Technische Universität Chemnitz
Fakultät für Mathematik

22. April 2009

Höhere Mathematik I.2

Aufgabenkomplex 2: Differenzialrechnung

Letzter Abgabetermin: 05. Mai 2009
(in Übung oder Briefkasten bei Zimmer Rh. Str. 41/615)
Bitte die Arbeiten deutlich mit „Höhere Mathematik I.2, Aufgabenkomplex 2“ kennzeichnen und die Übungsgruppe angeben, in der die Rückgabe erfolgen soll!

1. Die Gleichung $e^x = 3x$ soll mithilfe des Newtonverfahrens gelöst werden.
   a) Ermitteln Sie zunächst eine Lösung dieser Gleichung ausgehend vom Startwert $x_0 = 1$!
   b) Verwenden Sie nun den Startwert $x_0 = 1{,}1$! Erklären Sie den dabei zu beobachtenden Effekt!
   c) Wieviele Lösungen hat die Gleichung?
   d) Bestimmen Sie die evtl. noch fehlenden Lösungen mithilfe des Newtonverfahrens!

2. Bei einem Preis von 1,20 € pro Liter werden in Deutschland 75 Millionen Liter Benzin pro Tag abgesetzt, die Elastizität der Nachfrage bezüglich des Preises betrage $-0{,}3$. Welche relative und welche absolute Entwicklung der Nachfrage ist ungefähr zu erwarten, wenn der Preis von 1,20 € auf 1,25 € pro Liter steigt?

3. Bestimmen Sie die folgenden Grenzwerte mit Hilfe der l’Hospitalschen Regel:
   a) $\lim\limits_{x\to 0} \dfrac{1-\cos 2x}{x^2}$, b) $\lim\limits_{x\to\infty} \dfrac{\ln x}{x}$, c) $\lim\limits_{x\to 1} \dfrac{x^2+x-2}{x-1}$, d) $\lim\limits_{x\to\infty} \dfrac{e^{2x}}{x^2-x}$!

4. Der Querschnitt eines Tunnels habe die Form eines Rechtecks mit Grundseite d und Höhe h, auf das ein Halbkreis mit Durchmesser d aufgesetzt ist. Der Umfang des Querschnitts beträgt 20 m. Bestimmen Sie die Grundseitenlänge d, für die der Flächeninhalt des Querschnitts am größten wird!

5. Zum Zeitpunkt $t = 0$ werden 1000 Bakterien in eine Nährlösung gegeben. Die Zahl der Bakterien entwickelt sich nach der Formel $f(t) = \dfrac{G}{1 + Ae^{-0{,}2t}}$, wobei die Sättigungsgrenze bei 20 000 Bakterien liegt.
   a) Bestimmen Sie die Parameter G und A !
   b) Zeigen Sie, dass die Funktion $f(t)$ monoton wachsend ist!
   c) Zu welchem Zeitpunkt beträgt die Zahl der Bakterien 10 000 ? Wie groß ist zu diesem Zeitpunkt die Wachstumsgeschwindigkeit der Population?
   d) Zu welchem Zeitpunkt wächst die Population am stärksten?

6. Ein 30 Jahre alter Baum ist 11,52 m hoch und wächst in diesem Alter mit einer Geschwindigkeit von 48 cm/a (a: Jahr). Die Wachstumsgeschwindigkeit wächst ihrerseits um 1,2 cm/a^2.
   a) Bestimmen Sie mithilfe der Taylorschen Formel näherungsweise die Höhe, die der Baum in einem Alter von 35 Jahren erreicht haben wird!
   b) Auch die Änderung der Wachstumsgeschwindigkeit ist nicht konstant. Es wird aber angenommen, dass sie sich in der betrachteten Zeit um nicht mehr als 0,3 cm/a^3 ändert. Schätzen Sie unter dieser Annahme den Fehler des Ergebnisses von a) ab!

Zusatzaufgabe auf Seite 2

Zusatzaufgabe

Bei dieser Aufgabe können 10 Zusatzpunkte erworben werden, bei den Aufgaben 1 – 5 werden insgesamt 40 Punkte vergeben. Der Aufgabenkomplex ist bestanden, wenn mindestens 20 Punkte erreicht worden sind.

Lösen Sie die folgenden Aufgaben mit `MATLAB`. Protokollieren Sie Ihr Vorgehen in einer `diary`-Datei und speichern Sie erstellte Plots ab.

1. Zeichnen Sie die Funktion $f(x) = \sin(x)$ und die Taylor-Polynome ungeraden Grades bis zur 11ten Ordnung mit der Entwicklungsstelle $x_0 = 0$ im Intervall $[-4\pi, 4\pi]$ in einen gemeinsamen Plot. Beschriften Sie die Achsen und erstellen Sie eine Legende. Verwenden Sie den Befehl `axis`, um einen angemessenen Bildausschnitt auszuwählen.
2. a) Implementieren Sie das Newton-Verfahren. Erstellen Sie dazu ein extra `m-File` und arbeiten Sie mit `function-handles`. Ihrer Funktion werden die folgenden Parameter übergeben
      - die Funktion F, auf die das Verfahren angewandt werden soll (in Form eines `function-handles`),
      - die Ableitung der Funktion F (in Form eines `function-handles`),
      - der Startwert x_0.

