Dynamische Aspekte von Parameterdarstellungen: Generieren von Bewegungsbahnen sowie von Geraden und Kurven als Punktmengen

Andreas Filler, Heidelberg

Die Einbeziehung von Computervisualisierungen und die Erstellung einfacher Animationen durch die Schüler können dazu beitragen, bei der Behandlung von Parameterdarstellungen von Geraden und anderen geometrischen Objekten oft vernachlässigte Gesichtspunkte – insbesondere den Punktmengengedanken, funktionale Zusammenhänge sowie dynamische Aspekte – einzubeziehen und „mit Leben zu erfüllen“. In diesem Beitrag werden hierfür anhand von Geraden und Ebenen sowie als Bahnkurven aufgefassten Kreisen, Spiralen, Schraubenlinien und Wurfparabeln Vorschläge unterbreitet und Vorgehensweisen unter Verwendung der 3D-Grafiksoftware POV-Ray sowie (alternativ dazu) des CAS MuPAD skizziert. Die Einbeziehung elementarer Arbeitsweisen der Informatik (Nutzung von Schleifen und Prozeduren) kann u. a. dazu dienen, mithilfe einfacher Verfahren, die sehr oft ausgeführt werden, komplexe Objekte zu erzeugen – dies wird anhand von Darstellungen geometrischer Objekte als Punktmengen verdeutlicht.

Für alle in diesem Beitrag beschriebenen Inhalte stehen Beispieldateien, Videos und ergänzende Materialien auf der Internetseite **`www.afiller.de/3dcg`** (unter der Rubrik Downloads & Links) zur Verfügung.

1 Einleitung, Problemlage

Parameterdarstellungen von Geraden und Ebenen gehören zu den Standardinhalten des Unterrichts in analytischer Geometrie. Meist folgen dabei nach einer Einführung der Parameterdarstellungen sehr schnell Aufgaben zur Umformung von Parameter- in Koordinatenform und umgekehrt sowie zur Untersuchung von Lagebeziehungen, der Bestimmung von Schnittpunkten sowie (meist etwas später) zu Abstands- und Winkelberechnungen. Zwei wichtige, miteinander verbundene, Aspekte der analytischen Geometrie, die anhand der Parameterdarstellungen gut verfolgt werden könnten, kommen dabei nicht in ausreichendem Maße zur Geltung:[1]

- Die Schüler gelangen höchstens in Ansätzen zu einer *Auffassung geometrischer Objekte als Punktmengen*.[2]
- Der *funktionale Zusammenhang zwischen dem Parameter* (bzw. den Parametern) *und den zugehörigen Punkten* wird von den Schülern meist nicht erkannt. Das Erkennen dieses Zusammenhangs setzt eine Sicht auf geometrische Objekte als Punktmengen natürlich voraus, geht aber darüber noch insofern hinaus, als die Abhängigkeit der Lage von Punkten im Raum von dem Parameter bzw. den Parametern zu erfassen ist.[3]

Als didaktische Ansätze, die Herausbildung auf den konkret-gegenständlichen Aspekt eingeengter Konzepte von Parameterdarstellungen bei Schülern zu vermeiden sowie den Punktmengengedanken und den Schwerpunkt des funktionalen Zusammenhangs stärker einzubeziehen, bieten sich vor allem zwei Herangehensweisen an (vgl. auch [4]):

- Die Schüler *konstruieren die zu einigen Parameterwerten gehörenden Punkte* bei einer Parameterdarstellung der Form $P = P_0 + t \cdot \vec{a}$ und erkennen dabei, dass diese Punkte auf einer Geraden liegen.

[1] Als weiteres Defizit ist die vielfach beklagte Einengung auf die Betrachtung von linearen Objekten (Geraden und Ebenen) zu nennen.

[2] G. Wittmann untersuchte auf Parametergleichungen von Geraden bezogene Schülerkonzepte und stellte fest, dass Schüler diese oft nicht als Gleichungen ansahen, die Mengen von Punkten in Abhängigkeit von Parametern beschreiben, sondern lediglich den Aufpunkt und den Richtungsvektor als „kennzeichnend“ für die beschriebene Gerade betrachteten, vgl. [2], S. 140ff und [3], S. 377ff.

[3] Dies ist für zweiparametrige Objekte (Ebenen und Flächen) deutlich komplizierter als für einparametrige Gebilde, weshalb Überlegungen, wie Schüler funktionale Zusammenhänge zwischen Parametern und dadurch beschriebenen Punkten erfassen und verinnerlichen können, vor allem bei Parameterdarstellungen von Geraden bzw. Kurven ansetzen sollten.

Davon ausgehend können Parametergleichungen von Geraden eingeführt werden; auch die parameterabhängige Beschreibung verschiedener Kurven ist so möglich. Des Weiteren bieten sich Umkehrüberlegungen an, bei denen zu einzelnen Punkten von Geraden bzw. Kurven ermittelt wird, welchem Wert des Parameters sie zugeordnet sind. Vergleiche verschiedener Parametrisierungen derselben Objekte erscheinen in diesem Zusammenhang ebenfalls sinnvoll.

- Es lässt sich die *dynamische Sicht auf Geraden und andere Kurven als Bahnkurven* hervorheben, wodurch die Schüler mit dem Parameter eine konkrete Bedeutung verbinden. Die Interpretation des Parameters als Zeit stellt Bezüge zur Beschreibung von Bewegungen in der Physik her.

2 Einbeziehung von Grafiksoftware oder CAS für die Behandlung von Parameterdarstellungen

Die Einbeziehung von Elementen der Computergrafik in den Unterricht schafft Möglichkeiten, die beiden o. g. Herangehensweisen zur Herausarbeitung des Punktmengengedankens und dynamischer Aspekte umzusetzen. Die „Konstruktion" von Geraden und Kurven, aber auch von Ebenen, aus Punkten erfordert dabei lediglich eine Software, mithilfe derer die Schüler entsprechende grafische Darstellungen anfertigen können.

