SAGE-Tutorium 10 im SoSe 2009

Lars Fischer*

01.07.2009

Inhaltsverzeichnis

*WWW: `http://w3.countnumber.de/fischer`, EMail: vorname.nachname (bei der) uni-siegen.de

1 Wiederholung

- Projektive Geometrie
- Parametrisierung von Kegelschnitten

2 Projektive Geometrie

In diesem Abschnitt erkläre ich einige Dinge an, die ich letzte Woche ad-hoc erklärt habe.

2.1 In Bildern

(Wenn die Plots im Browser nicht funktionieren, dann per Copy&Paste in ein Terminal mit einer SAGE-Session einfügen. Komischerweise startet das Terminal Jmol ohne Probleme.)

Zuerst einmal die Visualisierung der Projektiven Ebene als Ebene im dreidimensionalen euklidischen Raum:

```
Origin = point([0,0,0], color="green", size=10 )+ text3d("[x,y,0]", (0,0,-0.5) )
m=5

t= [[-1,1,1],[1,1,1],[1,-1,1],[-1,-1,1],[-1,-1,1]]
v= [ [m*x,m*y,1] for x,y,z in t]

polygon3d.options["opacity"]=0.6
v.append([m,m,1])
ProjPlane = polygon3d(v, color=(0,0,1)) + text3d("[x,y,1]", (0,0,1.5) )
polygon3d.options["opacity"]=1
P=Origin+ProjPlane
P.show(aspect_ratio= True )



# Einer euklidischen Geraden entspricht ein projektiver Punkt:
t=(m-1,m-1,1)
ProjPoint=point(t, color="green", size=10 )

P1= line3d([(0,0,0),t])
P=Origin+ProjPlane+ProjPoint+P1
P.show(aspect_ratio=True)
```

```
# Die euklidische Ebene, die ProjPlane in einer Geraden schneidet und
# somit eine projektive Gerade ist:
L1= polygon3d( [[0,-m,0], [0,-m,1.1], [0,m,1.1],[0,m,0]], color="red")
# Projektive Gerade, die durch die Ebene dargestellt wird:
G1= line3d([[0,-m,1], [0,m,1]], color="green", thickness=3)
P=Origin+ProjPlane+L1+G1
P.show(aspect_ratio= True )


# Der Schnitt zweier euklidischer Ebenen (= zweier proj. Geraden) ist
# eine euklidische Gerade (= ein projektiver Punkt):
L2= polygon3d( [[m-1,-m,0], [m-1,-m,1.1], [-m+1,m,1.1],[-m+1,m,0] ],
               color="yellow")
G2= line3d([[m-1,-m,1], [-m+1,m,1]], color="green", thickness=3)

P2=point([0,0,1], color="green", size=10)
P=Origin+ProjPlane+L1+G1+L2+G2+P2
P.show(aspect_ratio= True)



# Jetzt werden zwei Geraden geschnitten, die in der proj. Ebenen parallel
# sind. Das sind euklidische Ebenen, die eine Schnittgerade im Ursprung
# haben. Das ist der projektive "Undendlich Ferne Punkt" (also eine
# euklidische Gerade), den parallele proj. Geraden gemeinsam haben.
L3=polygon3d( [[0,-m,0], [m-1,-m,1.1], [m-1,m,1.1],[0,m,0]], color="red")
P3=line3d([[0,-m,0], [0,m,0]], color="green", thickness=3) # uneigentlicher Fernpunkt
P=Origin+ProjPlane+L1+L3+P3
P.show(aspect_ratio= True)



# Die (euklidische) Richtung der Fernpunkte (als euklidische Gerade
# aufgefasst) hängt von der "Richtung" der beiden parallelen
# projektiven Geraden in der projektiven Ebene ab.
# Unterschiedliche Paare paralleler projektiver Geraden schneiden sich in
# unterschiedlichen Fernpunkten (= unterschiedliche euklidische Geraden).
L4= polygon3d( [[m-1,-m,0], [m-1+1,-m,1.1], [-m+1+1,m,1.1],[-m+1,m,0] ], color="yello
P4=line3d([[m-1,-m,0], [-m+1,m,0]], color="green", thickness=3) # uneigentlicher Fern
P=Origin+ProjPlane+L1+L3+L2+L4+P3+P4
P.show(aspect_ratio= True)
```