      Zurückgegeben werden soll ein Vektor $x = (x_0, x_1, \ldots, x_n)$, welcher die gesamte Iterationsfolge enthält. Verwenden sie $\varepsilon = 10^{-8}$ als Parameter für das Abbruchkriterium (Schritt 3 im Algorithmus).

   b) Wenden Sie Ihr `m-File` zum Newton-Verfahren auf die Funktion aus Hausaufgabenkomplex I, MATLAB-Aufgabe 2a) ($f(x) = \ln(x+1)\sin(x^2)$) mit den Startwerten
      i. x_0=`1.50`
      ii. x_0=`1.43`
      iii. x_0=`1.40`

      an. Zeichnen Sie jeweils die Funktion f und den Verlauf der Iterierten.

   c) Setzen Sie nun das Newton-Verfahren ein, um x mit $f'(x) = 0$ zu finden. Verwenden Sie die Startwerte
      i. x_0=`1.70`
      ii. x_0=`3.17`
      iii. x_0=`0.70`

      und zeichnen Sie die Funktion f und f' sowie jeweils den Verlauf der Iterierten. Bestätigen Sie anhand der zweiten Ableitung, dass es sich bei der jeweils letzten Iterierten um ein lokales Maximum, ein lokales Minimum bzw. einen Sattelpunkt handelt.

Öffnen Sie die erstellte `diary`-Datei (vorher mit $\gg$ `diary off` die Protokollierung abschließen) und entfernen Sie ggf. überflüssige Zeilen (z.B. Fehleingaben). Drucken Sie anschließend die bearbeitete `diary`-Datei und eventuell angefertigte Plots und `m-Files` möglichst sparsam (d.h. nach Möglichkeit duplex, mehrere Seiten pro Blatt, kleine Schriftgröße) aus. Fügen Sie den Ausdruck Ihrer „restlichen“ Hausaufgabe an.

Hinweise zur `MATLAB`aufgabe

Schleifen

`MATLAB` unterstützt die Verwendung von `while`-Schleifen und `for`-Schleifen, siehe:

```
≫ help while
≫ help for
```

Als Beispiel dient das folgende kleine Programm. Es tabelliert die Sinus-Funktion im Intervall $[0, 2\pi]$ im Abstand von $\frac{\pi}{4}$.

```
≫ m=1;
≫ fprintf('\n\nn    n*pi/4          sin(n*pi/4)\n')
≫ while m<10,
≫      x(m)=(m-1)*pi/4;
≫      fprintf('%1.0f%16.10f%16.10f\n',m-1,x(m),sin(x(m)))
≫      m=m+1;
≫ end
```

function-handles

`Function-handles` ermöglichen es, Funktionen an Funktionen zu übergeben. Als Beispiel wollen wir hier in einer Funktion die Werte $f(0) + f(1) + \ldots f(5)$ berechnen, wobei die Funktion f durch ein `function-handle` übergeben wird. Dazu erstellen wir das `m-File` `Summe0bis5.m` mit dem Inhalt:

```
function Summe = Summe0bis5(f)
    Summe = f(0) + f(1) + f(2) + f(3) + f(4) + f(5);
```

Wir nutzen als Beispiel für die Funktion f zunächst die `sin`-Funktion,

```
≫ Summe0bis5(@sin)
```

Ein `function-handle` für die `sin`-Funktion wird also durch das voranstellen des `@` erreicht. Anstatt für jede erdenkliche Funktion f eine eigene Funktion zur Summation zu verwenden, genügt die Funktion `Summe0bis5`. Sie können auch Ihre eigenen Funktionen übergeben, sofern diese in `m-Files` ausgelagert sind:

```
≫ Summe0bis5(@Name_des_m-Files)
```

Weitere Hinweise zu `function-handles` finden Sie in der `MATLAB`-Hilfe,

```
≫ help function_handle
```

Aufgabenkomplex 2: Differenzialrechnung

Letzter Abgabetermin: 05. Mai 2009

---

1. Die Gleichung $e^x = 3x$ soll mithilfe des Newtonverfahrens gelöst werden.

a) Ermitteln Sie zunächst eine Lösung dieser Gleichung ausgehend vom Startwert $x_0 = 1$!

b) Verwenden Sie nun den Startwert $x_0 = 1{,}1$! Erklären Sie den dabei zu beobachtenden Effekt!

c) Wieviele Lösungen hat die Gleichung?

d) Bestimmen Sie die evtl. noch fehlenden Lösungen mithilfe des Newtonverfahrens!

Lösung:

a) Zu bestimmen sind die Nullstellen der Funktion $f(x) = e^x - 3x$. Die Iterationsvorschrift des Newtonverfahrens lautet $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{e^{x_n} - 3x_n}{e^{x_n} - 3}$.

n	x_n	$f(x_n)$	$f'(x_n)$
0	1,0000000000	−0,2817181715	−0,2817181715
1	0,0000000000	1,0000000000	−2,0000000000
2	0,5000000000	0,1487212707	−1,3512787293
3	0,6100596550	0,0103622280	−1,1594588071
4	0,6189967797	0,0000737235	−1,1429359373
5	0,6190612834	0,0000000039	−1,1428161461
6	0,6190612867	0,0000000000	−1,1428161398
7	0,6190612867	0,0000000000	−1,1428161398
8	0,6190612867	0,0000000000	−1,1428161398

$$\underline{\underline{x^* = 0{,}6190612867}}$$

b)