Für die Herausbildung einer dynamischen Sicht auf Parameterdarstellungen eignet sich besonders die *Erstellung von Animationen* (Videos). Hierzu werden Positionen von Objekten oder auch die Position des Beobachters in Abhängigkeit von einem Zeitparameter beschrieben. Ein Argument für die Anfertigung von Animationen zur Erlangung einer dynamischen Sicht auf Parameterdarstellungen besteht darin, dass sich Schüler erfahrungsgemäß für die Generierung von Videos in überdurchschnittlichem Maße interessieren. Bei der Verwendung geeigneter Software sind dazu parameterabhängige Beschreibungen zwingend erforderlich.

Um durch Parameterdarstellungen gegebene Geraden und Kurven als Punktmengen „aufzubauen" sowie parameterabhängige Animationen zu erstellen, können u. a. die 3D-Grafiksoftware POV-Ray sowie Computeralgebrasysteme (CAS) wie Maple, Mathematica und MuPAD genutzt werden. Diese Softwarepakete lassen sich für Visualisierungen und Berechnungen auch an anderen Stellen des Stoffgebietes Analytische Geometrie einsetzen.[4] Im Folgenden werden Beispiele unter Verwendung von POV-Ray sowie MuPAD dargestellt.

3 Geraden und Ebenen als Punktmengen

3.1 Einführung von Parametergleichungen durch die Betrachtung einzelner Punkte

Zur Einführung der Parameterdarstellung von Geraden kann den Schülern eine Aufgabe folgender Art gestellt werden:

Gegeben sind der Punkt $P(0{,}5;1;1{,}5)$ und der Vektor $\vec{a}=\begin{pmatrix}-2{,}5\\1\\-1{,}5\end{pmatrix}$.

- Stellen Sie den Punkt P sowie den Vektor $\vec{a}$ (als Pfeil, beginnend an P) dar.
- Stellen Sie die Punkte $P+0{,}5\cdot\vec{a}$, $P+\vec{a}$, $P+1{,}5\cdot\vec{a}$, $P+2\cdot\vec{a}$ sowie $P-0{,}5\cdot\vec{a}$, $P-\vec{a}$, $P-1{,}5\cdot\vec{a}$ und $P-2\cdot\vec{a}$ dar.
- Betrachten Sie die Darstellung aus verschiedenen Richtungen.

Eine mögliche Lösung dieser Aufgabe unter Verwendung von POV-Ray sowie auf der Internetseite [5] zur Verfügung stehender Ergänzungen für die einfache Darstellung von Objekten der analytischen Geometrie („anageo"–Vorlagen) zeigt Abbildung 1.

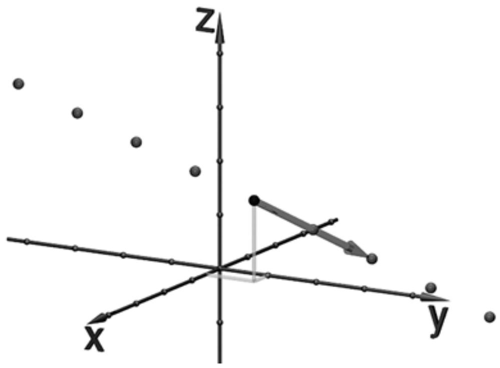

Abbildung 1: Punkte einer Geraden

[4] Einen Überblick über die Einbeziehung von Elementen der 3D-Computergrafik (unter Nutzung von POV-Ray) gibt [1]. POV-Ray ist unter **www.povray.org** frei verfügbar. Kurze Anleitungen zur Verwendung dieser Software speziell für den Unterricht im Stoffgebiet Analytische Geometrie sowie zugehörige Vorlagen und Ergänzungen stehen auf der Internetseite [5] zur Verfügung.

Um Abbildung 1 zu generieren, sind folgende Befehle einzugeben:

```
#declare a = <-2.5,1,-1.5>;
#declare P = <0.5,1,1.5>;
pluspunkt(P, schwarz)
vektoranpunkt(P, a, silbergrau)
punkt(P–2*a, blau_matt)
punkt(P–1,5*a, blau_matt)
    ...weitere 6 Punkte
```

Bei Verwendung von MuPAD entsteht durch die folgenden Eingaben eine (interaktiv drehbare) Grafik, wie in Abbildung 2 dargestellt.

```
a:=matrix([-2.5,1,-1.5])
P:=matrix([0.5,1,1.5])
Pfeila:=plot::Arrow3d(P,P+a)
PunktP:=plot::Point3d(P)
PunktMinus20:=plot::Point3d(P-2*a)
        ...weitere 6 Punkte
PunktPlus20:=plot::Point3d(P+2*a)
plot(Pfeila, PunktP, PunktMinus20,
     PunktMinus15, PunktMinus10,
     PunktMinus05, Punkt05,
     Punkt10, Punkt15, Punkt20)
```

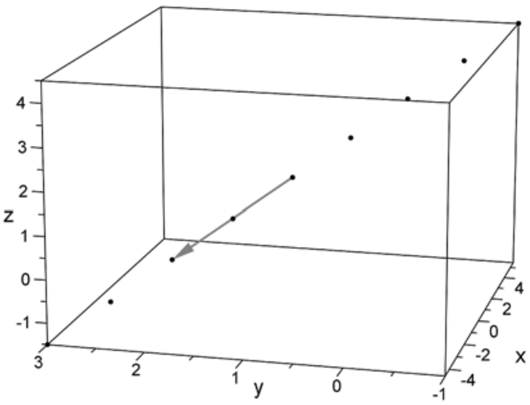

Abbildung 2: Punkte einer Geraden (MuPAD)

Anhand einer der beiden Abbildungen wird spätestens nach Betrachtung aus verschiedenen Richtungen deutlich, dass alle dargestellten Punkte auf einer Geraden liegen. Nach der Darstellung einer größeren Zahl von Punkten durch die Verkleinerung der Parameterabstände zeigt sich, dass alle Punkte der durch P verlaufenden Geraden, deren Richtung durch $\vec{a}$ gegeben ist, in der Form $P+t\cdot\vec{a}$ mit $t\in\mathbf{R}$ dargestellt werden können.