2.2 In Worten

- Jeder Punkt der Ebene $E := \{[x, y, z] | z = 1\}$ entspricht einem Punkt der Projektiven Ebene (aber es gibt noch mehr Punkte).
- Jede Gerade in der Ebene E (also jede Ebene die E schneidet und durch den Ursprung geht) hat unendliche viele Geraden, die parallel zu ihr verlaufen. Zu jeder solchen Richtung m existiert ein **Unendlich Ferner Punkt** P_m (das sind die grünen Geraden in der Ebene $z = 0$ aus den obigen Plots). Per Definitionem sagen wir, dass alle parallelen Geraden mit einer Richtung m auch den Punkt P_m enthalten sollen (wir fügen P_m zu jeder Geraden mit Richtung m hinzu). Nun schneiden sich die Parallelen in diesem Punkt.
- Alle Unendliche Fernen Punkte (die Geraden in der Ebene $z = 0$ bilden die **Unendlich Ferne Gerade** (das ist die Ebene $z = 0$).

2.3 In Formeln

2.3.1 Zwei Punkte ...

Zwei verschiedene Punkte der Projektiven Ebene bestimmen eine Gerade:

Und zwar ist die projektive Gerade, die durch $[p_1, p_2, p_3]$ und $[q_1, q_2, q_3]$ bestimmt wird, eine euklidische Ebene. Ist $[x, y, z]$ ein dritter Punkt der Ebene, so sind die drei Vektoren linear abhängig. Das führt auf die Gleichung.

$$\det\left(\begin{bmatrix} x & y & z \\ p_1 & p_2 & p_3 \\ q_1 & q_2 & q_3 \end{bmatrix}\right) = 0.$$

Daraus erhalten wir die die Gleichung der Ebene in Normalenform:

$$(p_2q_3 - p_3q_2)x + (p_3q_1 - p_1q_3)y + (p_1q_2 - p_2q_1)z = 0$$

(Das ist die projektive Gerade, die die beiden verschiedenen Punkte enthält.)

2.3.2 Zwei Geraden ...

Eine projektive Gerade ist durch ihren Normalenvektor gegeben. Da Skalierungen desselben nichts an seiner Eigenschaft ändern senkrecht zu sein, können wir auch den Normalenvektor (und damit die euklidische Ebenen, bzw. die projektive Gerade) in homogenen Koordinaten schreiben $[n_1, n_2, n_3]$.

Es seien $\vec{n} := [n_1, n_2, n_3]$ und $\vec{n}' := [n_1', n_2', n_3']$ verschiedene Normalenvektoren. Ein Vektor $[x, y, z]$ mit

$$\det\left(\begin{bmatrix} x & y & z \\ n_1 & n_2 & n_3 \\ n_1' & n_2' & n_3' \end{bmatrix}\right) = 0$$

ist aus der linearen Hülle der beiden Normalenvektoren. Das beschreibt die Ebene, die von den beiden Normalenvektoren aufgespannt wird. Allerdings in Normalenform. Das bedeutet an der Determinante lesen wir einen Vektor ab, der senkrecht zu $\vec{n}$ und $\vec{n}'$ ist. Das ist die euklidischen Schnittgerade, bzw. die homogenen Koordinaten des projektiven Schnittpunktes.

2.3.3 Dualität

Vornehm kann man sich so ausdrücken: Zwei verschiedene Punkte **inzidieren** zu einer Geraden. Zwei verschiedene Geraden **inzidieren** zu einem Punkt.

Die Determinantenformeln oben funktionieren immer, unabhängig davon, ob die Punkte eigentlich oder uneigentlich sind (also unabhängig davon, ob es Punkte in der Ebene $z = 1$ oder Linien in der Ebene $z = 0$ sind). Unabhängig ob die beiden Geraden parallel sind oder nicht (im ersten Fall kommt eine Gerade in der Ebene $z = 0$ im zweiten Fall ein Punkt in der Ebene $z = 1$ raus.

Praktischerweise inzidieren zwei verschiedene Objekte mit homogenen Koordinaten zu einem Objekt mit homogenen Koordinaten. Und die Formeln lassen sich nicht unterscheiden.

Liegt ein Punkt $[p_1, p_2, p_3]$ auf einer Geraden $[n_1, n_2, n_3]$, so gilt

$$\sum_{i=1}^{3} p_i n_i = 0 \quad \text{oder als Skalarprodukt} \quad \langle \vec{p} | \vec{n} \rangle = 0.$$

Diese Formel ist symmetrisch, $\vec{n}$ und $\vec{p}$ sind völlig gleichberechtigt. Hier kommt es ebenfalls nicht auf eigentliche oder uneigentliche Punkte oder Geraden an.

