EDV WS17/18 Abschlussbericht

Physik131

Shazam Ali Shah
Matrikelnummer: 3143817

25. August 2018

Inhaltsverzeichnis

Teil I.

LaTeX

1. Der kleine Hobbit

Es folgen ein paar kurze Auszüge aus dem kleinen Hobbit (siehe [3] für das englische Original) [1]. Abschnitt 1.1.1 beschreibt eine Hobbithöhle und Bilbos erste Begegnung mit Gandalf. In Abschnitt 1.2 geht es um das Gelage der Zwerge bei Bilbo.

1.1. Eine unvorhergesehene Gesellschaft

1.1.1. Die Hobbithöhle

Abbildung 1.1.: Eine typische Hobbithöhle

In einer Höhle in der Erde, da lebte ein **Hobbit**. Nicht in einem schmutzigen, nassen Loch, in das die Enden von irgendwelchen Würmern herabbaumelten und das nach Schlamm und Moder roch. Auch nicht etwa in einer trockenen Kieshöhle, die so kahl war, dass man sich nicht einmal niedersetzen oder gemütlich frühstücken konnte. Es war eine Hobbithöhle (siehe Abbildung 1.1), und das bedeutet Behaglichkeit. Diese Höhle hatte eine kreisrunde Tür wie ein Bullauge. Sie war grün gestrichen, und in der Mitte saß ein glänzend gelber Messingknopf. Die Tür führte zu einer röhrenförmig langen Halle, zu einer Art Tunnel, einem Tunnel mit getäfelten Wänden.

[1] Ich bin mir aber nicht sicher, ob wir unseren Text aus der Übersetzung von Walter Scherf [2] oder aus der von Wolgang Krege [1] genommen haben.

1.1.2. Ein guter Morgen (?)

Abbildung 1.2.: Der Zauberer Gandalf taucht eines Morgens unerwartet bei Bilbo auf.

Alles, was also der keineswegs misstrauische Bilbo an diesem Morgen sah, war ein kleiner, alter Mann mit einem Stab, hohem, spitzem blauen Hut, einem langen, grauen Mantel, mit einer silbernen Schärpe, über die sein langer, silberner Bart hing, ein kleiner, alter Mann mit riesigen schwarzen Schuhen. „Guten Morgen“, sagte Bilbo, und er meinte es ehrlich. Die Sonne schien, und das Gras war grün. Aber Gandalf schaute ihn scharf unter seinen buschigen Augenbrauen hervor an. „Was meint Ihr damit?“ fragte er.

- Wünscht Ihr mir einen guten Morgen?
- Oder meint Ihr, dass dies ein guter Morgen ist, gleichviel, ob ich es wünsche oder nicht?
- Meint Ihr, dass Euch der Morgen gut bekommt?
- Oder dass dies ein Morgen ist, an dem man gut sein muss?

„Alles auf einmal“ ,sagte Bilbo.

1.2. Die Zwerge

Abbildung 1.3.: Zwei der Zwerge, die bei Bilbo einen feuchtfröhlichen Nachmittag mit viel Musik verbrachten

Bevor Gandalf am folgenden Nachmittag erschien, wurde der arme Hobbit von 13 ungebetenen Gästen, es waren Zwerge, heimgesucht. Ihre Namen finden sich, zusammen mit einigen Zusatzinformationen, in Tabelle 1.1. Ein Bild von zwei Zwergen ist in Abbildung 1.3 zu sehen.

Tabelle 1.1.: Namen und besondere Merkmale der 13 Zwerge, die den Hobbit heimgesucht haben.

Name	Bart	Kleidung		Instrument	Sonstiges
		Gürtel	Kapuze		
Dwalin	blau	gold	dunkelgrün	Bratsche	
Balin	weiß	pupurrot	Bratsche		
Kili	gelb	silber	blau	Fiedel	Werkzeug
Fili	gelb	silber	blau	Fiedel	Spaten
Dori	gold	purpurrot	Flöte		
Nori	gold	purpurrot	Flöte		
Ori	gold	grau	Flöte		
Oin	silber	braun			
Gloin	silber	silber			
Bifur	gelb			Klarinette	
Bofur	gelb			Klarinette	
Bombur	blassgrün			Trommel	fett
Thorin	himmelblau mit silberner Schärpe			Harfe	sehr berühmt

2. Mathematik

2.1. Mathematikformeln des zweiten Übungsblattes

2.1.1. Brüche

Für Abbildungen an Sammellinsen gilt

$$\frac{1}{g} + \frac{1}{b} = \frac{1}{f} \text{ und } \frac{B}{G} = \frac{b}{g} = \gamma$$