3.2 Nutzung von Schleifen oder Prozeduren für die Darstellung großer Zahlen von Punkten

Um größere Zahlen von Punkten zu generieren und somit tatsächlich sichtbar werden zu lassen, dass durch das Einsetzen beliebiger Parameter (bzw. Parameterpaare) in die Parametergleichungen von Geraden, Kurven und Ebenen die Objekte „vollständig aufgebaut" werden können, wäre es sehr mühsam, für jeden darzustellenden Punkt eine Zeile in das Programm einzugeben (wie oben beschrieben). Durch die Nutzung elementarer Programmierkonstrukte ist es hingegen leicht möglich, so große Anzahlen von Punkten zu generieren, dass sich das Ergebnis nicht mehr sichtbar von Geraden bzw. Strecken[5], Kurvenstücken oder Teilen von Ebenen unterscheidet. Durch die schrittweise Erhöhung der Zahl dargestellter Punkte können sich die Schüler einen „plastischen Eindruck" vom Punktmengencharakter geometrischer Objekte verschaffen.

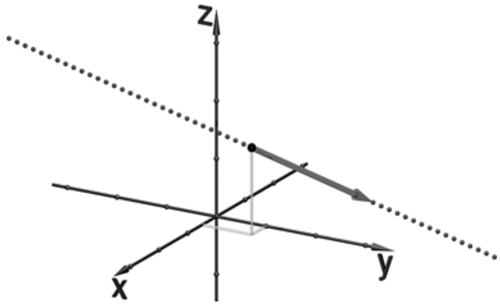

Abbildung 3: 60 Punkte einer Geraden

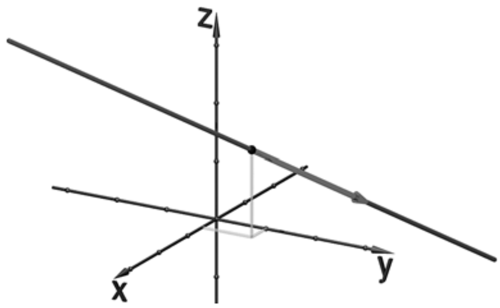

Abbildung 4: 400 Punkte einer Geraden

Um die in den Abbildungen 3 und 4 dargestellten Grafiken zu erzeugen, kann in POV-Ray eine Schleife genutzt werden:

```
#declare i=-200;
  #while (i <= 200)
    punkt(P+i*a/100  blau_matt)
  #declare i=i+1;
  #end
```

Solange die Bedingung $i\leq 200$ erfüllt ist, wird bei dieser Anweisung eine kleine Kugel mit dem Mittelpunkt $P+\frac{i}{100}\cdot\vec{a}$ erzeugt und der Wert der Schleifenvariablen i um 1 erhöht. Durch Veränderung der Werte 200 und

[5] Nicht berücksichtigt wird bei den beschriebenen Überlegungen die unendliche Ausdehnung von Geraden; natürlich können nur endliche Parameterwerte betrachtet werden. Dieses Problem gilt für jede grafische Darstellung, unabhängig von den verwendeten Hilfsmitteln. Die Unendlichkeit der Ausdehnung kann den Schülern nur mitgeteilt werden, während grafische Darstellungen immer nur Strecken zeigen. Diese Frage wird bereits in der S I, ansatzweise sogar in der Grundschule thematisiert.

100 lassen sich beliebig viele „Punkte“ in wählbaren Parameterintervallen erzeugen.

Bei der Verwendung des CAS MuPAD ist die Verwendung einer Prozedur der einfachste Weg zur Darstellung sehr vieler Punkte einer Geraden. Eine zu Abbildung 4 analoge Darstellung kann durch die folgende Prozedur und ihren 400-maligen Aufruf in einer plot-Anweisung erzeugt werden:

```
Puenktchen := proc(i)
begin
  plot::Point3d(P+i/100*a);
end_proc:

plot(Puenktchen(i) $ i=-200..200):
```

Durch eine verschachtelte Schleife bzw. eine Prozedur in 2 Variablen, lassen sich analog zu dem für Geraden beschriebenen Vorgehen Ebenen „punktweise aufbauen“; die folgenden Anweisungen und die zugehörige Abbildung 5 verdeutlichen das Vorgehen unter Verwendung von POV-Ray, auch hierbei lässt sich natürlich die Zahl der dargestellten Punkte erhöhen.

```
#declare P = <1,1,1> ;
#declare a = <-1,1,1.5> ;
#declare b = <1,-1.5,0.5> ;
#declare j=-4;
  #while (j <= 4)
    #declare i=-4;
      #while (i <= 4)
        punkt(P + i*a/2 + j*b/2
               rot_matt)
      #declare i=i+1;
      #end
  #declare j=j+1;
  #end
```