n	x_n	$f(x_n)$	$f'(x_n)$
0	1,1000000000	−2,958E−01	4,166E−03
1	72,1111079191	2,077E+31	2,077E+31
2	71,1111079191	7,641E+30	7,641E+30
3	70,1111079191	2,811E+30	2,811E+30
…	…	…	…
64	9,1129760751	9,045E+03	9,069E+03
65	8,1156597498	3,322E+03	3,343E+03
66	7,1220444379	1,218E+03	1,236E+03
67	6,1369039991	4,442E+02	4,596E+02
68	5,1704333119	1,605E+02	1,730E+02
69	4,2427566938	5,687E+01	6,660E+01
70	3,3888280170	1,946E+01	2,663E+01
71	2,6579290264	6,293E+00	1,127E+01
72	2,0993876223	1,863E+00	5,161E+00
73	1,7384214931	4,731E−01	2,688E+00
74	1,5624431157	8,313E−02	1,770E+00
75	1,5154878737	5,178E−03	1,552E+00
76	1,5121510261	2,531E−05	1,536E+00
77	1,5121345521	6,156E−10	1,536E+00
78	1,5121345517	0,000E+00	1,536E+00
79	1,5121345517	0,000E+00	1,536E+00
80	1,5121345517	0,000E+00	1,536E+00

Da die Tangente für $x_0 = 1{,}1$ sehr flach ist (Anstieg $0{,}4 \cdot 10^{-3}$) führt die Iteration zunächst sehr weit weg. Die folgenden Iterationspunkte sind wegen $f(x_n) \approx f'(x_n)$ jeweils ca. 1 voneinander entfernt.

(Führt man die Iteration dennoch ausreichend lange fort, so erreicht man in der Nähe der bei ca. 1,5 liegenden Lösung doch wieder eine schnelle Konvergenz.)

Der gewählte Startpunkt ist deshalb so schlecht, weil er in der Nähe des Minimums von $f(x)$ liegt, in dem die Tangente waagerecht und das Newtonverfahren damit nicht anwendbar ist: $f'(x) = e^x - 3 = 0$ gilt für $x = \ln 3 \approx 1{,}0986$.

c) Offensichtlich gilt $\lim\limits_{x\to-\infty} f(x)=\lim\limits_{x\to-\infty}(\mathrm{e}^x-3x)=0-(-\infty)=\infty$ und $\lim\limits_{x\to\infty} f(x)=\lim\limits_{x\to\infty}(\mathrm{e}^x-3x)=\infty$, da exponentielles sehr viel größer als lineares Wachstum ist.

Ferner ist $f'(x)=\mathrm{e}^x-3\begin{cases} <0 & x<\ln 3 \quad f(x) \text{ streng mon. fallend} \\ =0 & x=\ln 3 \\ >0 & x>\ln 3 \quad f(x) \text{ streng mon. wachsend}\end{cases}$ $\Longrightarrow$ Min. bei $f(\ln 3)\approx -0{,}2958$

Da die Funktion $f(x)$ zunächst von $+\infty$ zum negativen Minimum streng monoton fällt und von dort wieder streng monoton nach $+\infty$ wächst, hat sie genau 2 reelle Nullstellen, nämlich je eine links und rechts vom Minimum.

d) Wenn die zweite Nullstelle nicht schon bei b) ermittelt wurde, erhält man sie mit dem Newtonverfahren bei Wahl eines geeigneten Startwerts rechts von $\ln 3$, z.B. mit $x_0=2$:

n	x_n	$f(x_n)$	$f'(x_n)$
0	2,0000000000	1,3890560989	4,3890560989
1	1,6835182628	0,3339118655	2,3844666539
2	1,5434819721	0,0504146377	1,6808605540
3	1,5134886228	0,0020845605	1,5425504288
4	1,5121372501	0,0000041460	1,5364158963
5	1,5121345517	0,0000000000	1,5364036550
6	1,5121345517	0,0000000000	1,5364036550
7	1,5121345517	0,0000000000	1,5364036550

$\underline{\underline{x^*=1{,}5121345517}}$

---

2. Bei einem Preis von 1,20 € pro Liter werden in Deutschland 75 Millionen Liter Benzin pro Tag abgesetzt, die Elastizität der Nachfrage bezüglich des Preises betrage $-0{,}3$. Welche relative und welche absolute Entwicklung der Nachfrage ist ungefähr zu erwarten, wenn der Preis von 1,20 € auf 1,25 € pro Liter steigt?

Lösung:

$$\frac{\Delta p}{p}=\frac{1{,}25-1{,}20}{1{,}20}=\frac{5}{120}=\frac{1}{24}\approx 4{,}1667\,\%,\quad \varepsilon_N(1{,}20)=-0{,}3,$$

$$\frac{\Delta N}{N}=\varepsilon_N(1{,}20)\,\frac{\Delta p}{p}=-0{,}3\cdot\frac{1}{24}=-0{,}0125=-1{,}25\,\%,$$

$$\Delta N=\frac{\Delta N}{N}\cdot N=-1{,}25\,\%\ \cdot\ 75\,000\,000=-937\,500$$

Der Preisanstieg um 4,17 % führt zu einem Nachfragerückgang um ca. 1,25 %, das sind 937 500 Liter pro Tag.

---

3. Bestimmen Sie die folgenden Grenzwerte mit Hilfe der l'Hospitalschen Regel:

a) $\lim\limits_{x\to0}\dfrac{1-\cos 2x}{x^2}$, b) $\lim\limits_{x\to\infty}\dfrac{\ln x}{x}$, c) $\lim\limits_{x\to1}\dfrac{x^2+x-2}{x-1}$, d) $\lim\limits_{x\to\infty}\dfrac{\mathrm{e}^{2x}}{x^2-x}$!