2.1.2. Vektoren

Das Skalarprodukt zwischen 2 Vektoren ist:

$$\vec{a} \cdot \vec{b} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{2.1}$$

$$= |\vec{a}| \cdot |\vec{b}| \cdot \cos(\alpha) \tag{2.2}$$

2.1.3. Ableitung nach der Zeit

Die Differentialgleichung für die freie, ungedämpfte Schwingung lautet:

$$\ddot{x} + \omega_0^2 x = 0$$

2.2. Mathematikformeln des dritten Übungsblattes

2.2.1. Summen, Wurzeln und Brüche

Mittelwert $\bar{x}$ und Standardabweichung σ_x einer Reihe von N Messwerten $x_1, ...,_N$ berechnen sich als

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

und

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N(N-1)}}$$

2.2.2. Partielle und totale Ableitungen

$$\frac{\mathrm{d}}{\mathrm{d}t} f(x(t), t) = \frac{\partial f}{\partial x} \cdot \frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial t}$$

2.2.3. Integral

Das Trägheitsmoment θ eines starren Körpers berechnet man gemäß

$$\theta = \int_V r^2 \mathrm{d}m$$

2.2.4. Komplexere Formeln (I)

Mit fortgesetzter Anwendung der Regel von l'Hôspital erhalten wir:

$$\begin{aligned}
\lim_{x \to 0} \frac{\ln \sin(\pi x)}{\ln \sin(x)} &= \lim_{x \to 0} \frac{\pi \frac{\sin(\pi x)}{\cos(\pi x)}}{\frac{\sin(x)}{\cos(x)}} \\
&= \lim_{x \to 0} \frac{\tan(\pi x)}{\tan(x)} \\
&= \lim_{x \to 0} \frac{\pi / \cos^2(x)}{\pi / \cos^2(\pi x)} \\
&= 1
\end{aligned}$$

2.2.5. Komplexere Formeln (II)

Ein lineares Gleichungssystem $A \cdot x = b$, wobei $A = (a_{ij})_{n \times n}$ eine $n \times n$ Matrix und $x = (x_i)_n$ und $b = (b_i)_n$ Vektoren mit n Elementen sind, sieht ausgeschrieben so aus:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

Die Summenschreibweise für die i. te Zeile dieser Matrix ist dann:

$$\sum_{m=1}^{n} a_{im} \cdot x_m = b_i.$$

3. Typographie und LaTeX-Quiz

3.1. Typographie

3.1.1. Text mit typographische Fehler

Das Defizit von -7 db, welches in Gl. 7-9 diskutiert wird, kann auf das g(t)= $sin(2\pi ft)$Signal zurückgeführt werden. Hierbei ist t die Zeit und f =48Khz die “Abtastrate”.

3.1.2. Fehler in diesem Text

1. „-7 db“:
   Die beschriebenen -7 Dezibel wurden ohne die Umgebung der SI-Einheiten geschrieben. Dadurch ist die Einheit Dezibel unformatiert und falsch ausgegeben worden.

2. „7-9“:
   Hier wurde der Zahlenbereich 7 bis 9 nicht in die Mathe-Umgebung gesetzt.

3. „g(t)= “:
   Dieser Teil der Funktion wurde nicht in dei Mathe-Umgebung gesetzt und dadurch sind Fehler in der Schreibweise und der Leerzeichensetzung entstanden.

4. „$sin(2\pi ft)$“:
   In diesem Teil der Funktion wurde Sinus ohne Weiteres geschrieben. Deshalb ist die Ausgabe des Sinus falsch. Um die richtige Ausgabe von Sinus zu erhalten, ist es wichtig, dass man das Backslash vor dem sin nicht vergisst.

5. „t“:
   Da sich dieses t auf dem t, welches in der Funktion verwendet wird, bezieht, muss dieses t ebenfalls in der Mathe-Umgebung geschrieben werden.

6. „f =48Khz“:
   In diesem Abschnitt, in welchen der HiWi die Frequenz beschreibt kommen sogar zwei Fehler vor, da hier nicht nur f = und 48Khz einzeln geschrieben wurden, sondern auch vergessen wurde, dass Khz in der Umgebung der SI-Einheiten geschrieben werden sollte.

7. „48Khz“:
   Wie oben bereits geschildert, wurde dieser Abschnitt nicht in die SI-Umgebung gesetzt. Dadurch wie hier das K für Kilo, groß geschrieben, obwohl es klein geschrieben werden sollte. auch das hz ist falsch geschrieben worden da das H in der Einheit Hertz großgeschrieben wird.