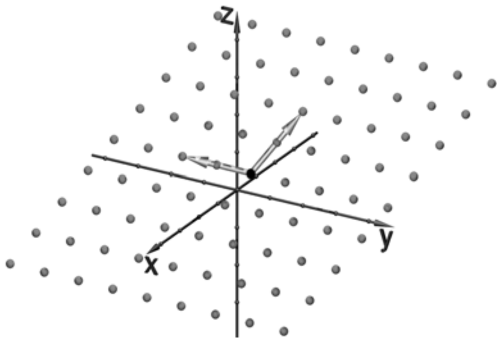

Abbildung 5: 81 Punkte einer Ebene

Die Verwendung von Schleifen bzw. Prozeduren geht über die im Mathematikunterricht im Allgemeinen behandelten Inhalte hinaus und ist eher Gegenstand des Informatikunterrichts. Jedoch wird anhand der dargestellten Beispiele deutlich, dass dadurch auch ein Beitrag zu genuinen Zielen des MU, wie zur Herausbildung eines Verständnisses von Geraden und Ebenen als Punktmengen erreicht werden kann. Eine Beschränkung auf Geraden und Ebenen erscheint dabei weder notwendig noch sinnvoll, wie die im letzten Abschnitt dieses Beitrags beschriebenen Überlegungen zur Darstellung einfacher Kurven durch Parametergleichungen zeigen werden.

4 Die Zeit als Parameter – Generieren einfacher Videos

Wie bereits erwähnt wurde, ist die Erstellung von Videos für viele Schüler eine sehr motivierende Aufgabe. In geeigneten CAS und in skriptgesteuerter Grafiksoftware wie MuPAD müssen dazu Koordinaten oder andere Größen in Abhängigkeit von einem speziellen Parameter, der Zeit, ausgedrückt werden. In POV-Ray hat dieser Parameter den Namen **clock**. So ist es möglich, ein Video mit einer geradlinig gleichförmigen Bewegung einer kleinen blauen Kugel durch

```
#declare a = <-2.5,1,-1.5>;
#declare P = <0.5,1,1.5>;
punkt(P+2*clock*a  blau_matt)
```

zu generieren.[6] Anschaulicher wird die Bewegungsbahn, wenn gleichzeitig die „Spur“ des sich bewegenden Objekts als Strecke zwischen dem Ausgangspunkt und der jeweils erreichten Position dargestellt wird. Dies ist in POV-Ray u. a. durch die bereits beschriebene Veranschaulichung von Geraden bzw. Strecken als Punktmengen möglich. Um in MuPAD die Animation eines Punktes auf einer geradlinigen Bahn und gleichzeitig die jeweils zurückgelegte Strecke darzustellen, lassen sich folgende Anweisungen nutzen:

```
P:=matrix([[0.5], [1], [1.5]]):
a:=matrix([[-2.5], [1], [-1.5]]):
Kugel:=plot::Sphere(0.2,
                  P+t*a, t = 0..2):
Strecke:=plot::Line3d(P, P+t*a,
                          t = 0..2):
plot(Kugel, Strecke):
```

Für den Zeitparameter t werden dabei in MuPAD „überschüssige Darstellungsbereiche“ (die für die Beschreibung der Objekte „Kugel“ bzw. „Strecke“ nicht nötig sind) angegeben, dabei handelt es sich um die Parameterintervalle, innerhalb derer die Animation erfolgt.

[6] Zu technischen Aspekten der Erstellung von Videos mithilfe von POV-Ray sei auf die Hilfe des Programms oder eine unter [5] (Rubrik Downloads) bereit stehende Kurzanleitung verwiesen. Die Anweisung „punkt“ ist kein Bestandteil der POV-Ray-Syntax, sondern wird durch die bereits erwähnten (ebenfalls auf der Internetseite [5] verfügbaren) „anageo“-Erweiterungen den POV-Ray-Anweisungen hinzugefügt. Statt dessen könnte aber auch das POV-Ray-interne Kugelobjekt (sphere) genutzt werden.

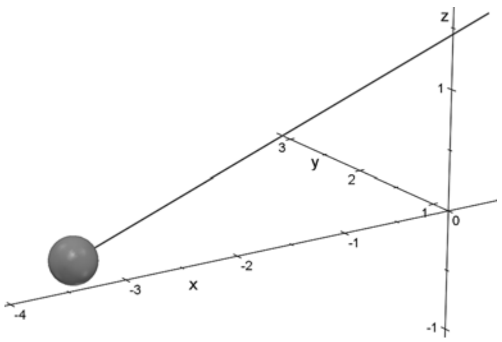

Abbildung 6: Animation einer Kugel auf einer Geraden (MuPAD-Grafik)

Parameterdarstellungen erhalten bei Animationen einen Aspekt, der die geometrische Gestalt dieser Objekte nicht beeinflusst, nämlich die Geschwindigkeit der Bewegung. So beschreiben z. B. die beiden Parameterdarstellungen

(1) $X(t) = P + t \cdot \vec{a}$ und

(2) $X(t) = P + t^2 \cdot \vec{a}$

(jeweils mit $t \in \boldsymbol{R}^+$) dieselbe Halbgerade. Werden diese Parametergleichungen verwendet, um Animationen zu generieren, so ergibt (1) eine gleichförmige und (2) eine gleichmäßig beschleunigte Bewegung auf dieser Halbgeraden. In Abb. 7 ist dies durch die Abstände der Punkte erkennbar; zwischen zwei benachbarten Punkten verstreicht jeweils gleich viel Zeit.