Lösung:

a) $$\lim_{x\to0}\underbrace{\frac{1-\cos 2x}{x^2}}_{0/0}=\lim_{x\to0}\frac{2\sin 2x}{2x}=\lim_{x\to0}\underbrace{\frac{\sin 2x}{x}}_{0/0}=\lim_{x\to0}\frac{2\cos 2x}{1}=2$$

b) $\lim\limits_{x\to\infty} \underbrace{\frac{\ln x}{x}}_{\infty/\infty} = \lim\limits_{x\to\infty} \frac{\frac{1}{x}}{1} = 0$

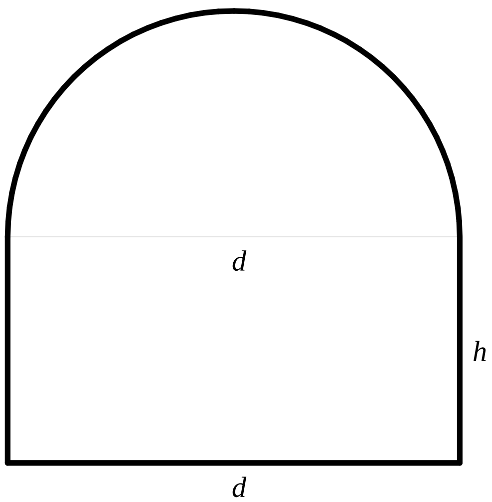
c) $\lim\limits_{x\to 1} \underbrace{\frac{x^2+x-2}{x-1}}_{0/0} = \lim\limits_{x\to 1} \frac{2x+1}{1} = 3$

d) $\lim\limits_{x\to\infty} \underbrace{\frac{e^{2x}}{x^2-x}}_{\infty/\infty} = \lim\limits_{x\to\infty} \underbrace{\frac{2e^{2x}}{2x-1}}_{\infty/\infty} = \lim\limits_{x\to\infty} \frac{4e^{2x}}{2} = 2\lim\limits_{x\to\infty} e^{2x} = \infty$

---

4. Der Querschnitt eines Tunnels habe die Form eines Rechtecks mit Grundseite d und Höhe h, auf das ein Halbkreis mit Durchmesser d aufgesetzt ist. Der Umfang des Querschnitts beträgt 20 m. Bestimmen Sie die Grundseitenlänge d, für die der Flächeninhalt des Querschnitts am größten wird!

Lösung:



Umfang $U = d + 2h + \frac{\pi}{2}d = 20,\;\; 2h = 20 - \left(1 + \frac{\pi}{2}\right)d,\;\; h = 10 - \frac{2+\pi}{4}d$

Fläche $F = dh + \frac{\pi}{8}d^2 = d\left(10 - \frac{2+\pi}{4}d\right) + \frac{\pi}{8}d^2 = 10d - \frac{4+2\pi-\pi}{8}d^2 = 10d - \frac{4+\pi}{8}d^2 \longrightarrow \min$

$F'(d) = 10 - \frac{4+\pi}{4}d = 0,\;\; d = \frac{40}{\pi+4} \approx 5{,}6$ [m], $F''(d) = -\frac{4+\pi}{4} < 0$, also Maximum

Somit wird für die Grundseitenlänge 5,6 m der maximale Tunnelquerschnitt erreicht.

5. Zum Zeitpunkt $t=0$ werden 1000 Bakterien in eine Nährlösung gegeben. Die Zahl der Bakterien entwickelt sich nach der Formel $f(t)=\dfrac{G}{1+Ae^{-0{,}2t}}$, wobei die Sättigungsgrenze bei 20 000 Bakterien liegt.

a) Bestimmen Sie die Parameter G und A !
b) Zeigen Sie, dass die Funktion $f(t)$ monoton wachsend ist!
c) Zu welchem Zeitpunkt beträgt die Zahl der Bakterien 10 000? Wie groß ist zu diesem Zeitpunkt die Wachstumsgeschwindigkeit der Population?
d) Zu welchem Zeitpunkt wächst die Population am stärksten?

Lösung:

a) $f(0)=\dfrac{G}{1+A}=1000,\quad \lim\limits_{t\to\infty} f(t)=G=20\,000,\quad \dfrac{20\,000}{1+A}=1000,\quad 1+A=20,\quad A=19$

Somit gilt $f(t)=\dfrac{20\,000}{1+19e^{-0.2t}}$.

b) Da die Exponentialfunktion überall streng monoton wachsend ist, ist $e^{-0.2t}$ und damit auch $1+19e^{-0.2t}$ streng monoton fallend, ihre Reziprokes und damit auch $f(t)$ wiederum überall streng monoton wachsend.

oder $f'(t)=\dfrac{-20\,000\cdot 19\left(-0.2\,e^{-0.2t}\right)}{(1+19e^{-0.2t})^2}=\dfrac{76\,000\,e^{-0.2t}}{(1+19e^{-0.2t})^2}>0$ überall,

also überall streng monoton wachsend.

c) $\dfrac{20\,000}{1+19e^{-0.2t}}=10\,000,\ 2=1+19e^{-0.2t},\ 1=19e^{-0.2t},\ e^{-0.2t}=\dfrac{1}{19},\ -0.2t=\ln\dfrac{1}{19}=-\ln 19,$

$t=\dfrac{\ln 19}{0.2}=\underline{\underline{5\ln 19\approx 14.72}}$

$f'(5\ln 19)=\dfrac{76\,000\,e^{-\ln 19}}{(1+e^{-\ln 19})^2}=\dfrac{76\,000\,\frac{1}{19}}{(1+\frac{19}{19})^2}=\dfrac{4000}{4}=1000$

