranz zu erreichen?

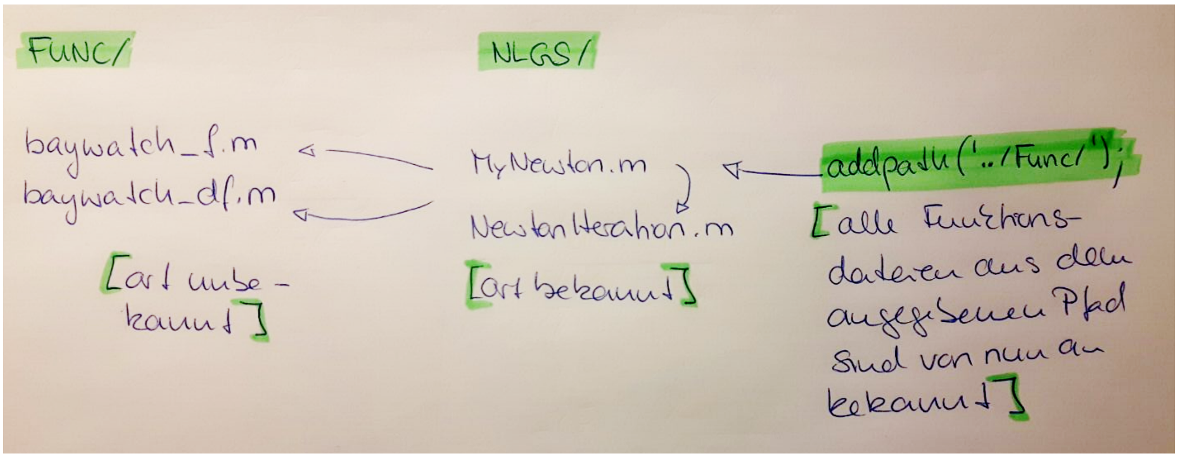
Aufgabe 6 :

Häufig kommt es vor, dass Funktionen an mehreren Stellen im Programm auftreten. Man möchte sie dann nicht jedes mal neu definieren. Es bietet sich dann an, im Hinblick auf höherer Modularität des Programms, Funktionsdateien zu erstellen, die dann von verschiedenen Funktionen ihrerseits aufgerufen werden. Wie das geht wissen wir schon. Bisher war es aber so, dass diese Dateien, dann im gleichen Verzeichnis enthalten sein müssen wie die Funktion, die sie aufruft. Das ist unproaktisch. Der Ordnung wegen sollen diese Funktionen ein eigenes Verzeichnis erhalten. Dieses Verzeichnis muss dann noch in den Pfad gesetzt werden:

(a) Erstellen Sie ein Verzeichnis `Func/` auf gleicher Höhe von `NLGS/`

(b) Erzeugen Sie Funktionsdateien `baywatch_f.m` und `baywatch_df.m`, die die Funktion T', bzw. `f(x)` und T'', bzw. `df(x)` aus Aufgabe 5 bereitstellen.

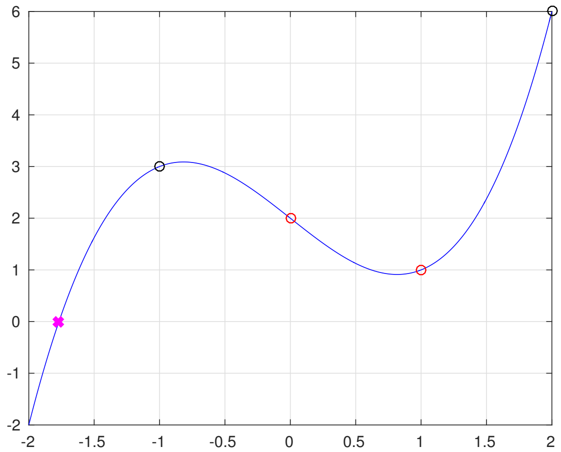
(c) Mit dem Befehl `addpath('../Func/')` in `MyNewton.m` machen Sie Ihrer Routine alle Funktionen im Verzeichnis `../Func/` bekannt. Die folgende kleine Übersicht sollte beim Verständnis dienlich sein:



Schreiben Sie `MyNewton.m` so um, dass die Funktionen `baywatch_f.m` und `baywatch_df.m` an die Funktion `MyNewton.m` übergeben werden können.