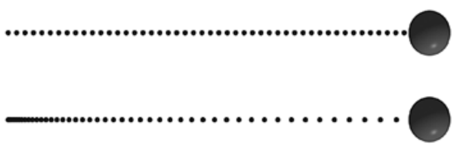

Abbildung 7

Bei etwas komplexeren Animationen ist es oft erforderlich, in verschiedenen Zeitintervallen unterschiedliche Funktionsterme für Positionen in Abhängigkeit vom Zeitparameter zu verwenden oder auch andere Größen zu animieren. Dazu lassen sich Verzweigungen (**`if-else`**-Anweisungen) einsetzen. Dies sei anhand des schrittweisen „Aufbaus“ einer Ebene illustriert. Zunächst (für Parameterwerte **`clock`** < 1) erfolgt der Aufbau einer Geraden entlang eines Richtungsvektors der gegebenen Ebene, danach (**`clock`** = 1..2) wird die Ebene aus dazu parallelen, entlang des anderen Richtungsvektors der Ebene verschobenen, Geraden konstruiert. In POV-Ray lässt sich dieses Vorgehen z. B. folgendermaßen realisieren:

```
#declare a=<-2.5,1,-1.5>;
#declare b=<1.5,1,-0.2>;
#declare P=<0.5,1,1.5>;
#if (clock<1)
 #declare i=-2*clock*20;
  #while (i <= 2*clock*20)
    punkt(P+i*a/20 blau_matt)
  #declare i=i+1;
  #end
#else
 #declare j=-2*20*(clock-1);
  #while (j <= 2*20)
    #declare i=-2*20;
     #while (i <= 2*20)
       punkt(
          P+i/20*a+j/20*(clock-1)*b
          blau_matt )
     #declare i=i+1;
     #end
  #declare j=j+1;
  #end
#end
```

Am Ende der Animation wird die Ebene dabei durch $40 \cdot 40$ kleine Kugeln repräsentiert. Abbildung 8 zeigt zwei Zwischenschritte dieser Animation.[7]

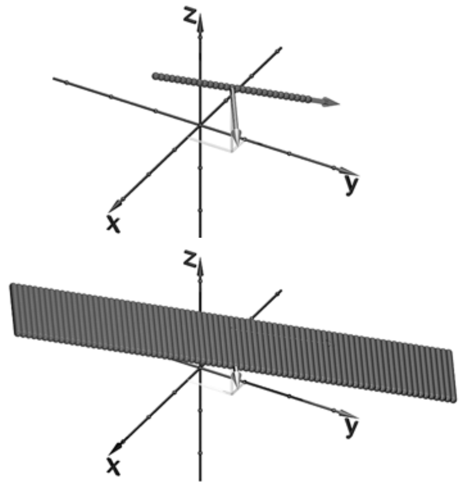

Abbildung 8: Bilder eines Videos zum Aufbau einer Ebene aus Punkten

5 Parameterdarstellungen von Kreisen und einigen weiteren Kurven

Die Darstellung von durch Parameterdarstellungen gegebenen geometrischen Objekten durch Punktmengen sowie die Erzeugung von Animationen durch die zeitabhängige Beschreibung von Koordinaten ist nicht auf lineare Objekte (Geraden und Ebenen) beschränkt – im Gegenteil: wesentlich interessantere und in Bezug auf die Form ansprechendere Überlegungen und Darstellungen

[7] Das Video mit der Animation lässt sich, wie alle anderen hier erwähnten Videos, auf der Internetseite [5] betrachten. Auch die Quelldateien stehen dort zur Verfügung, so dass Variationen daran vorgenommen und entsprechend veränderte Videos erstellt werden können.

sind anhand von Kurven (sowie auch Flächen) möglich. Als Ausgangspunkte für die Beschreibung von Kurven durch Parameterdarstellungen eignen sich Geraden, deren Parametergleichungen sich z. B. anhand physikalischer Überlegungen variieren lassen, sowie Kreise.

5.1 Parameterdarstellungen von Kreisen

Die wichtigste Grundlage für *Parameterdarstellungen von Kreisen in der Ebene* haben die Schüler bereits in der S I kennen gelernt – gewöhnlich werden dort die Sinus- und die Kosinusfunktion am Einheitskreis eingeführt; mit den in Abbildung 9 verwendeten Bezeichnungen durch

$$\sin\alpha = y_\alpha\ ,\ \ \cos\alpha = x_\alpha\ .$$

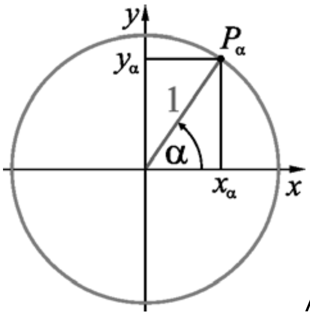

Abbildung 9

Eine Verallgemeinerung auf Kreise in Mittelpunktslage mit beliebigem Radius *r* ist leicht möglich, woraus die Parameterdarstellung

$$x(\alpha) = r\cdot\cos\alpha\ ,\ \ y(\alpha) = r\cdot\sin\alpha$$

mit $\alpha \in [0;2\pi)$ eines Kreises der Ebene, dessen Mittelpunkt im Koordinatenursprung liegt, hergeleitet werden kann.[8] Sollen verschiedene Größen animiert werden, so ist es im Sinne der Übersichtlichkeit sinnvoll, das Intervall des Animationsparameters (der Zeit) zu normieren und die obige Parameterdarstellung in der Form

$$x(t) = r\cdot\cos(2\pi\cdot t)\ ,\ \ y(t) = r\cdot\sin(2\pi\cdot t)$$

mit $t \in [0;1)$ zu schreiben. Parameterdarstellungen von Kreisen, die im Raum auf Koordinatenebenen oder dazu parallelen Ebenen liegen, ergeben sich daraus, indem eine der drei Koordinaten als Konstante dargestellt wird, z. B. $z(t) = h$. Ausgehend von diesen Überlegungen können die Schüler eine Animation einer kreisförmigen Bewegung generieren. In POV-Ray lässt sich z. B. durch die Anweisungen

```
#declare r = 10
sphere { < r*cos(2*pi*clock), 0,
           r*sin(2*pi*clock) >  1 }
```