Zum Zeitpunkt $5\ln 19$ wird eine Population von 10 000 erreicht, das Wachstum beträgt zu diesem Zeitpunkt 1000 pro Zeiteinheit..

d) Gesucht ist das Maximum von $f'(t)$:

$$f''(t)=76\,000\,\frac{-0.2e^{-0.2t}(1+19e^{-0.2t})^2-e^{-0.2t}\cdot 2(1+19e^{-0.2t})19(-0.2e^{-0.2t})}{(1+19e^{-0.2t})^4}$$
$$=15\,200\,e^{-0.2t}\,\frac{-(1+19e^{-0.2t})+38e^{-0.2t}}{(1+19e^{-0.2t})^3}=15\,200\,e^{-0.2t}\,\frac{19e^{-0.2t}-1}{(1+19e^{-0.2t})^3}=0$$

Also muss wie bei c) $19e^{-0.2t}-1=0$, d.h. $t=5\ln 19\approx 14.72$ sein.

Wegen des Faktors -0.2 im Exponenten ist die Exponentialfunktion monoton fallend, so dass gilt $f'(t)\begin{cases} >0 & t<5\ln 19: \quad f'(t) \text{ monoton wachsend} \\ <0 & t>5\ln 19: \quad f'(t) \text{ monoton fallend}\end{cases}$.

Somit liegt das Maximum von $f'(t)$ bei $t=5\ln 19\approx 14.72$, so dass zu diesem Zeitpunkt die Population am stärksten wächst.

6. Ein 30 Jahre alter Baum ist 11,52 m hoch und wächst in diesem Alter mit einer Geschwindigkeit von 48 cm/a (a: Jahr). Die Wachstumsgeschwindigkeit wächst ihrerseits um 1,2 cm/a^2.

a) Bestimmen Sie mithilfe der Taylorschen Formel näherungsweise die Höhe, die der Baum in einem Alter von 35 Jahren erreicht haben wird!

b) Auch die Änderung der Wachstumsgeschwindigkeit ist nicht konstant. Es wird aber angenommen, dass sie sich in der betrachteten Zeit um nicht mehr als 0,3 cm/a^3 ändert. Schätzen Sie unter dieser Annahme den Fehler des Ergebnisses von a) ab!

Lösung:

a) Wird die Höhe h in cm und die Zeit t in a angegeben, so gilt
$h(30) = 1152,\ \ h'(30) = 48,\ \ h''(30) = 1.2,$

also $h(35) \approx h(30) + h'(30)\,(35-30) + \dfrac{h''(30)}{2}\,(35-30)^2 = 1152 + 48\cdot 5 + 0.6\cdot 25 = 1407,$

der Baum wird also im Alter von 35 Jahren eine Höhe von ca. 14,07 m erreicht haben.

b) $R_3(35,30) = \dfrac{h'''(\tau)}{6}\,(35-30)^3,\ \ 30<\tau<35,$ also $|R_3(35,30)| \le \dfrac{0.3}{6}\,125 = 6.25,$

der Fehler ist also nicht größer als ca. 6,25 cm.

Zusatzaufgabe

Lösen Sie die folgenden Aufgaben mit `MATLAB`. Protokollieren Sie Ihr Vorgehen in einer `diary`-Datei und speichern Sie erstellte Plots ab.

1. Zeichnen Sie die Funktion $f(x) = \sin(x)$ und die Taylor-Polynome ungeraden Grades bis zur 11ten Ordnung mit der Entwicklungsstelle $x_0 = 0$ im Intervall $[-4\pi, 4\pi]$ in einen gemeinsamen Plot. Beschriften Sie die Achsen und erstellen Sie eine Legende. Verwenden Sie den Befehl `axis`, um einen angemessenen Bildausschnitt auszuwählen.
2. a) Implementieren Sie das Newton-Verfahren. Erstellen Sie dazu ein extra `m-File` und arbeiten Sie mit `function-handles`. Ihrer Funktion werden die folgenden Parameter übergeben
      - die Funktion F, auf die das Verfahren angewandt werden soll (in Form eines `function-handles`),
      - die Ableitung der Funktion F (in Form eines `function-handles`),
      - der Startwert x_0.

      Zurückgegeben werden soll ein Vektor $x = (x_0, x_1, \ldots, x_n)$, welcher die gesamte Iterationsfolge enthält. Verwenden sie $\varepsilon = 10^{-8}$ als Parameter für das Abbruchkriterium (Schritt 3 im Algorithmus).

   b) Wenden Sie Ihr `m-File` zum Newton-Verfahren auf die Funktion aus Hausaufgabenkomplex I, MATLAB-Aufgabe 2a) ($f(x) = \ln(x+1)\sin(x^2)$) mit den Startwerten
      i. x_0=`1.50`
      ii. x_0=`1.43`
      iii. x_0=`1.40`

      an. Zeichnen Sie jeweils die Funktion f und den Verlauf der Iterierten.

   c) Setzen Sie nun das Newton-Verfahren ein, um x mit $f'(x) = 0$ zu finden. Verwenden Sie die Startwerte
      i. x_0=`1.70`
      ii. x_0=`3.17`
      iii. x_0=`0.70`