3 Kettenbrüche

(Eine Zusammenfassung des Kapitels über Kettenbrüche in Stein: **Elementary Number Theory: Primes, Congruences, and Secrets**)

Zum Anfang eine paar einfache Beobachtungen:

$$[a_0] = a_0$$
$$[a_0, a_1] = a_0 + \frac{1}{a_1} = \frac{a_0 a_1 + 1}{a_1}$$

$$[a_0, a_1, a_2] = a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = \frac{a_0 a_1 a_2 + a_0 + a_2}{a_1 a_2 + 1}$$
$$\begin{aligned}[a_0, a_1, \ldots, a_{n-1}, a_n] &= \left[a_0, a_1, \ldots, a_{n-2}, a_{n-1} + \frac{1}{a_n}\right] \\ &= a_0 + \frac{1}{[a_1, \ldots, a_{n-1}, a_n]} \\ &= [a_0, [a_1, \ldots, a_{n-1}, a_n]]\end{aligned}$$

Wir können mit der SAGE Funktion `continued_fraction` Kettenbrüche erzeugen. Der Parameter `bits` steuert die Genauigkeit. Ebenso lassen mit Kettenbrüchen Berechnungen ausführen.

```
a=continued_fraction(9/13)
b=continued_fraction(4/13)

print a, b,  a+b
print a.value(), b.value()

for prec in [10,20,30,40,50]:
    print "pi mit %d:"%prec, continued_fraction( pi, prec)
```

3.1 Näherungsbrüche (engl. Partial Convergents)

Es sei $[a_0, \ldots, a_m]$ ein Kettenbruch (gerne auch mit unendlich vielen a_i). Wir wollen für $n \leq m$ die Näherung $[a_0, \ldots, a_n]$ an $[a_0, \ldots, a_m]$ als Bruch $\frac{p_n}{q_n}$ angeben. Dafür gibt es Rekursionsformeln:

Wir definieren für $-2 \leq n \leq m$ die Zahlen p_n, q_n als:

$$\begin{array}{llll} p_{-2} := 0, & p_{-1} := 1, & p_0 := a_0, & \ldots \quad p_n := a_n p_{n-1} + p_{n-2}, \ldots \\ q_{-2} := 1, & q_{-1} := 0, & q_0 := \ \ 1, & \ldots \quad q_n := a_n q_{n-1} + q_{n-2}, \ldots \end{array}$$

Aufgabe 1: Implementiert eine Funktion, die zu einer Liste $[a_0, \ldots, a_m]$ von Zahlen die Liste der Näherungsbrüche $\left[\frac{p_0}{q_0}, \ldots \frac{p_m}{q_m}\right]$ berechnet.

Testet Eure Funktion an `c= continued_fraction(pi, 33)`. Vergleicht mit der Ausgabe von `c.convergents()`, sowie `c.pn(n)/c.qn(n)`

Mit den Methoden `pn` und `qn` eines Kettenbruchsobjektes können wir die Formeln aus der Vorlesung nachvollziehen, z.B.:

```
c= continued_fraction(pi , 50)
for n in range(-1,len(c) ):
    print c.pn(n)*c.qn(n-1) - c.qn(n) * c.pn(n-1),
```

```
for n in range(len(c) ):
    print c.pn(n)*c.qn(n-2) - c.qn(n) * c.pn(n-2),
```

Sei nun $cn := [a_0, \ldots, a_n] = \frac{p_n}{q_n}$ die Folge der Näherungsbrüche. Nach einem Satz der Vorlesung ist die Folge der c_n mit geradem n wachsend und die Folge mit ungeradem n ist fallend:

```
c= continued_fraction(e , 50)
n=0
while n+1 < len(c):
    print "%10f, %#10g"%((c.convergent(n)),(c.convergent(n+1)))
    n+=2
```

Das kann man auch grafisch darstellen:

```
c= continued_fraction( golden_ratio, 10)
v = [(i, c.pn(i)/ c.qn(i)) for i in range(len(c)) ]
P= points(v, rgbcolor="red" )
L= line(v, rgbcolor="black")
(L+P).show()
```