8. „“Abtastrate”“:
   Die Abtastrate wird hier nicht in der enquote-Umgebung gesetzt, obwohl dies hier die Intention des Autors zu sein scheint.

3.1.3. Text ohne typographische Fehler

Das Defizit von $-7\,\mathrm{dB}$, welches in Gl. $7-9$ diskutiert wird, kann auf das $g(t) = \sin(2\pi f t)$ Signal zurückgeführt werden. Hierbei ist t die Zeit und $f = 48\,\mathrm{kHz}$ die „Abtastrate“.

3.2. LaTeX-Quiz

1. Was versteht man unter der Latex Präambel?
   Antwort: Die Präambel ist die Spezifikation zu dem darauffolgenden Dokument. Hier werden die Angaben zum Dateitypen und anderen Informationen gemacht, mithilfe wessen das Dokument korrekt formatiert wird. Benötigte Pakete werden auch hier importiert.

2. *section*: Nummeriert das Kapitel entsprechend der bisherigen Kapiteln. *section*$*$ gibt den Kapitelnamen ohne Nummerierung an.

3. Alternative zu der Matheumgebung ist die Umgebung $...$

4. Tabular-Umgebung gibt das Format der Tabelle an. Ob die Tabelle Linien besitzt, links-rechtsbündig, zentriert oder wie breit ist, wird in der *tabular*-Umgebung bestimmt. die *table*−Umgebung ist für die Positionierung der kompletten Tabelle im Dokument zuständig. Fehlt die *table*−Umgebung, so wird die Tabelle im Dokument so gesetzt, wie diese passt.

5. *htbp* ist eine Abkürzung für 4 Positionen nach Priorität der Position. Zunächst wird versucht, die Abbildung an der Stelle einzufügen, in der es eingeführt wird. h steht für *here*. Falls das nicht funktioniert, wird das Bild am Anfang der Datei eingefügt. t steht für *top*. Sollte dies auch nicht funktionieren wird es am ende der aktuellen Seite eingefügt. b steht für bottom. Falls nichts funktioniert, wird das Bild auf die nächste bzw auf einer speziellen Seite nur für floats eingefügt. p steht für page.

6. Mit *cite* kann man Dokumente zitieren, welche in der bibliography angegeben werden.

7. Das gewöhnliche Leerzeichen ist ein ungesichertes Leerzeichen, wohingegen das mit \, gesetzte Leerzeichen ein geschütztes Leerzeichen ist.

8. Latex neigt dazu doppelte Bindestriche als ein Bindestrich auszugeben. Eine Methode, dies zu verhindern, ist die Bindestriche in n, ist eine Mathe-Umgebung zu setzen. So entsteht das Konstrukt: $--$.

9. hypcap ist ein Package, welches für das Integrieren von Hyperlinks verantwortlich ist.

Teil II.

Linux

4. Linux Quiz

1. „Wie finden Sie heraus, in welchem Verzeichnis Sie sich gerade befinden?“

   Über den Befehl **pwd** kann man herausfinden, in welchen Verzeichnis man sich befindet. Dabei wird der gesamte Pfad bishin zu diesem Verzeichnis angegeben.

2. „Wie finden Sie den Namen Ihres Rechners heraus?“

   Eine der Möglichkeiten den Rechnernamen zu ermitteln, ist der Linux Befehl **hostname**.

3. „Wie finden Sie heraus, wer gerade (außer Ihnen selber) auf Ihrem Rechner angemeldet ist?“

   Es ist möglich durch den Befehl **who** herauszufinden, wer außer mir auf meinem Rechner angemeldet ist.

4. „Wie finden Sie heraus, wie groß eine bestimmte Datei ist?“

   Um die Speichernutzung eines Verzeichnisses aufzurufen, wird der Befehl **du** also disk usage genutzt.

5. „Sie wollen eine Datei test.txt von /home/someuser/uebung_01 nach /home/someuser kopieren. Sie befinden sich in /home/someuser/uebung_01. Geben Sie zwei cp Befehle an, die dies erledigen; einmal soll das Ziel als relativer und einmal als absoluter Pfad angegeben werden.“

   Die Methode über einen relativen Pfad die test.txt Datei zu kopieren, nutzt die Tatsache, dass man die Datei in das übergeordnete Verzeichnis kopiert. Somit braucht man nur in das übergeordnete Verzeichnis zu gehen und die Datei dort zu kopieren. Dies sieht wie folgt aus: **cp test.txt ../test.txt**. Die zweite Methode die test.txt-Datei zu kopieren, ist den absoluten Pfad anzugebender wie folgt lautet: **cp test.txt / home/someuser/test.txt**