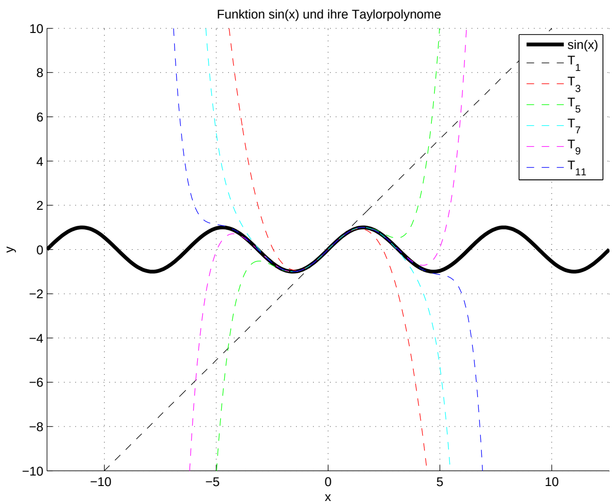
      und zeichnen Sie die Funktion f und f' sowie jeweils den Verlauf der Iterierten. Bestätigen Sie anhand der zweiten Ableitung, dass es sich bei der jeweils letzten Iterierten um ein lokales Maximum, ein lokales Minimum bzw. einen Sattelpunkt handelt.

Öffnen Sie die erstellte `diary`-Datei (vorher mit $\gg$ `diary off` die Protokollierung abschließen) und entfernen Sie ggf. überflüssige Zeilen (z.B. Fehleingaben). Drucken Sie anschließend die bearbeitete `diary`-Datei und eventuell angefertigte Plots und `m-Files` möglichst sparsam (d.h. nach Möglichkeit duplex, mehrere Seiten pro Blatt, kleine Schriftgröße) aus. Fügen Sie den Ausdruck Ihrer „restlichen" Hausaufgabe an.

Lösung:

nachbereitete `diary`-Datei (Kommentare durch `%` gekennzeichnet) und Plots auf den nächsten Seiten

