
Der unmögliche Freistoß

Wolfgang Bock und Andreas Roth

Zusammenfassung

Die Autoren befassen sich mit der Ableitung und Bearbeitung eines Modellierungsprojektes aus der populären Sportart Fußball: Ein Freistoß wird unter Beachtung der gegebenen physikalischen Effekte mathematisch modelliert und simuliert. Der Fokus liegt auf der möglichen Durchführung dieses Modellierungsprojekts mit Schülerinnen und Schülern der Sekundarstufe II.

1 Einleitung/Problemstellung

Im Fußball zählen Freistoßstore zu den spektakulärsten Arten ein Tor zu erzielen. Der Schütze muss hierbei den Ball auf eine Flugbahn befördern, die das Tor trifft, aber auch schnell und

unvorhersehbar genug ist, um den Torwart zu überlisten (Abb. 1). Daher stellt sich die Frage, wie ein Spieler den Ball schießen soll, damit er gute Chancen hat ein Tor zu erzielen.

Aus dieser Fragestellung lassen sich viele verschiedene Problemstellungen ableiten, die von Schülerinnen und Schüler der Klasse 5 bis zu Studierenden des Masterstudiums bearbeitbar sind. Hier liegt der Fokus auf Problemstellungen für die Sekundarstufe II. Eine sehr allgemeine Fragestellung ist der Zusammenhang der Trajektorie des Balles mit der Abschusstechnik, wobei angenommen wird, dass der Spieler jede Schusstechnik perfekt beherrscht. Als Daten können dabei entweder Videos von Freistößen zur Verfügung gestellt oder auch selbst Szenarien ausprobiert und dabei Daten selbst erhoben werden.

Das Projekt wurde schon mehrmals von den Autoren im Rahmen der mathematischen Modellierungswoche und mathematischen Modellierungstagen durchgeführt. Hierbei arbeiteten Schü-

Die Erstveröffentlichung dieses Beitrags erfolgte auf dem Dokumentenserver KLUEDO der Technischen Universität Kaiserslautern mit dem *Uniform Resource Name* urn:nbn:de:hbz:386-kluedo-43376 unter Standardlizenzbedingungen gemäß der KLUEDO-Leitlinien vom 30.07.2015 und ist zugänglich über <https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/4337>.

W. Bock ✉

KOMMS, Fachbereich Mathematik, Technische Universität Kaiserslautern, Gottlieb-Daimler-Straße 48, 67663, Kaiserslautern, Deutschland

A. Roth

KOMMS, Fachbereich Mathematik, Technische Universität Kaiserslautern, Gottlieb-Daimler-Straße 48, 67663, Kaiserslautern, Deutschland