die Animation einer Kugel auf einer Kreisbahn erzeugen.[9] Da für die Erlangung eines Überblicks über den Ablauf von Animationen oft die Darstellung der verwendeten Bewegungsbahnen sinnvoll ist, kann zusätzlich die „Spur" der Kugel dargestellt werden, so dass bei Betrachtung der Animation sichtbar wird, welche Bahn das Objekt zurückgelegt hat. Die Vorgehensweise dazu entspricht der bereits für Geraden beschriebenen Erzeugung einer Vielzahl kleiner Kugeln (siehe Abbildung 10); im Falle einer kreisförmigen Bahn z. B. mithilfe folgender Schleife

```
#declare i=0;
#while (i <= 200*clock)
  sphere { < r*cos(2*pi*(i/200)),
     0, r*sin(2*pi*(i/200)) > 0.1 }
#declare i=i+1;
#end
```

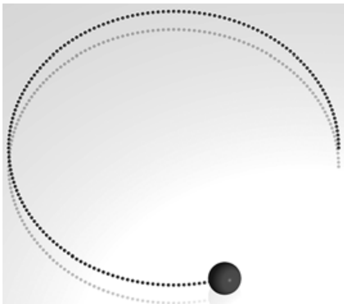

Abbildung 10

5.2 Kameraanimationen

Wird anstelle eines geometrischen Objektes die Position der „Kamera", von welcher aus die Szene betrachtet wird, zeitabhängig beschrieben, so entsteht eine Animation, bei der sich der Blick auf alle Objekte einer Szene verändert. So kann z. B. mittels

```
camera { location
           < r*cos(2*pi*clock), 4,
             r*sin(2*pi*clock)   >
     angle 12  look_at <0,0,0> }
```

in POV-Ray ein Kameraflug auf einer kreisförmigen Bahn simuliert werden, wobei die Kamera stets auf den Koordinatenursprung

[8] Durch die Addition von Mittelpunktskoordinaten lässt sich diese Parameterdarstellung auf beliebige Kreise in der Ebene verallgemeinern: $x(\alpha) = r\cdot\cos\alpha + x_M$, $y(\alpha) = r\cdot\sin\alpha + y_M$.

[9] Auch mithilfe der CAS Maple, Mathematica und MuPAD können Animationen erstellt werden, eine sich auf einer Kreisbahn in einer zur *x*-*y*-Ebene parallelen Ebene mit konstanter Winkelgeschwindigkeit bewegende Kugel mit dem Radius r_k lässt sich in MuPAD z. B. durch

```
plot::Sphere(rk, [r*cos(t),r*sin(t),h],
                               t = 0..2*PI)
```

animieren.

gerichtet bleibt. Einige Beispiele zu Kameraanimationen befinden sich auf der Internetseite [5].

Hinsichtlich der notwendigen mathematischen Überlegungen ist es unbedeutend, ob die Schüler die Animation eines sich auf einer Kreisbahn bewegenden Objektes (z. B. einer Kugel) oder eine Kameraanimation erstellen, bei der sich die Sicht auf eine gesamte Szene verändert; erfahrungsgemäß ist Letzteres für die Mehrzahl der Schüler interessanter. Um jedoch Bewegungskurven sichtbar werden zu lassen, empfiehlt es sich, nicht nur Kameraanimationen zu erstellen, sondern in Ergänzung dazu auch sichtbare Objekte zu animieren.[10]

5.3 Variationen von Kreisen: Spiralen und Schraubenlinien

Nach der Behandlung der Parametergleichungen von Kreisen liegt es nahe, durch geeignete Veränderungen daran „verwandte" Kurven parametrisch zu beschreiben. Dazu lassen sich folgende Fragen, die von den Schülern zu erwarten sind, aufgreifen:

- Wie kann die Kamera um ein Objekt kreisen und sich diesem gleichzeitig annähern?
- Wie lässt sich bei einer kreisförmigen Bewegung der Kamera gleichzeitig deren Höhe ändern, so dass Objekte aus unterschiedlichen Höhen betrachtet werden.

Natürlich lassen sich beide Fragen auch so stellen, dass sie den Verlauf von Kurven betreffen. Für die Realisierung der zuerst genannten Eigenschaft kann (zumindest mit Hilfen) von den Schülern herausgearbeitet werden, dass dazu die Konstante *r*, die für den Radius des zuvor betrachteten Kreises eingesetzt wurde, durch eine Funktion $r(t)$ des sich zeitlich verändernden Parameters *t* ersetzt werden muss, z. B. durch $r \cdot (1 - t)$, falls sich der Abstand zum Mittelpunkt im Verlauf der Animation von r auf 0 verringern soll (für $t \in [0;1]$). Durch diese Überlegung ergibt sich unmittelbar die *Parameterdarstellung einer archimedischen Spirale*:

$$x(t) = r \cdot (1-t) \cdot \cos(2\pi \cdot t)$$
$$y(t) = r \cdot (1-t) \cdot \sin(2\pi \cdot t) \qquad t \in [0;1]$$
$$z(t) = h\,.$$

Auf dieser Grundlage können geometrische Objekte oder die Kamera entlang einer archimedischen Spirale bewegt werden, wobei die oben für kugelförmige Bahnkurven beschriebenen Befehle entsprechend zu variieren sind. Allerdings wird die Form der Spirale besser sichtbar, wenn zwei Umdrehungen durchlaufen werden, was durch Ersetzen von $(2\pi \cdot t)$ durch $(4\pi \cdot t)$ in den trigonometrischen Termen der Parameterdarstellung erreicht wird (siehe Abb. 11).