6. „Was bewirken die Zeichen * und ? in Verbindung mit Shell-Kommandos?“

   Das Zeichen * wird in Kombination mit einer Dateiendung oder einem Dateinamen verwendet. Es wird benutzt, um alle Dateien desselben Dateitypen auszugeben, mit z.B. **find someuser/ *tex**, oder verschiedene Dateitypen mit denselben Dateinamen auszugeben, mit z.B. **find someuser/Test***

   Das Fragezeichen ? gibt alle Dateien aus, die die Länge der Anzahl der Fragezeichen entsprechen. Zum Beispiel wird die Suche nach aa.txt durch **ls ??.txt** oder **ls ??????** initialisiert. Diese Art der Suche hilft allerdings nur dann, wenn man die Länge der Datei kennt.

7. „Was bewirken die Zeichen > und | in Verbindung mit Shell-Kommandos?“

   Das Zeichen | verbindet Ausgabe des ersten Befehls mit der Eingabe vom zweiten Befehl. Zum Beispiel wird bei dem Befehl **cat test.txt | grep testWord** testWord aus test.txt gesucht und ausgegeben. Falls testWord in test.txt nicht enthalten ist, wird es nicht ausgegeben.

   Das Zeichen > lenkt Ausgaben, die auf dem Terminal gemacht werden auf gegebenen Dateien um. Zum Beispiel wird bei dem Befehl **echo testWord > test.txt** testWord in die neue Datei test.txt geschrieben.

8. „Wie lauten die Befehle, um sich auf einem entfernten Rechner einzuloggen bzw. um Dateien dorthin (oder von dort) zu kopieren?“

   Um sich auf entfernten Servern einzuloggen und um auf diese zuzugreifen, bedient man sich der secure shell.So kann man sich über den Befehl **ssh username@server** auf ein entfernten Server zugreifen.

   Um eine Datei von einem entfernten Server auf seinen eigenen Rechner zu kopieren, verwendet man den scp (secure copy-Befehl). So kann man als Beispiel eine Datei mit dem Befehl **scp username@ausgangsrechner:/someuser/ausgangsordner/datei username@zielrechner:/someuser/zielordner** auf einem anderen Rechner kopiert werden.

9. „Sie werden gefragt, was die Option −s bei dem Befehl ls bewirkt? Wie finden Sie das heraus, falls Sie es nicht wissen?“

   Durch den Befehl **ls −−help**, ist es möglich, nicht nur die Funktionsweise der Option −s herauszufinden, sondern auch die Funktionsweise aller anderen Optionen, welche mit ls in Verbindung stehen, herauszufinden.

10. „Mit welchem Befehl können Sie bestimmte Spalten aus einer Textdatei extrahieren?“

    Der Befehl, um eine Spalte aus einer Textdatei zu extrahieren,ist **awk**. So kann man mit
    awk ’print $ <Spaltennummer>’<Dateiname>
    awk ’print $ 1 testfile
    die gewählte Spalte der Datei ’Dateiname’ extrahieren.

11. „Was ist der Unterschied zwischen den Befehlen: cd ~terben und cd ~/terben?“

    Der Befehl ~terben wechselt zum Verzeichnis namens ’~terben’.

    Der Befehl ~/terben wechselt zum Verzeichnis namens ’~terben’, solange dieser im Heimverzeichnis liegt.

12. „Sie befinden sich im Ihrem Homeverzeichnis. Was ist der Unterschied zwischen den Befehlen: rm -rf ./uebung/* und rm -rf ./uebung/ *“

    rm -rf ./uebung/* löscht alle Inhalte vom Verzeichnis uebung/ samt der verschachtelten

Verzeichnissen.

rm -rf ./uebung/ * entfernt das Verzeichnis ./uebung/ und alle Inhalte des aktuellen Verzeichnisses.

13. „Angenommen, der Systemverwalter hat sich einen Spaß erlaubt und das ls-Kommando gelöscht. Wie können Sie trotzdem eine Liste der Dateien im gegenwärtigen Verzeichnis bekommen?“

    Der Befehl **echo *** gibt Inhalte des aktuellen Verzeichnisses aus.
    Durch den Befehl **find *** könnte man dann alle Unterverzeichnisse in dem aktuellen Verzeichnis finden.

14. „Geben Sie eine Kommandofolge an (Pipes!), mit der Sie eine alphabetisch sortierte Liste aller am System angemeldeten Benutzer bekommen. Jeder Benutzername soll in Ihrer Liste nur einmal vorkommen“

    awk -F':' 'print $ 1' /etc/passwd | sort | uniq ist eine solche Kommandofolge, welche zum Ziel führt.