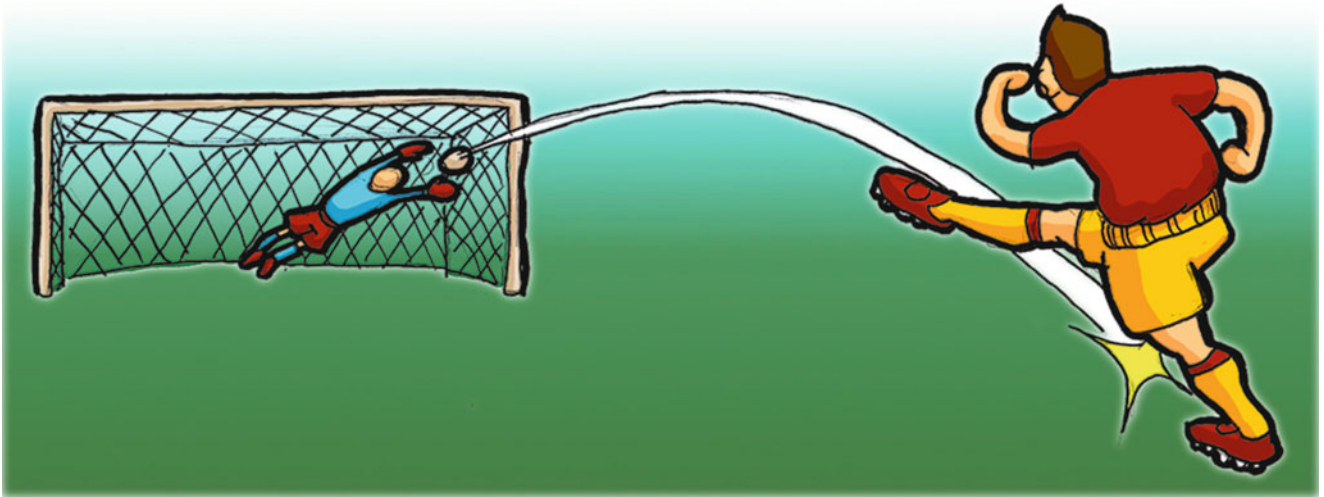


Abb. 1 Freistoßsituation im Fußball, Grafik: Andreas Roth, 2015

lerinnen und Schüler der Klassenstufen 11 und 12 jeweils in 5er bzw. 6er Gruppen zusammen. Pro Gruppe waren jeweils eine bzw. zwei Mathematiklehrkräfte anwesend, die sich aber in der ersten Arbeitsphase mit mathematischen Hinweisen zurückhalten sollten und nur bei technischen Problemen, wie etwa der Datenverarbeitung oder der Programmierung, eingriffen. Die Problemsteller blieben im Hintergrund und beantworteten nur Fragen zum Problemkontext.

In beiden Gruppen hatten mehrere Mitglieder Programmierkenntnisse und waren geübt im Umgang mit einer Tabellenkalkulation.

1.1 Anforderungen an die Schülerinnen und Schüler

Computerkenntnisse Die Erhebung von Flugbahn-
daten aus Videos ist mit einem Videobearbeitungs-
programm gut zu bewältigen. Für die Modellie-
rung der Trajektorie müssen je nach Komplexität
des Modells Differentialgleichungen numerisch
gelöst werden. hierzu kann entweder auf eine
gängige Programmiersprache oder auf eine Ta-
bellenkalkulation zurückgegriffen werden. Für die
Optimierung kann man dann etwa Schieberegler
einsetzen und die geeigneten Parametersätze durch
Ausprobieren finden. Es ist daher ausreichend

wenn Schüler und Schülerinnen gute Kenntnisse in
Tabellenkalkulation und rudimentäre Kenntnisse
(Schleifen, if/else-Verzweigungen) einer Program-
miersprache haben.

Physikkenntnisse Die Modelle basieren auf dem
schiefen Wurf. Daher ist es hilfreich, wenn die-
ser bereits als bekannt vorausgesetzt werden kann.
Dies ist auch wichtig, um das Konzept der zu-
sammengesetzten Bewegung zu nutzen, um die
Trajektorien mit Hilfe anderer Kräfte zu modifizie-
ren. Konzepte wie Magnuseffekt und Luftreibung
sind in der Regel nicht aus dem Unterricht be-
kannt. Hier kann es, zur Zeitersparnis, hilfreich
sein, den Schülerinnen und Schülern Formeln vor-
zugeben. Der Zusammenhang zwischen Leistung,
Kraft und Geschwindigkeit und die daraus resul-
tierenden Trajektorien sollten jedoch im Physikun-
terricht behandelt worden sein, kann aber auch aus
dem Physiklehrbuch nachvollzogen werden.

2 Physikalisches Modell und Weiterentwicklungen

Oft ist der schiefe Wurf das erste von Schülerseite
genutzte Modell zur Beschreibung der Trajektorie
des Balles. Grund hierfür ist, dass dieses Modell
bereits in Klasse 11 im Physikunterricht behandelt

wird und daher auch bei Internet- oder Buchrecherche schnell gefunden wird. Der schiefe Wurf ist eine zusammengesetzte Bewegung aus einer geradlinigen Bewegung parallel zur Spielfeldebene und einem dazu senkrechten Wurf.

