

## Vorbemerkung

Dies ist ein abgegebener Übungszettel aus dem Modul math241.

Dieser Übungszettel wurde nicht korrigiert. Es handelt sich lediglich um meine Abgabe und keine Musterlösung.

Alle Übungszettel zu diesem Modul können auf [http://martin-ueding.de/de/university/bsc\\_physics/math241/](http://martin-ueding.de/de/university/bsc_physics/math241/) gefunden werden.

Sofern im Dokument nichts anderes angegeben ist: Dieses Werk von Martin Ueding ist lizenziert unter einer [Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#).

[disclaimer]

# math240 Übung 9

## Tutorin: Inka Hammer

Martin Ueding  
mu@uni-bonn.de

13. Juni 2012

|         |     |     |     |     |     |          |
|---------|-----|-----|-----|-----|-----|----------|
| Aufgabe | 1   | 2   | 3   | 4   | 5   | $\Sigma$ |
| Punkte  | / 4 | / 4 | / 4 | / 4 | / 4 | / 16     |

### Aufgabe 1 Cantor-Menge

$T_\epsilon(I)$  ist letztlich das (maximal abgeschlossene) Intervall  $I$  nur ohne einen  $\epsilon$ -Ball in der Mitte. Somit ist es die Vereinigung von zwei kompakten Intervallen.

Die Vereinigung

$$\bigcup_{I \in M_{n-1}} T_{2^{-2n-1}}(I)$$

ist wieder ein fraktales Gebilde. Aus allen Teilintervallen wird nun wieder ein Stück herausgeschnitten, allerdings ein immer kleineres Stück.

Zur Veranschaulichung des Problems habe ich ein kleines Programm in Python geschrieben. Dies ist ab Seite 4 mit Ausgabe abgedruckt.

Die Menge  $\mathcal{C}$  ist der Durchschnitt aller  $(M_n)_{n \in \mathbb{N}}$ , also in diesem Fall das letzte  $M_n$  in der Folge. Somit brauche ich nur  $\lim_{n \rightarrow \infty} M_n$  für  $\mathcal{C}$  zu betrachten.

Eine Menge ist genau dann Jordan-messbar, wenn ihre charakteristische Funktion  $\chi$  integrierbar ist. Und dass ist sie genau dann, wenn die Menge ihrer Unstetigkeitsstellen eine Nullmenge ist. Die Unstetigkeiten von  $\chi$  treten an den Randpunkten der  $I \in M$  auf.

Es ist leicht zu sehen, dass die Intervalle  $I$  in jedem Iterationsschritt mehr oder weniger halbiert werden. Das  $\epsilon$ , das in der Mitte herausgeschnitten wird, wird auch in jedem Schritt halbiert. Von daher wird kein Intervall komplett weggeschnitten. Ich kann also sagen, dass die Intervalle sich niemals überlappen und sich deren Anzahl in jedem Schritt verdoppelt.

Die Menge der Randpunkte ist für  $n = 0$  gegeben durch  $R_0 = \{0, 1\}$ . Diese Menge ist eine Nullmenge. In jedem Iterationsschritt verdoppelt sich die Menge der Randpunkte, es gilt also:

$$|R_n| = 2 |R_{n-1}|$$

Eine endliche Kardinalität verdoppelt ist immer noch endlich. Nach dem Prinzip der vollständigen Induktion ist also jedes  $n$  gegeben, dass  $|R_n| \leq \aleph_0$  gilt, also  $R_n$  eine Nullmenge ist. Eine endliche Vereinigung von Nullmengen ist immer noch eine Nullmenge. Da die  $M$  gerade durch  $n$  abgezählt werden, ist auch der Rand von  $\mathcal{C}$  eine Nullmenge. Somit ist  $\mathcal{C}$  Jordan-messbar.

## Aufgabe 2 Transformation zu Polarkoordinaten

Es ist zu zeigen, dass gilt:

$$\int_{\mathbb{R}^2} f(x, y) \, d(x, y) = \int_0^{2\pi} \int_0^\infty r f(r \cos(\phi), r \sin(\phi)) \, dr \, d\phi$$

Das erste Integral kann ich aufspalten in:

$$\int_{-\infty}^\infty \int_{-\infty}^\infty f(x, y) \, dx \, dy \quad (1)$$

Die Transformation von Polarkoordinaten zu kartesischen Koordinaten lautet:

$$\Phi(r, \phi) = r \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}$$

Die Jakobimatrix ist:

$$J = \begin{pmatrix} \frac{\partial \Phi_x}{\partial \phi} & \frac{\partial \Phi_x}{\partial r} \\ \frac{\partial \Phi_y}{\partial \phi} & \frac{\partial \Phi_y}{\partial r} \end{pmatrix}$$

Die Determinante ist einfach:

$$|J| = r \sin(\phi)^2 + r \cos(\phi)^2 = r$$

Mit der Transformationsformel, in der die Determinante als Multiplikator auftaucht, kann ich (1) umschreiben zu:

$$\int_0^{2\pi} \int_0^\infty \underbrace{r}_{|J|} f(\Phi(r, \phi)) \, dr \, d\phi$$

Nun setze ich die Transformation explizit ein:

$$\int_0^{2\pi} \int_0^\infty r f(r \cos(\phi), r \sin(\phi)) \, dr \, d\phi$$

Dies ist das Integral, das gezeigt werden sollte.