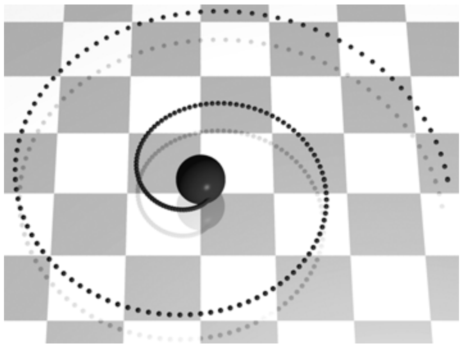

Abbildung 11: Archimedische Spirale

Bei der Diskussion der o. g. zweiten Frage dürfte es den Schülern leicht fallen, zu erkennen, dass zur zeitabhängigen Veränderung der „Höhe" die vorher konstant gehaltene dritte Koordinate durch eine Funktion des Parameters zu ersetzen ist. Wird dafür eine lineare Funktion gewählt (im einfachsten Falle $y = t$ bzw. $z = t$ – je nachdem, welche Koordinate bei der Beschreibung von Kreisen konstant gehalten wurde), so entsteht aus der Kreisgleichung die *Gleichung einer Schraubenlinie*, z. B.:

$$x(t) = r \cdot \cos(4\pi \cdot t)$$
$$y(t) = t \qquad t \in [0;1]$$
$$z(t) = r \cdot \sin(4\pi \cdot t)\,.$$

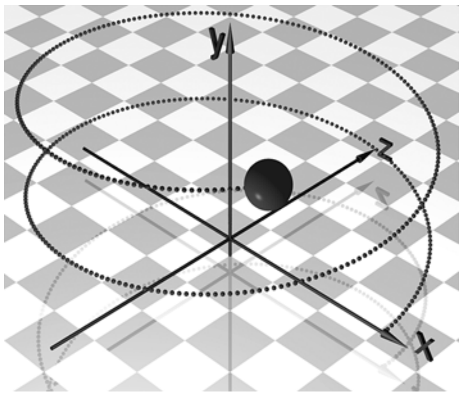

Abbildung 12: Schraubenlinie

[10] Mit der Animation sichtbarer Objekte zu beginnen, lässt sich leicht dadurch motivieren, dass auch für Schüler, die „Kamerafahrten" erstellen möchten, damit der Ablauf von Animationen verständlicher wird und sie die Bahn, auf der sich später die Kamera bewegen soll, zunächst sehen können. Bei der Arbeit mit Studierenden konnte die Erfahrung gemacht werden, dass einige Studierende, die sich zunächst mehr für Kameraanimationen interessierten, dann Objektanimationen und insbesondere die Visualisierung der dabei bestehenden Bewegungskurven sehr interessant fanden.

Durch die Kombination beider Überlegungen, die vom Kreis zur Spirale bzw. zur Schraubenlinie führten (parameterabhängige Beschreibungen des Radius und der „Höhe“ in der ursprünglichen Parameterdarstellung eines Kreises) entsteht bei Verwendung linearer Funktionen in t eine konische Spirale mit einer Parameterdarstellung der Form

$$\begin{aligned} x(t) &= r\cdot(1-t)\cdot\cos(4\pi\cdot t) \\ y(t) &= t \qquad\qquad\qquad\qquad t\in[0;1] \\ z(t) &= r\cdot(1-t)\cdot\sin(4\pi\cdot t)\ . \end{aligned}$$

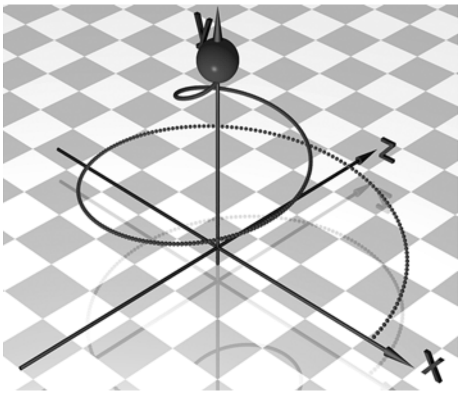

Abbildung 13: Konische Spirale

Weitere Variationen der bisher betrachteten Kurven ergeben sich aus der Verwendung nichtlinearer Funktionsterme in t für die Höhe bzw. den Radius. So kann die Aufgabe gestellt werden, die Parameterdarstellung der Schraubenlinie so zu verändern, dass sich deren Punkte zunächst sehr langsam und später schneller von denen des ursprünglichen betrachteten Kreises entfernen. Ebenso sind Variationen der archimedischen Spirale möglich.

Durch Multiplikation der trigonometrischen Terme in den Parameterdarstellungen mit unterschiedlichen Faktoren (anstelle eines einheitlichen Radius) ist auch die Erzeugung von Ellipsenbahnen sowie Bahnen auf „elliptischen Spiralen“ und „elliptischen Schraubenlinien“ möglich.[11]

Durch Variationen an Parameterdarstellungen von Kurven und die dadurch erfolgende Beschreibung „neuer“ Kurven ergeben sich reichhaltige Möglichkeiten für funktionale Überlegungen, bei denen die Schüler ausgehend von qualitativen Beschreibungen gewünschter Kurvenverläufe überlegen, durch welche Funktionsterme diese entstehen können und ihre Überlegungen mithilfe der Software überprüfen.

[11] Mit POV-Ray und MuPAD erstellte Beispiele mit Animationen auf Ellipsenbahnen stehen (einschließlich der zugehörigen Beschreibungen, die sich natürlich vielfältig varieeren lassen) auf der Internetseite [5] zur Verfügung.