Als Formeln für Strecke und Geschwindigkeit ergibt sich für einen schiefen Wurf der im Nullpunkt startet:

$$\begin{aligned}x(t) &= v_{x,0}t, & v_x(t) &= v_{x,0}, \\y(t) &= v_{y,0}t - \frac{1}{2}gt^2, & v_y(t) &= v_{y,0} - gt,\end{aligned}$$

wobei g der Vektor der Erdbeschleunigung und $v_{x,0}, v_{y,0}$ die Anfangsgeschwindigkeiten in x - und y -Richtung sind. Mittels Pythagoras ermittelt sich die Gesamtgeschwindigkeit zu

$$v(t) = \sqrt{v_x^2(t) + v_y^2(t)}.$$

Dieses recht einfache Modell kann bereits zu ersten Problemen auf Schülerseite führen. Aus dem Physikunterricht sind Weg-Zeit-Diagramme bekannt. Bei der Berechnung der Trajektorie spielt die Zeit jedoch nur eine untergeordnete Rolle. Vielmehr müssen die Schüler nun jedem Punkt in x -Richtung den zugehörigen Punkt in y -Richtung zuordnen. Die Berechnung der Trajektorie als Funktion $y(x)$ stellt daher eine erste Hürde dar, selbst wenn beim schiefen Wurf die x -Koordinate linear von der Zeit abhängt.

Der schiefe Wurf ist eine zweidimensionale Bewegung. Im realen Leben ist der Ball allerdings in einem drei-dimensionalen Raum. Ob man nun direkt in 3D-Raumkoordinaten startet oder zunächst in die Ebene projiziert, hängt vom Abstraktionsvermögen der Schülerinnen und Schüler ab. In den von den Autoren betreuten Gruppen wurde zunächst mit einem 2D-Modell begonnen, welches dann sukzessive auf 3 Dimensionen ausgebaut wurde.

2.1 Luftwiderstand

Überprüft man die Flugbahn einer Kugel beim Kugelstoßen in der Halle, so stimmt sie gut mit

der Wurfparabel des schiefen Wurfs überein. Vergleicht man die Wurfparabel mit einem Freistoß, so kommen erhebliche Abweichungen ins Spiel, die sich mit Hilfe des Modelles nicht erklären lassen. Eine Weiterentwicklung des schiefen Wurfs ist dann die Berücksichtigung des Luftwiderstandes F_W im Modell. Die Kraft, die aufgrund des Luftwiderstandes dem Flug eines Körpers entgegenwirkt ist

$$F_W = -\frac{1}{2}\rho c_w A v^2,$$

hierbei ist ρ die Luftdichte, c_w der Koeffizient des Luftwiderstandes und A die Angriffsfläche des Windes. Der Luftwiderstand hängt demnach vom Quadrat der Gesamtgeschwindigkeit ab. Um den Luftwiderstand nun in das Modell einfließen zu lassen, betrachtet man die auf den Ball in jedem Punkt wirkende Gesamtkraft:

$$F = F_G + F_W$$

wobei F_G die Gewichtskraft bezeichnet. Damit gilt für die Gesamtbeschleunigung $a(t) = \dot{v}(t)$ in jedem Punkt

$$\dot{v}(t) = \frac{1}{m} \begin{pmatrix} -\frac{1}{2}\rho c_w A v_x^2(t) \\ -g - \frac{1}{2}\rho c_w A v_y^2(t) \end{pmatrix}$$

also hängt die Änderung der Geschwindigkeit in jedem Punkt von der momentanen Geschwindigkeit ab. Dies kann etwa dadurch behandelt werden, indem man numerische Verfahren, wie das Euler Verfahren nutzt, siehe dazu Abschn. 3.