15. „Sie wollen die drei größten Dateien in dem Verzeichnisbaum startend von Ihrem Homeverzeichnis wissen. Geben Sie eine Befehlsfolge an, die dies leistet.“

    ls -lSR | sort -k 5 -n -r | head -3 gibt die drei größten Dateien in dem Verzeichnisbaum startend von Ihrem Homeverzeichnis. Dabei listet **ls -lSR** die Dateien rekursiv nach Größe der Datei auf, der Befehl **sort -k 5 -n -r** sortiert die gelisteten Dateien nach der Größe von groß nach klein und der Befehl **head -3** gibt die ersten drei Kopfzeilen, also die drei größten Dateien aus.

16. „Jemand hat eine Datei test.txt mit dem Inhalt:
    .
    .
    | a | b | c | d |
    | d | e | f | g |
    .
    .
    Er möchte Zeilen mit dem Muster „f | “ mit einem grep Befehl finden und führt dafür
    user$ grep f | test.txt
    test.txt: command not found
    aus. Erklären Sie, wie und warum es zu der Fehlermeldung test.txt: command not found kommt. Geben Sie eine korrekte Version des Befehls an. “

17. „Erklären Sie, warum folgender Befehl zu einer Fehlermeldung führt:
    user$ echo “Text in der Log-Datei “ > \
    log_ { $ (date +' %D').txt

bash: log_ { $ (date +’ %D’).txt: No such file ... “

18. „Sie wollen den gesamten Verzeichnisbaum unter Ihrem Home in einer tar Datei archivieren und mit gzip packen. Die Datei soll als Backup in Ihr Homeverzeichnis auf dem Rechner cosmix.someuni.com übertragen werden. Der Benutzername auf cosmix.someuni.com sei identisch mit ihrem Login auf den Rechnern des CIP-Pools der Physik. Geben Sie die nötigen Befehle hierzu an.“

5. Pipeline Analyse

user$ cat mobydick.txt | tr -cs '[:alpha:]' '\n' | sort |\uniq -c | sort -nr -k1,1

1. **cat mobydick.txt** gibt den Inhalt der Datei mobydick.txt auf dem Terminal aus bzw. gibt es als Ausgabe für den nächsten Befehl als Eingabe.

2. **tr -cs '[:alpha:]' '\n'** tr ersetzt Zeichen systematisch durch andere Zeichen, welche mit den Optionen angegeben werden. **[:apha:]** ist eine davon. Die Option -cs benutzt Komplement der angegeben Buchstaben und ersetzt sich wiederholende Buchstaben. **\n** startet dann bei jedem neuen Wort eine neue Zeile.

3. **sort** sortiert die gegebenen Wörter nach Buchstaben von groß z nach a.

4. **\uniq -c** zählt die Anzahl der vorgekommenen Wörter

5. **sort -nr -k1,1** sortiert numerisch von groß nach klein.

6. Shell-Skripte

```
#!/bin/bash

set -v        # zeigt die auszufuehrenden Kommandos an
set -x        # zeigt die ausgefuehrten Kommandos an
              # ist hilfreich wenn das Programm Prob-
              # leme macht Debug infos werden angezeigt


$SOURCE = "D:\Uni"
$RECEIVER = "C:\Users\Uzaira846\Desktop\shazam"

$DATE = $(date +%y-%m-%d)

touch backup_$DATE.tgz          # dateien erstellen
touch backup-$DATE.log

chmod u+x backup_$DATE.tgz      # zur vereinfachung read
                                # write- und execute-Rechte allen geben
chmod u+x backup-$DATE.log      # damit aenderungen der datei moeglich sind

mkdir backup-$DATE
tar -ucpf $SOURCE > $RECEIVER\backup-$DATE
echo "Es wurde "+$SOURCE+"-Ordner"+ "gesichert" > backup-$DATE.log
```

Teil III.