5.4 Vektorielle Parameterdarstellungen – der schräge Wurf

Sollen im Unterricht vektorielle Beschreibungen und Vorgehensweisen im Vordergrund stehen, so bietet es sich an, von Parameterdarstellungen von Geraden auszugehen. Wie bereits ausgeführt wurde, erhält der Parameter, wenn er als Zeit interpretiert wird, einen neuen Aspekt, der die geometrische Gestalt der beschriebenen Objekte nicht beeinflusst, aber Auswirkungen auf die Geschwindigkeit von Bewegungen hat. Bei der Arbeit mit Animationen lassen sich somit Verbindungen zum Physikunterricht herstellen, funktionale Aspekte durch die Betrachtung unterschiedlicher Funktionen $f(t)$, die den Zeitparameter ersetzen, vertiefen sowie einfache Simulationen erstellen.

Nach den gängigen Rahmenplänen werden im Physikunterricht der S II Bewegungen vektoriell beschrieben, wobei der schräge Wurf Unterrichtsgegenstand ist. Dieser kann als eine aus einer gleichförmigen und einer gleichmäßig beschleunigten Bewegung zusammengesetzte Bewegung aufgefasst werden. Durch die Addition einer in t linearen Komponente und des mit t^2 multiplizierten Beschleunigungsvektors in der Gleichung

$$\vec{x} = \vec{x}_0 + \vec{v}\cdot t + \frac{1}{2}\vec{g}\cdot t^2$$

des schrägen Wurfes ergibt sich die Wurfparabel. Eine entsprechende Animation lässt sich in POV-Ray durch die folgenden Anweisungen generieren.

```
#declare x0 = <-2.5,0,0>;
#declare v0 = <5,5,0>;
#declare g = <0,-10,0>;
sphere { x0 + v0*clock
         + g/2*clock*clock  0.25}
```

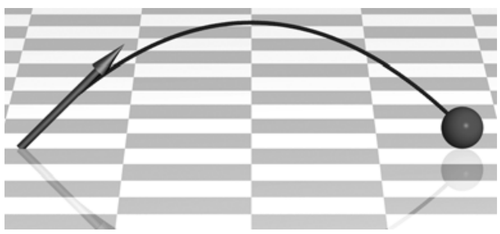

Abbildung 14: Wurfparabel

Fazit

Die vorgestellten Beispiele zeigen, dass es mit recht elementaren mathematischen Mitteln, die an den Unterricht der S I anknüpfen, möglich ist, interessante Kurven zu modellieren und auf dieser Grundlage Animationen zu erstellen. Allerdings darf die dazu erforderliche Zeit nicht unterschätzt werden. Selbst für relativ elementar erscheinende funktionale Überlegungen, die z. B. von Kreisen zu Spiralen oder Schraubenlinien führen, brauchten in von mir durchgeführten Seminaren auch Studierende recht lange – da sie diese Überlegungen als interessant empfanden und in Experimentierfreude verfielen, waren sie jedoch bereit, relativ viel Freizeit dafür aufzuwenden. Dies dürfte auch bei vielen Schülern der Fall sein, da die Erstellung interessanter Videos, wie mehrfach erwähnt, überaus motivierend für Jugendliche ist.

Für die Beschreibung der geometrisch und auch hinsichtlich der Erstellung von Kameraanimationen besonders interessanten Spiralen ist die Vektorrechnung nicht erforderlich; hierfür ist es sinnvoller, mit Koordinatenbeschreibungen zu arbeiten. Allerdings sind auch ausgehend von vektoriellen Geradengleichungen interessante Überlegungen zu Bewegungsbahnen möglich. Aus mathematikdidaktischer Sicht halte ich den beschriebenen Gegenstandsbereich vor allem deshalb für lohnenswert, weil die Beschäftigung damit anspruchsvolle Überlegungen zu funktionalen Zusammenhängen „anstößt“, die bei der gegenwärtig dominierenden Behandlung von Parameterdarstellungen im Unterricht oftmals nicht hinreichend auftreten.

Die Veranschaulichung von Geraden, Ebenen und Kurven durch eine große Zahl von Punkten erfordert die Einbeziehung einiger typischer Inhalte der Informatik in den Mathematikunterricht (Schleifen bzw. Prozeduren und für komplexere Animationen auch Verzweigungen). Diese können anhand der behandelten Inhalte plausibel gemacht und genutzt werden, ohne dass dafür ein zu großer Zeitaufwand betrieben werden müsste. Natürlich wäre es hierbei auch sinnvoll, mit dem Fach Informatik zu kooperieren, falls Schüler der unterrichteten Kurse den Informatikunterricht besuchen.

Literatur und Internetseiten

[1] Filler, A.: Dreidimensionale Computergrafik und Analytische Geometrie. In: mathematica didactica 24 (2001), 2, S.21–56.

[2] Tietze, U.-P.: Didaktik der Analytischen Geometrie und Linearen Algebra (unter Mitarbeit von Schroth, P.; Wittmann, G.). In: Tietze, U.-P.; Klika, M.; Wolpers, H. (Hrsg.): Mathematikunterricht in der Sekundarstufe II, Bd.2. Vieweg, Braunschweig, 2000.

[3] Wittmann, G.: Schülerkonzepte zur Analytischen Geometrie. Franzbecker, Hildesheim, 2003.

[4] MaDiN (Mathematik-Didaktik im Netz) - Didaktik der Oberstufengeometrie und Linearen Algebra:
`http://www.madin.net`
(letzter Zugriff: 1. 11. 2005).

[5] 3D-Computergrafik und die Mathematik dahinter (mit Beispielen, Materialien und Ergänzungen zu dem vorliegenden Beitrag):
`http://www.afiller.de/3dcg`
(letzter Zugriff: 1. 11. 2005).