2.2 Magnus-Effekt

Will man dreidimensionale Flugbahnen beschreiben und auch den beim Fußball wichtigen Effekt berücksichtigen, so ist der Magnus-Effekt, bzw. die zugehörige Kraft bei einer Weiterentwicklung des Modells nötig. Der Effekt ist mit Hilfe der Relation zwischen Druck- und Geschwindigkeit einer idealen Strömung erklärbar. Überlagert man die

Umströmung eines nicht rotierenden Körpers und die wirbelfreie Zirkulationsströmung um einen in ruhender Luft rotierenden Körper, so ist die Strömungsgeschwindigkeit auf der Seite des Körpers, die sich in Strömungsrichtung dreht, größer als auf gegenüberliegenden Seite. Daher ist dort der Druck geringer und der Körper wird in diese Richtung gesogen. Für den Magnus-Effekt findet man

$$F_M = \frac{c_M \rho A}{2} \cdot \omega(e_r \times v),$$

wobei c_M der Magnus-Koeffizient, ρ die Luftdichte, A die Oberfläche des Balls und ω die Winkelgeschwindigkeit um die Drehachse e_r ist.

Nach Einbeziehen von Luftwiderstand und Magnus-Effekt erhält man als Modell für die Beschleunigung:

$$\dot{v} = \frac{F_G + F_W + F_M}{m}$$

Die Beschleunigung ändert sich, da sich auch die Gesamtgeschwindigkeit ändert in jedem Zeitpunkt. Die Berechnung der Trajektorie erfolgt dann mit den Techniken aus Abschn. 3.

3 Numerik

Sei $x(t) \in \mathbb{R}^3$ die gesuchte Ballposition in Abhängigkeit von der Zeit t , dann ist die zu lösende Gleichung gegeben durch

$$m\ddot{x} = F = F_G + F_W + F_M \quad (1)$$

wobei m die Masse des Balls, F_G die Gewichtskraft, F_W die Kraft durch Luftwiderstand und F_M die Kraft durch den Magnus-Effekt ist. Hierbei handelt es sich um eine gewöhnliche Differentialgleichung zweiter Ordnung. Es bietet sich an, diese für die weitere Verarbeitung auf ein System von Gleichungen erster Ordnung zu reduzieren, indem wir die Variable $\dot{x} = v \in \mathbb{R}^3$ einführen. Physikalisch entspricht diese der Geschwindigkeit und (1) lässt sich aufteilen in eine Gleichung für

die Position und eine für die Geschwindigkeit:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= \frac{F_G + F_W + F_M}{m} \end{aligned} \quad (2)$$

Die Kraftterme sind – wie bereits hergeleitet – gegeben durch

$$\begin{aligned} F_G &= -mg \\ F_W &= \frac{c_W \rho A}{2} \cdot v \\ F_M &= \frac{c_M \rho A}{2} \cdot \omega(e_r \times v) \end{aligned}$$

Fassen wir x und v im Vektor $X = (x, v)^T$ zusammen, so können wir (2) schreiben als

$$\dot{X} = MX + b \quad (3)$$

mit

$$M = \begin{pmatrix} 0 & I \\ 0 & A \end{pmatrix} \quad (4)$$

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5)$$

$$A = \begin{pmatrix} C_W & -C_M e_{r3} & C_M e_{r2} \\ C_M e_{r3} & C_W & -C_M e_{r1} \\ -C_M e_{r2} & C_M e_{r1} & C_W \end{pmatrix} \quad (6)$$

$$\begin{aligned} C_W &= -\frac{\rho A c_W}{2m} \\ C_M &= -\frac{\rho A c_M \omega}{2m} \end{aligned} \quad (7)$$

$$b = (0 \ 0 \ 0 \ 0 \ 0 \ -g)^T \quad (8)$$

Für solche Systeme von gewöhnlichen Differentialgleichungen erster Ordnung gibt es eine reichhaltige Lösungstheorie (Wirsching 2006). Ist die Funktion der rechten Seite Lipschitz-stetig und ergänzen wir das System um eine Anfangsbedingung $X(0) = X_0$, dann ist sichergestellt, dass eine eindeutige Lösung existiert. Da das System (2) linear in X ist, erfüllt es diese Eigenschaft natürlicherweise und hat eine eindeutige Lösung, wenn