Python

7. Projekt Euler - Problem 14

1. Version der Collatz-Sequenz:

```
# the function to determine the collatz sequence and its length
def collatz(n):
length = 0
while (n > 1):
    # c. s. uses operations to reduce an integer
    # until it reaches 1

    # length counts the number of operations until
    # 1 is reached (end of collatz sequence)
    length += 1
    #print(n)
    if n % 2 == 1:
        n = 3*n+1
        continue
    n = n/2
    # print     length of the collatz seq.
return length


def max_collatz(r):
    l= [0]
    for k in range(r):
        l.append(collatz(k))
    value = max(l)
    index = l.index(value) - 1
    print(index, ": ", value)

print (max_collatz(1000000))
```

2. (optimierte) Version der Collatz-Sequenz:

```
def coll(limit):
    from array import array
    maximum = 0
    known = array("L", (0 for i in range(limit)))
    for num in range(2, limit):
        steps = known[num]
        if steps:
            if steps > maximum:
                print(num, "\t", steps)
                maximum = steps
        else:
            start_num = num
            steps = 0
            while num != 1:
                if num < start_num:
                    steps += known[num]
                while num & 1:
                    num += (num >> 1) + 1
                    steps += 2
                while num & 1 == 0:
                    num >>= 1
                    steps += 1
            if steps > maximum:
                print(start_num, "\t", steps)
                maximum = steps
            while start_num < limit:
                known[start_num] = steps
                start_num <<= 1
                steps += 1

coll(1000000)
```

Quelle: https://stackoverflow.com/questions/38885614/longest-collatz-or-hailstone-sequence-optimization-python-2-7

8. Aufgabe 10 - Flächen konvexer Polygone

```
import numpy as np

def triangle_area(B):
    #implemented according to Heron's formula
    #https://en.wikipedia.org/wiki/Heron%27s_formula
    length = len(B)
    if length<3:
        return 0
    ax=B[1][0]-B[0][0]
    dx=B[2][0]-B[1][0]
    cx=B[0][0]-B[2][0]
    ay=B[1][1]-B[0][1]
    dy=B[2][1]-B[1][1]
    cy=B[0][1]-B[2][1]
    a = np.sqrt((ax*ax)+(ay*ay))
    d = np.sqrt((dx*dx)+(dy*dy))
    c = np.sqrt((cx*cx)+(cy*cy))
    s=((a+d+c)/2)
    return np.sqrt([s*(s-a)*(s-d)*(s-c)])

def polygon_area(B):
    length = len(B)
    if length<3:
        return 0
    area = 0
    for i in range(1, length-1):
        temp = [[B[0][0], B[0][1]], [B[i][0], B[i][1]],
                [B[i+1][0], B[i+1][1]]]
        print(temp)
        area = area + triangle_area(temp)
    return area

B = [[-1, -2], [1, -2], [3, 2], [0,4], [-3,1]]
print(polygon_area(B))
```

9. Aufgabe 11 - Geldwechsel

```
#money_1.py
def change(z):
    B = [1,2,5,10,20,50,100,200]
    amount = [0] * len(B)
    temp = z
    for i in range(7,-1,-1):
        amount[i] = temp // B[i]
        temp = temp % B[i]
        print(B[i], " * ", amount[i], sep="", end="", flush=True)
        # print all amounts with exchange in the same line as an
        # equation
        if temp == 0:
        break
        print(" + ", sep="", end="", flush=True )
        # sep gives the separating string, end gives the string
        # on which the current string should end
    print(" = ", z)
print(change(1391))
```

Dieses Verfahren bricht für jede Zahl z ab, da der Basissatz eine eins beinhaltet. Durch die Annahme, dass die gegebene Zahl ein Integer ist und dem Vorwissen, dass jede ganze Zahl durch eins teilbar ist, ergibt sich die Tatsache, dass jede Zahl spätestens mit modulo 1 den Rest null ergibt.

```
#money_2.py
def change(z, B):
    #B = [1,2,5,10,20,50,100,200]
    temp = z
    current_amount = z+2
    for i in range(len(B)-1,-1,-1):
        temp = z
        amount = [0] * len(B)
        for j in range( i, -1, -1):
            amount[j] = temp // B[j]
            temp = temp % B[j]
                if (temp == 0) :
                    break
                    if (sum(amount)<current_amount) and (temp == 0):
                        current_amount = sum(amount)
                        #print("Amount of change is: ", current_amount)
```

```
                    print("Given base is: ", B[0:len(B)],
                    " to change: ", z)
                    if(current_amount < z+2):
                            print("Minimal amount of change is: ",
                            current_amount)
                    else:
                            print("Changing the given amount with
                    ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣current base rate is not possible")
base = [2]
print(change(215, base))
base2 = [1, 2, 5, 10, 20, 50, 100, 200]
print(change(215, base2))
base3 = [2, 5, 10, 11]
print(change(215, base3))
```

Die Funktion change(z) liest eine Zahl z ein, welche durch den Basissatz ausgedrückt werden soll. Diese Variable wird einem temporären Wert temp zugeordnet, damit sie nicht verändert wird. Diese temporäre Variable wird durch eine modulo-Rechnung durch den Basissatz dividiert. Damit man aus diesem Basissatz die minimale Anzahl aus Münzen erhält, wird von dem größten Wert im Basissatz zum kleinsten Wert gerechnet. Ist die Ausgabe der modulo-Rechnung zu irgendeinem Zeitpunkt null, wird die for-Schleife abgebrochen und das Ergebnis ausgegeben.