## Aufgabe 3 uneigentliche Integrale

### Aufgabe 3.a Existenz

Da momentan nur die Existenz zu zeigen ist, benutze ich das Lebesguesche Integrabilitätskriterium (Satz 3.6). Es bleibt also zu zeigen, dass die Integranden beschränkt sind und dass die Menge der Unstetigkeitsstellen eine Nullmenge ist.

Die Exponentialfunktionen sind komplett stetig, die Exponenten auch, da es Polynome sind. Somit sind die Integranden komplett stetig.

Beschränkt sind sie auch, sie haben das Maximum in Koordinatenursprung und als Infimum 0.

Somit existieren beide Integrale.

**Aufgabe 3.b Gleichheit**

Es ist zu zeigen, dass gilt:

$$\int_{\mathbb{R}^2} \exp(-x^2 - y^2) \, d(x, y) = \left( \int_{\mathbb{R}} \exp(-x^2) \, dx \right)^2$$

Ich schreibe das implizierte Doppelintegral in kartesischen Koordinaten.

$$\int_{\mathbb{R}^2} \exp(-x^2 - y^2) \, d(x, y) = \int_{\mathbb{R}} \int_{\mathbb{R}} \exp(-x^2 - y^2) \, dx \, dy$$

Die Exponentialfunktion ziehe ich auseinander.

$$= \int_{\mathbb{R}} \int_{\mathbb{R}} \exp(-x^2) \exp(-y^2) \, dx \, dy$$

Die Integrale sind komplett voneinander unabhängig, ich ziehe sie auseinander.

$$= \int_{\mathbb{R}} \exp(-x^2) \, dx \int_{\mathbb{R}} \exp(-y^2) \, dy$$

Jetzt kann ich noch  $y$  durch  $x$  umbenennen und ich bin fertig.

$$= \left( \int_{\mathbb{R}} \exp(-x^2) \, dx \right)^2$$

**Aufgabe 4 Integral über Glockenkurve**

Es soll folgendes Integral berechnet werden:

$$\int_{-\infty}^{\infty} \exp(-x^2) \, dx$$

$$\int_{-\infty}^{\infty} \exp(-x^2) \, dx = \sqrt{\left( \int_{-\infty}^{\infty} \exp(-x^2) \, dx \right)^2}$$

Wie eben in der Aufgabe hergeleitet, kann ich dieses Integral auch mit zwei verschiedenen Variablen ausführen.

$$= \sqrt{\left( \int_{-\infty}^{\infty} \exp(-x^2) \, dx \right) \left( \int_{-\infty}^{\infty} \exp(-y^2) \, dy \right)}$$

Die Integrale „mische“ ich nun.

$$\begin{aligned} &= \sqrt{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-x^2) \exp(-y^2) \, dy \, dx} \\ &= \sqrt{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-x^2 - y^2) \, dy \, dx} \end{aligned}$$

Nun transformiere ich zu Polarkoordinaten.

$$= \sqrt{\int_0^{2\pi} \int_0^\infty \exp(-r^2) r \, dr \, d\phi}$$

Das Integral über  $\phi$  ist trivial.

$$= \sqrt{2\pi \int_0^\infty \exp(-r^2) r \, dr}$$

Jetzt kann ich substituieren, dazu nehme ich  $z = r^2$ ,  $dz = 2r \, dr$ . Somit ist das Integral:

$$= \sqrt{\pi \int \exp(-z) \, dz}$$

Das Integral ist recht einfach:

$$= \sqrt{\pi [-\exp(-r^2)]_0^\infty}$$

Das Einsetzen der Grenzen ergibt schlicht 1. Somit folgt für das Integral:

$$= \sqrt{\pi}$$

## Aufgabe 5 offene Kugel

Diese Aufgabe lasse ich aus.

## Programm zur Cantormenge

Ich schneide von jedem kompakten Teilintervall  $I \in M$  genau den  $\epsilon$ -Ball in der Mitte heraus. Das Programm gibt die Intervalle dann aus. Es entstehen in jedem Schritt mehr und mehr Teilintervalle, die auch immer kleiner werden. Für die ersten paar Schritte sind die Teilmengen abgedruckt, danach nur noch der Inhalt. Über 19 Schritte hinaus ist auf meinem Computer mit dem naiv implementierten Algorithmus nicht sinnvoll machbar.

Das Programm errechnet für  $|M_{19}| = 0.750000476837$ . Ich gehe davon aus, dass im Grenzfall  $n \rightarrow \infty$  einfach  $\frac{3}{4}$  herauskommt.

### Python Programm

Dieses Programm kann auch unter <http://uni-bonn.de/~s6mauedi/math240/> heruntergeladen werden.