Lösung 1:

Bei der der Wahl der Startwerte $x_0 \in \{0, 1\}$ konvergiert der Algorithmus nicht (○), hingegen bei $x_0 \in \{-1, 2\}$ (○) schon. Die gesuchte Nullstelle befindet sich bei x.



Lösung 2:

$f(x) = x^2 + 2$	$x^0 = 1$	divergiert
$f(x) = \cos x$	$x^0 = \dfrac{\pi}{4}$	konvergiert zu $\dfrac{\pi}{2}$
	$x^0 = \operatorname{acot}(2\,\pi)$	divergiert (theoretisch)
$f(x) = x^3 - 2\,x + 2$	$x^0 = 0$	oszilliert

Lösung 3:

(a)

$$f(x) = g(x) - g(0)\frac{1}{2} = 10 \cdot \mathrm{e}^{-3t} + 2 \cdot \mathrm{e}^{-5t} - 6$$

(b) g ist monoton fallend. Mit

$$g(0) > 0 \quad \text{und} \quad \lim_{x\to\infty} g(t) < 0$$

wissen wir, dass es nur eine Nullstelle gibt.

(c) Mit TOL=1.0e-08 und x0=0 erhält man die Nähernug an die eindeutige Nullstelle:

$$t \approx 2.11 \cdot 10^{-1}$$

Lösung 4:

Sie können den Algorithmus auf Abfrage nach Toleranz oder Anzahl bestimmter Iterationen folgendermassen umschalten lassen, ohne die Übergabestruktur zu ändern: In `NLGS/NewtonIteration.m`

(a)

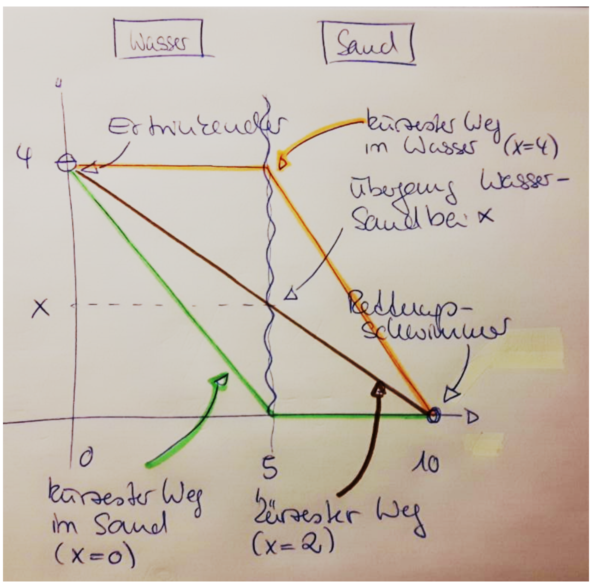
$$
\begin{aligned}
& & \sqrt{2} &= a \\
&\Leftrightarrow & 2 &= a^2 \\
&\Leftrightarrow & 0 &= a^2 - 2 \\
&\Rightarrow & f(x) &= x^2 - 2
\end{aligned}
$$

(b)
```
%[Error = 5.86e-01 nach 0 Iterationen]
%[Error = 8.58e-02 nach 1 Iterationen]
%[Error = 2.45e-03 nach 2 Iterationen]
%[Error = 2.12e-06 nach 3 Iterationen]
%[Error = 1.59e-12 nach 4 Iterationen]
%[Error = 0.00e+00 nach 5 Iterationen]
```

(c)
```
>> MyNewtonEOC
Error(1) = 8.58e-02
Error(2) = 2.45e-03
Error(3) = 2.12e-06
Error(4) = 1.59e-12

alpha(2) = 1.98392
alpha(3) = 1.99977
```

Lösung 5:



(1)