10. Aufgabe 12 - Test für Wortdoppelungen

```
def word_express():
    #counting words and the lines
    with open("Wortdopplung.txt") as wordrep:
    u = wordrep.read()
    print(u)
    #split the words
    v = u.split()
    print(v)
    print(len(v))
    n = len(v)
    for i in range(n-1):
        if (v[i] in v[i+1]) or (v[i+1] in v[i]):
            for num, line in enumerate(open('Wortdopplung.txt'), 0):
                if (v[i] in line) or (v[i+1] in line):
                print("In line: ", (num+1), ": ", v[i+1])

print(word_express())
```

11. Aufgabe 13 - Fehlerfortpflanzung

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

mu_1 = 40
sigma_1 = 0.05
M_1 = np.random.normal(mu_1, sigma_1, 1000000) # 10^4 kg
print(M_1)

mu_2 = 30
sigma_2 = 0.1
M_2 = np.random.normal(mu_2, sigma_2, 1000000) # 10^4 kg
print(M_2)

mu_r = 3.2
sigma_r = 0.011
r = np.random.normal(mu_r, sigma_r, 1000000) # m
print(r)

G = 6.67384 # 10^(-11) m(^3) kg^(-1) s^(-2)

Z = M_1*M_2*G
N = r*r
F = Z/N
print(F)

plt.hist(F/1000, 50, normed=1, facecolor='blue')
plt.xlabel('Force')
plt.ylabel('Frequency')
plt.show()

#b)

mu_1_b = 40
sigma_1_b = 8
M_1_b = np.random.normal(mu_1_b, sigma_1_b, 1000000) # 10^4 kg
print(M_1_b)

mu_2_b = 30
```

```
sigma_2_b = 6
M_2_b = np.random.normal(mu_2_b, sigma_2_b, 1000000) # 10^4 kg
print(M_2_b)

mu_r_b = 3.2
sigma_r_b = 0.6
r_b = np.random.normal(mu_r_b, sigma_r_b, 1000000) # m
print(r_b)

Z_b = M_1_b*M_2_b*G
N_b = r_b*r_b
F_b = Z_b/N_b
print(F_b)