3.2 Kettenbruchdarstellung von rationalen Zahlen

In der Vorlesung habt Ihr gesehen, wie sich jede rationale Zahl a/b mit $\mathrm{ggT}(a,b) = 1$ als endlicher Kettenbruch darstellen lässt. Das Verfahren leitet sich vom euklidischen Algorithmus ab.

$$\begin{aligned} a &= ba_0 + r_1 && 0 \leq r_1 < b \\ b &= r_1 a_1 + r_2 && 0 \leq r_2 < r_1 \\ &\dots \\ r_{n-2} &= r_{n-1}a_{n-1} + r_n && 0 \leq r_n < r_{n-1} \\ r_{n-1} &= r_n a_n + 0 \end{aligned}$$

Wir dividieren jede Zeile durch den Faktor links von a_i und erhalten folgendes Verfahren:

$$\begin{aligned} a/b &= a_0 + r_1/b = a_0 + \frac{1}{\frac{b}{r_1}} \\ b/r_1 &= a_1 + r_2/r_1 a_1 + \frac{1}{\frac{r_1}{r_2}} \\ &\dots \\ r_{n-1}/r_n &= a_n \end{aligned}$$

Am Ende erhalten wir

$$\frac{a}{b} = [a_0, \dots a_n]\,.$$

Aufgabe 2: Implementiert eine SAGE-Funktion, die zu einem vollständig gekürzten Bruch die Darstellung als Kettenbruch berechnet.

3.3 Beispiele unendlicher Kettenbrüche

Es sei $\phi := \frac{1+\sqrt{5}}{2}$ der goldene Schnitt. Wir ermitteln den Kettenbruch zu ϕ.

Zuerst einmal ist $\phi = 1 + \frac{-1+\sqrt{5}}{2}$. Es sei $a_0 := 1$ und $r_0 := \frac{-1+\sqrt{5}}{2}$. Wir berechnen $\frac{1}{r_0}$:

$$\begin{aligned} \frac{2}{-1+\sqrt{5}} &= \frac{2}{-1+\sqrt{5}}\frac{1+\sqrt{5}}{1+\sqrt{5}} = \frac{2+2\sqrt{5}}{4} \\ &= \frac{1+\sqrt{5}}{2} = \phi \end{aligned}$$

Das führt zu $a_1 := 1 = a_0, r_1 := \frac{-1+\sqrt{5}}{2} = r_0$ und allgemein zu $a_i := 1$. Damit ist $\phi = [1,1,1,\dots]$.

Das kann uns SAGE ebenfalls bestätigen:

```
for b in [20,30,40,50,60,100]:
    print continued_fraction( golden_ratio , b)
```

Oben sieht man einen allgemeinen (d.h. $x \in \mathbb{R}$) Algorithmus zur Bestimmung der a_i:

Es ist $a_0 := floor(x)$, $t_0 := x - a_0$. Falls $t_0 \neq 0$ ist, berechne $\frac{1}{t_0} = a_1 + t_1$ mit $a_1 \in \mathbb{N}$ und $0 \leq t_1 < 1$ (wieder über floor). Dann haben wir $t_0 = \frac{1}{a_1 + t_1}$ berechnet.
Solange $t_n \neq 0$ fahre fort mit $\frac{1}{t_n} = a_{n+1} + t_{n+1}$.

Aufgabe 3: Implementiert dieses Verfahren in SAGE. Ihr könnt es rekursiv oder mit einer Schleife machen.

4 Fragen zur Vorlesung?

5 Fragen zu den Projekten?

Denkt daran, dass es nur noch noch drei Wochen bis zur Präsentation am 22.07.2009 sind!

Zur Ausstellung der Scheine benötigen wir die folgenden Informationen:

- Name
- Matrikelnummer
- Studiengang (genaue Bezeichnung)
- Geburtstag
- Anschrift

Sendet mir diese Daten bitte per EMail zu.

6 Nächstes Mal

- Habt Ihr noch ein besonderes Interesse an einem Themengebiet?
- Reguläre Ausrücke?

7 Quellcode

Das gesamte Worksheet ist als Text-Datei in dem PDF eingebettet.

- Im Acrobat-Reader lässt es sich unter dem Büroklammer-Symbol in der linken Leiste herunterladen.
- Okular zeigt es im File-Menu als Embeded Files an.
- Unter Linux kann man die Text-Datei auch mit
  pdftk Tutorium10.pdf unpack_files
  aus dem PDF herauslösen.

Anschließend lässt sich die Text-Datei mit der Upload-Funktion des SAGE-Notebooks hochladen.