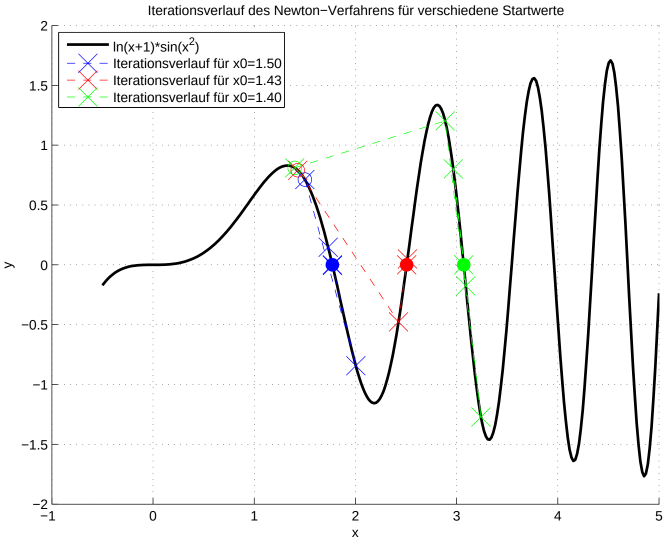
```
% ----------------------------------------------------------------------
% Aufgabe 1
% ----------------------------------------------------------------------

x=linspace(-4*pi,4*pi,300);
% Funktion sin auswerten
y=sin(x);

% Taylorpolynome auswerten
T1=x;
T3= T1 - 1/6*x.^3;
T5= T3 + 1/120*x.^5;
T7= T5 - 1/5040*x.^7;
T9= T7 + 1/362880*x.^9;
T11= T9 - 1/39916800*x.^11;

% Funktionen zeichnen
figure(1); clf; hold on;
plot(x,y,'k-','LineWidth',3);
plot(x,T1,'k--');
plot(x,T3,'r--');
plot(x,T5,'g--');
plot(x,T7,'c--');
plot(x,T9,'m--');
plot(x,T11,'b--');
% Dargestellten Bereich anpassen
axis([-4*pi,4*pi, -10,10])

% Label, Title, Legend
xlabel('x'); ylabel('y'); grid on;
title('Funktion sin(x) und ihre Taylorpolynome');
legend('sin(x)', 'T_1', 'T_3', 'T_5', 'T_7', 'T_9', 'T_{11}', 'Location',
       'NorthEast');
print -depsc HA02_matlab_plot_1.eps
```




```
% ----------------------------------------------------------------------
% Aufgabe 2
% ----------------------------------------------------------------------


% a) -------------------------------------------------------------------

% Listing des m-Files HA02_matlab_solution_Newton_Verfahren.m
type HA02_matlab_solution_Newton_Verfahren.m


function x=Newton_Verfahren(F, dF,  x0)

m=1;
x=x0;
fprintf('\n\n n    x_n             F(x_n)\n%2.0f%16.10f%16.10f\n',
        m-1,x(m),F(x(m)))

while abs(F(x(m))) > 10^-8,

    x(m+1) = x(m) - F(x(m)) / dF(x(m));
    fprintf('%2.0f%16.10f%16.10f\n',m,x(m+1),F(x(m+1)))
    m=m+1;

end


% ----------------------------------------------------------------------
% Der Einfachheit halber wurde in dem vorstehenden m-File auch die
% Ausgabe der Iterationsschritte mit dem fprintf-Befehl realisiert.
%
% Wenn ein solches m-File für aufwändigere Berechnungen genutzt wird,
% sollte dieses möglichst keine Ausgabebefehle für Zwischenschritte
% enthalten.
% ----------------------------------------------------------------------
```

```
% b) ---------------------------------------------------------------------

% Listing des m-Files HA01_matlab_solution_f.m für die Funktion f
type HA01_matlab_solution_f.m


function y=HA01_matlab_solution_f(x)
   y=log(x+1).*sin(x.^2);

% ------------------------------------------------------------------------

% Listing des m-Files HA01_matlab_solution_df.m für die Ableitung von f
type HA01_matlab_solution_df.m


function y=HA01_matlab_solution_df(x)
    y = sin(x.^2)./(x+1) + 2*x.*log(x+1).*cos(x.^2);

% ------------------------------------------------------------------------

% 1. Startwert setzen
x0=1.50;
% Newton-Verfahren auf Funktion f(x) = ln(x+1) * sin(x^2) anwenden
x_newton_1=HA02_matlab_solution_Newton_Verfahren( @HA01_matlab_solution_f,
                                             @HA01_matlab_solution_df , x0);
 n     x_n              F(x_n)
 0     1.5000000000     0.7129412590
 1     2.0036533031    -0.8427791426
 2     1.7306319547     0.1466439023
 3     1.7739377079    -0.0053689696
 4     1.7724552348    -0.0000050025
 5     1.7724538509    -0.0000000000

% 2. Startwert setzen
x0=1.43;
% Newton-Verfahren auf Funktion f(x) = ln(x+1) * sin(x^2) anwenden
x_newton_2=HA02_matlab_solution_Newton_Verfahren( @HA01_matlab_solution_f,
                                             @HA01_matlab_solution_df , x0);
 n     x_n              F(x_n)
 0     1.4300000000     0.7899588841
 1     2.4259215887    -0.4773508498
 2     2.5144168797     0.0491399509
 3     2.5066500514     0.0001369752
 4     2.5066282748     0.0000000013

% 3. Startwert setzen
x0=1.40;
% Newton-Verfahren auf Funktion f(x) = ln(x+1) * sin(x^2) anwenden
x_newton_3=HA02_matlab_solution_Newton_Verfahren( @HA01_matlab_solution_f,
                                             @HA01_matlab_solution_df , x0);
 n     x_n              F(x_n)
 0     1.4000000000     0.8099937619
 1     2.8871820468     1.2031045389
 2     3.2405112138    -1.2715088923
 3     2.9672152060     0.8011657528
 4     3.0903309575    -0.1761336286
 5     3.0700121012    -0.0002755927
 6     3.0699801242    -0.0000000030
```

```
x=linspace(-0.5,5,300);
% Funktion auswerten
y=HA01_matlab_solution_f(x);
y_newton_1=HA01_matlab_solution_f(x_newton_1);
y_newton_2=HA01_matlab_solution_f(x_newton_2);
y_newton_3=HA01_matlab_solution_f(x_newton_3);

% Funktion zeichnen
figure(2); clf; hold on;
plot(x,y,'k-','LineWidth',2);
% Iterationsverläufe einzeichnen
plot(x_newton_1, y_newton_1,'xb--','MarkerSize',20);
plot(x_newton_2, y_newton_2,'xr--','MarkerSize',20);
plot(x_newton_3, y_newton_3,'xg--','MarkerSize',20);
% Startwerte hervorheben
plot(x_newton_1(1), y_newton_1(1),'ob','MarkerSize',10);
plot(x_newton_2(1), y_newton_2(1),'or','MarkerSize',10);
plot(x_newton_3(1), y_newton_3(1),'og','MarkerSize',10);
% Letzte Iterierte hervorheben
plot(x_newton_1(end), y_newton_1(end),'.b','MarkerSize',30);
plot(x_newton_2(end), y_newton_2(end),'.r','MarkerSize',30);
plot(x_newton_3(end), y_newton_3(end),'.g','MarkerSize',30);

% Label, Title, Legend
xlabel('x'); ylabel('y'); grid on;
title('Iterationsverlauf des Newton-Verfahrens für verschiedene
Startwerte');
legend('ln(x+1)*sin(x^2)', 'Iterationsverlauf für x0=1.50', '
Iterationsverlauf für x0=1.43', 'Iterationsverlauf für x0=1.40', '