plt.hist(F_b/1000, 50, normed=1, facecolor='red')
plt.xlabel('Force b')
plt.ylabel('Frequency b')
plt.xlim((0,3))
plt.show()
```

Obwohl die Gauß'sche Fehlerfortpflanzungsformel für kleine Abweichungen der physikalischen Gegebenheiten, wie in etwa kleine Abweichungen in den Massen und Abständen gut durch die Gauß'sche Glockenkurve annähert (siehe 11.1), ist diese Formel für große Schwankungen unzureichend genau (siehe 11.2), da diese die Fehler nicht mehr gut bzw. in einer wiedererkennbaren Gauß'schen Glockenkurve darstellen kann. Aus diesen Graphen kann man feststellen, dass die Gauss'sche Fehlerfortpflanzungsformel nur für hinreichend kleine Werteschwankungen anwendbar ist.

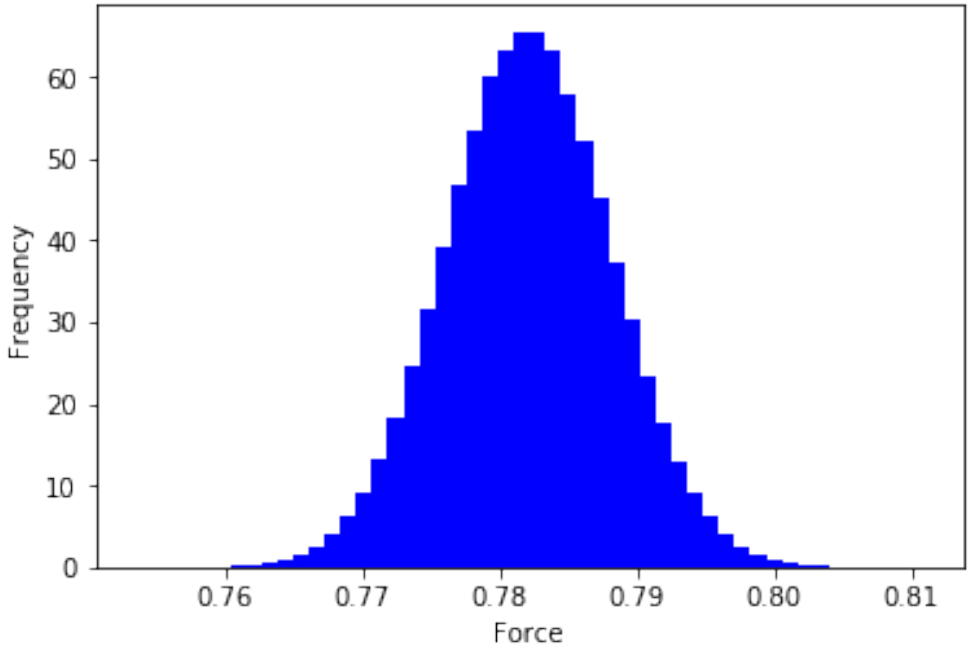

Abbildung 11.1.: Ein Gauß'sche Fehlerfortpflanzungsgraph

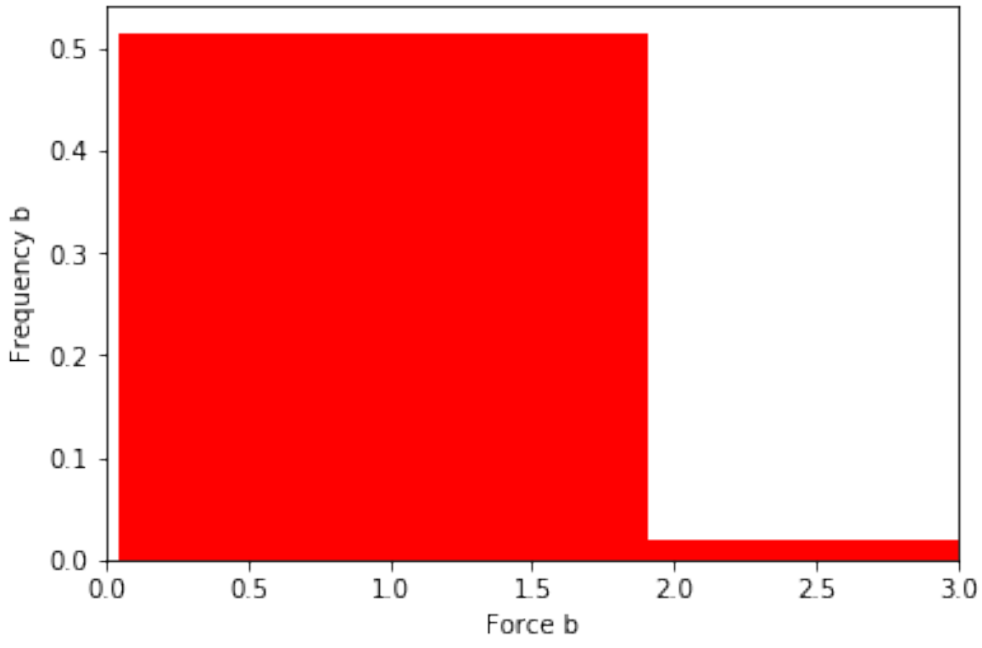

Abbildung 11.2.: Gauß'scher Fehlerfortpflanzungsgraph für große Abweichungen

12. Aufgabe 14 - Felix Baumgartners Fallschirmsprung

```
import scipy as sp
import numpy as np

def air_density(z):
    return 1.22*np.exp(-z/10)

def cross_section(t):
    e = np.exp(-(t-t_f)/5)
    return 1+50*((1-e)/(1+e))

def friction(position, mass, velocity, t):
    gravity = 9.8
    if position>2:
        a_t = 0.9
        c_f = 0.3
    else:
        c_f = 1.5
        a_t = cross_section(t)
    mass_acc = -mass*gravity+(1/2)*air_density(position)*c_f*a_t*
    velocity*velocity
```

Dieser Teil der Aufgabe zu Felix Baumgartners Fallschirmsprung ist die Basis durch welcher die Differentialgleichung gelöst werden kann und die Aufgabe somit implementiert werden kann.

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[1] John Ronald Reuel Tolkien. *Der Hobbit oder Hin und Zurück*. Übers. von Wolfgang Krege. 1997. ISBN: 978-3-608-93818-0.

[2] John Ronald Reuel Tolkien. *Kleiner Hobbit und der große Zauberer (später: Der kleine Hobbit)*. Übers. von Walter Scherf. 1957.

[3] John Ronald Reuel Tolkien. *The Hobbit or There and Back Again*. 1937.