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3
4 # Copyright © 2012 Martin Ueding <dev@martin-ueding.de>
5
6 # This program is free software: you can redistribute it and/or modify it

```

```

7 # under the terms of the GNU General Public License as published by the Free
8 # Software Foundation, either version 2 of the License, or (at your option)
9 # any later version.
10 #
11 # This program is distributed in the hope that it will be useful, but WITHOUT
12 # ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
13 # FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for
14 # more details.
15 #
16 # You should have received a copy of the GNU General Public License along
17 # with this program. If not, see <http://www.gnu.org/licenses/>.
18
19 # Dieses Programm iteriert die Cantormenge aus math240, Übungszettel 9,
20 # Aufgabe 1.
21
22 import pprint
23
24 def main():
25     printer = pprint.PrettyPrinter(indent=4)
26
27     # Maximalwert für die Iterationen:  $n_{\max} := 19$ 
28     nmax = 19
29
30     # Die gegebene Anfangsmenge  $M_0 = \{[0,1]\}$ . Das
31     # ganze wird hier als Tupel in einer Liste dargestellt. Die Menge  $M$ 
32     # (hier als Liste implementiert) enthält dann die Intervalle  $I$  (hier als
33     # Tupel implementiert).
34     M = [(0.0, 1.0)]
35
36     # Gehe nun die  $n = 1, \dots, n_{\max}$  durch.
37     for n in range(1, nmax + 1):
38         # Wende die Iterationsvorschrift für  $M_n$  mit  $M_{n-1}$  an.
39         M = schritt(M, n)
40
41         # Gebe  $n$  und  $|M_n|$  aus.
42         print "n: {n:2d}, Inhalt {inhalt:.10f}".format(n=n, inhalt=inhalt(M))
43
44         # Gebe für  $n \leq 4$  die  $I \in M_n$  aus.
45         if n <= 4:
46             printer.pprint(M)
47
48     # Wende die Formel  $T_n(I)$  für jedes  $I \in M$  an.
49     #  $M := \bigcup_{I \in M} T_n(I)$ 
50     def schritt(M, n):
51         return [item for sublist in (T(n, I) for I in M) for item in sublist]
52
53     # Berechne ein  $\epsilon$  aus dem  $n$  und schneide dieses aus  $I$  heraus.
54     def T(n, I):
55         a = I[0]
56         b = I[1]
57
58         #  $\epsilon := 2^{2-n-1}$ 
59         epsilon = 2.0**(-2.0*n - 1)

```

```

60
61     #  $[a, \frac{a+b}{2} - \frac{\epsilon}{2}] \cup [\frac{a+b}{2} + \frac{\epsilon}{2}, b]$ 
62     return [(a, (a+b)/2-epsilon/2), ((a+b)/2+epsilon/2, b)]
63
64 # Berechne den Inhalt von I, indem der Durchmesser der einzelnen Teile
65 # addiert wird.
66 def inhalt(M):
67     # Sei I gegeben durch [a,b]. Dann ist die Summe:
68     #  $\sum_{I \in M} |I| = \sum_{I \in M} (b - a)$ 
69     return sum([I[1]-I[0] for I in M])
70
71 if __name__ == '__main__':
72     main()

```

### Ausgabe des Programs

```

n: 1, Inhalt 0.8750000000
[(0.0, 0.4375), (0.5625, 1.0)]
n: 2, Inhalt 0.8125000000
[(0.0, 0.203125), (0.234375, 0.4375), (0.5625, 0.765625), (0.796875, 1.0)]
n: 3, Inhalt 0.7812500000
[ (0.0, 0.09765625),
  (0.10546875, 0.203125),
  (0.234375, 0.33203125),
  (0.33984375, 0.4375),
  (0.5625, 0.66015625),
  (0.66796875, 0.765625),
  (0.796875, 0.89453125),
  (0.90234375, 1.0)]
n: 4, Inhalt 0.7656250000
[ (0.0, 0.0478515625),
  (0.0498046875, 0.09765625),
  (0.10546875, 0.1533203125),
  (0.1552734375, 0.203125),
  (0.234375, 0.2822265625),
  (0.2841796875, 0.33203125),
  (0.33984375, 0.3876953125),
  (0.3896484375, 0.4375),
  (0.5625, 0.6103515625),
  (0.6123046875, 0.66015625),
  (0.66796875, 0.7158203125),
  (0.7177734375, 0.765625),
  (0.796875, 0.8447265625),
  (0.8466796875, 0.89453125),
  (0.90234375, 0.9501953125),
  (0.9521484375, 1.0)]
n: 5, Inhalt 0.7578125000
n: 6, Inhalt 0.7539062500
n: 7, Inhalt 0.7519531250
n: 8, Inhalt 0.7509765625
n: 9, Inhalt 0.7504882812
n: 10, Inhalt 0.7502441406
n: 11, Inhalt 0.7501220703

```

n: 12, Inhalt 0.7500610352  
n: 13, Inhalt 0.7500305176  
n: 14, Inhalt 0.7500152588  
n: 15, Inhalt 0.7500076294  
n: 16, Inhalt 0.7500038147  
n: 17, Inhalt 0.7500019073  
n: 18, Inhalt 0.7500009537  
n: 19, Inhalt 0.7500004768