Location', 'NorthWest');
print -depsc HA02_matlab_plot_2b.eps
```



```
% c) -------------------------------------------------------------------

% Listing des m-Files HA02_matlab_solution_ddf.m für die 2. Abl. von f
type HA02_matlab_solution_ddf.m

function y=HA02_matlab_solution_ddf(x)
   y = -sin(x.^2) ./ (x+1).^2 + 4*cos(x.^2).*x ./ (x+1)
                 - 4*log(x+1).*sin(x.^2).*x.^2 + 2*log(x+1).*cos(x.^2);

% ----------------------------------------------------------------------

% 1. Startwert setzen
x0=1.70;
% Newton-Verfahren auf die Ableitung der Funktion f(x)= ln(x+1) * sin(x^2)
% anwenden
x_newton_1=HA02_matlab_solution_Newton_Verfahren(@HA01_matlab_solution_df,
                                           @HA02_matlab_solution_ddf , x0);
 n    x_n            F(x_n)
 0    1.7000000000   -3.1785342143
 1    1.2619286162    0.3973648832
 2    1.3344789358   -0.0528211827
 3    1.3268414970   -0.0005550841
 4    1.3267595102   -0.0000000645
 5    1.3267595006   -0.0000000000

% 2. Startwert setzen
x0=3.17;
% Newton-Verfahren auf die Ableitung der Funktion f(x)= ln(x+1) * sin(x^2)
% anwenden
x_newton_2=HA02_matlab_solution_Newton_Verfahren(@HA01_matlab_solution_df,
                                           @HA02_matlab_solution_ddf , x0);
 n    x_n            F(x_n)
 0    3.1700000000   -7.4864270413
 1    3.4300358051    6.9426812567
 2    3.3028109460   -1.0692309434
 3    3.3197918903    0.0157454002
 4    3.3195484906    0.0000023162
 5    3.3195484548    0.0000000000

% 3. Startwert setzen
x0=0.70;
% Newton-Verfahren auf die Ableitung der Funktion f(x)= ln(x+1) * sin(x^2)
% anwenden
x_newton_3=HA02_matlab_solution_Newton_Verfahren(@HA01_matlab_solution_df,
                                           @HA02_matlab_solution_ddf , x0);
 n    x_n            F(x_n)
 0    0.7000000000   0.9323057960
 1    0.1633666388   0.0723612045
 2    0.0776187517   0.0171950662
 3    0.0378533558   0.0041934572
 4    0.0186937033   0.0010354985
 5    0.0092893247   0.0002572838
 6    0.0046303689   0.0000641232
 7    0.0023116220   0.0000160061
 8    0.0011549218   0.0000039985
 9    0.0005772388   0.0000009992
10    0.0002885639   0.0000002498
11    0.0001442681   0.0000000624
12    0.0000721306   0.0000000156
13    0.0000360644   0.0000000039

x=linspace(-0.5,5,300);
% Funktion auswerten
y=HA01_matlab_solution_f(x);
y_newton_1=HA01_matlab_solution_f(x_newton_1);
y_newton_2=HA01_matlab_solution_f(x_newton_2);
y_newton_3=HA01_matlab_solution_f(x_newton_3);

% Funktion zeichnen
figure(3); clf; hold on;
plot(x,y,'k-','LineWidth',2);
% Iterationsverläufe einzeichnen
plot(x_newton_1, y_newton_1,'xb--','MarkerSize',20);
plot(x_newton_2, y_newton_2,'xr--','MarkerSize',20);
plot(x_newton_3, y_newton_3,'xg--','MarkerSize',20);
% Startwerte hervorheben
plot(x_newton_1(1), y_newton_1(1),'ob','MarkerSize',10);
plot(x_newton_2(1), y_newton_2(1),'or','MarkerSize',10);
plot(x_newton_3(1), y_newton_3(1),'og','MarkerSize',10);
% Letzte Iterierte hervorheben
plot(x_newton_1(end), y_newton_1(end),'.b','MarkerSize',30);
plot(x_newton_2(end), y_newton_2(end),'.r','MarkerSize',30);
plot(x_newton_3(end), y_newton_3(end),'.g','MarkerSize',30);

% Label, Title, Legend
xlabel('x'); ylabel('y'); grid on;
title('Iterationsverlauf des Newton-Verfahrens für Nullstellen der
Ableitung für verschiedene Startwerte');
legend('ln(x+1)*sin(x^2)', 'Iterationsverlauf für x0=1.70', '
Iterationsverlauf für x0=3.17', 'Iterationsverlauf für x0=0.70', '
Location', 'NorthWest');
print -depsc HA02_matlab_plot_2c1.eps

% Analog für die Ableitung von f
% Ableitung auswerten
dy=HA01_matlab_solution_df(x);
dy_newton_1=HA01_matlab_solution_df(x_newton_1);
dy_newton_2=HA01_matlab_solution_df(x_newton_2);
dy_newton_3=HA01_matlab_solution_df(x_newton_3);

% Ableitung zeichnen
figure(4); clf; hold on;
plot(x,dy,'k-','LineWidth',2);
% Iterationsverläufe einzeichnen
plot(x_newton_1, dy_newton_1,'xb--','MarkerSize',20);
plot(x_newton_2, dy_newton_2,'xr--','MarkerSize',20);
plot(x_newton_3, dy_newton_3,'xg--','MarkerSize',20);
% Startwerte hervorheben
plot(x_newton_1(1), dy_newton_1(1),'ob','MarkerSize',10);
plot(x_newton_2(1), dy_newton_2(1),'or','MarkerSize',10);
plot(x_newton_3(1), dy_newton_3(1),'og','MarkerSize',10);
% Letzte Iterierte hervorheben
plot(x_newton_1(end), dy_newton_1(end),'.b','MarkerSize',30);
plot(x_newton_2(end), dy_newton_2(end),'.r','MarkerSize',30);
plot(x_newton_3(end), dy_newton_3(end),'.g','MarkerSize',30);
```

```
% Label, Title, Legend
xlabel('x'); ylabel('y'); grid on;
title('Iterationsverlauf des Newton-Verfahrens für Nullstellen der
Ableitung für verschiedene Startwerte');
legend('Ableitung von f(x)=ln(x+1)*sin(x^2)', 'Iterationsverlauf für x0=1.
70', 'Iterationsverlauf für x0=3.17', 'Iterationsverlauf für x0=0.70', '
Location', 'NorthWest');
print -depsc HA02_matlab_plot_2c2.eps


% Zweite Ableitungen an den letzten Iterierten bestimmen

HA02_matlab_solution_ddf(x_newton_1(end))
ans =
   -6.7688
% Zweite Ableitung negativ ==> lokales Maximum

HA02_matlab_solution_ddf(x_newton_2(end))
ans =
   64.6704
% Zweite Ableitung positiv ==> lokales Minimum

HA02_matlab_solution_ddf(x_newton_3(end))
ans =
  2.1638e-004
% Zweite Ableitung nahe Null ==> möglicherweise Sattelpunkt

diary off
```