(2)

$$
\begin{aligned}
T(x) &= \frac{\text{Strecke im Wasser}}{\text{Geschwindigkeit im Wasser}} + \frac{\text{Strecke auf dem Sand}}{\text{Geschwindigkeit auf dem Sand}} \\
&= \frac{S_W}{V_W} + \frac{S_S}{V_S} = \frac{\sqrt{(4-x)^2+25}}{0,5} + \frac{\sqrt{x^2+25}}{1,4} \\
&= 2\sqrt{(4-x)^2+25} + \frac{5}{7}\sqrt{x^2+25}
\end{aligned}
$$

(3) Die Funktion $T(x)$ soll minimiert werden, dass heißt wir suchen die Nullstelle der Ableitung $T'(x) \rightsquigarrow$`f(x)`:

$$T'(x) = \mathtt{f(x)} = -2\,\frac{4-x}{\sqrt{(4-x)^2+25}} + \frac{5}{7}\,\frac{x}{\sqrt{x^2+25}}$$

Die Ableitungsfunktion im Algorithmus entspricht dann der zweiten Ableitung von T, also $T''(x) \rightsquigarrow$`df(x)`:

$$T''(x) = \mathtt{df(x)} = \frac{50}{\sqrt{(4-x)^2+25}^3} + \frac{125}{7\sqrt{x^2+25}^3}$$

(4)

$$I = [0, 4]$$

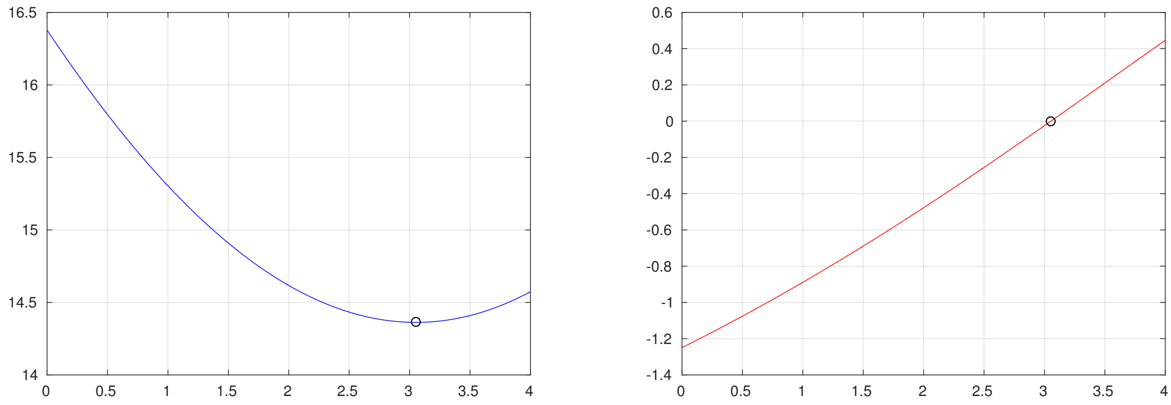
(5) Die wesentlichen Angaben in Kürze:

```
f  = @(x) -2*(4-x)./sqrt((4-x).^2+25) + 5./7.*x./sqrt(x.^2+25);
df = @(x) 50./sqrt((4-x).^2+25).^3+125./7./sqrt(x.^2+25).^3;

[x k] = NewtonIteration(0.,f,df,1.0e-12,1000,0);
```

Sie finden das vollständige Programm unter Baywatch.m.

Der graphische Output: links: Die Dauer in Sekunden, die der Rettungsschwimmer benötigt in Abhängigkeit von x. rechts: Der entsprechende Verlauf der Ableitungsfunktion $T'(x)$, dessen Nullstelle vom Newton-Verfahren ermittelt wurde. Es ist jeweils das Ergebnise des Algorithmus mit $\circ$ gekennzeichnet.



(a) Auf der Höhe von $x = 3.052884$ Metern springt der Rettungsschwimmer in's Wasser.

(b) Der Rettungsschwimmer braucht $T(x) = 14.362350$ Sekunden, um den Ertrinkenden zu erreichen.

(c) $k = 4$

Lösung 6:

(b) baywatch_f.m und baywatch_df.m

(c) In BaywatchFunc.m braucht es dann im Wesentlichen die folgenden beiden Zeilen:

```
addpath('../Func/');
[x k] = NewtonIteration(0,@baywatch_f,@baywatch_df,1.0e-12,1000,0);
```