wir eine Abschussposition und -geschwindigkeit vorgeben. Das bedeutet, dass es für eine gegebene Startposition und -geschwindigkeit genau eine richtige Flugkurve gibt. Kann oder will man diese nicht analytisch ausrechnen, so ist ein einfacher Weg durch die Numerik gegeben. Die Auswahl an Näherungsverfahren für Systeme wie (2) ist vielfältig (Deuffhard 2002). Allen Näherungsverfahren ist gemein, dass die kontinuierliche Zeitachse in diskrete Punkte t_0, t_1, \dots, t_n aufgeteilt wird und dass an jedem dieser Zeitpunkte eine Näherung der Lösung $X(t_i)$ errechnet wird. Im vorliegenden Fall soll das wahrscheinlich einfachste und am leichtesten umzusetzende Verfahren kurz vorgestellt werden. Es handelt sich um das *Euler*-Verfahren. Sei $t_{i+1} - t_i = \tau$ unsere feste Schrittweite für alle $i = 0, 2, \dots, n - 1$ und X_i die Näherung des tatsächlichen Wertes $X(t_i)$, dann berechnen wir X_{i+1} am nächsten Zeitpunkt über

$$X_{i+1} = X_i + \tau \cdot (MX_i + b) \quad (9)$$

Ausgehend von der bekannten Startbedingung X_0 können wir dann iterativ X an allen Zeitpunkten t_i nähern. Die Qualität der Näherung hängt dabei entscheidend von der Schrittweite τ , und damit von der Feinheit der Auflösung des Zeitintervalls durch die t_i ab. Für zu große Schrittweiten wird die Näherung beliebig schlecht, für kleinere Schrittweiten immer exakter. Dabei ist nun auch interessant, wie schnell die Näherung bei Verkleinerung der Schrittweite sich an die tatsächliche Lösung annähert. Um dies zu beurteilen, bedient man sich der *Taylor-Entwicklung*. Wenn wir annehmen, dass die Funktion $X : [0, T] \rightarrow \mathbb{R}^6$ in einer Umgebung um $t_i \in [0, T]$ genügend oft nach t differenzierbar ist, dann lässt sie sich nach Taylor exakt darstellen durch die Reihe

$$\begin{aligned} X(t_i + \tau) &= X(t_i) + \frac{\tau}{1!} \cdot \frac{dX}{dt}(t_i) \\ &+ \frac{\tau^2}{2!} \cdot \frac{d^2X}{dt^2}(t_i) + \frac{\tau^3}{3!} \cdot \frac{d^3X}{dt^3}(t_i) + \dots \end{aligned} \quad (10)$$

Die Qualität eines numerischen Verfahrens kann daran gemessen werden, bis zu welcher Ordnung

```
% Abschussposition
p.x0=[0;0;0];
% Abschusswinkel
p.theta=20; % vertikal
p.phi=0; % horizontal
% Abschussgeschwindigkeit
p.vbkmh=95;
% Luftwiderstand
% CW Wert Ball
p.cw=0.45;
% Luftdichte
p.rho=1;
% Ballumfang
p.U=0.69;
% Magnus Effekt
% Rotationsachse
p.rot=[0;1;0];
% Magnuskoeffizient
p.cm=1;
% Drehgeschwindigkeit
p.w=-0;
% Schwerkraft
p.g=9.81;
% Ballmasse
p.m=0.437;
% Euler Params
p.nt=300;
p.tmax=3;
```

Abb. 2 Parameter erzeugen

diese Reihe durch das Verfahren korrekt dargestellt wird.

Ist X die Lösung von (2), dann wissen wir, dass

$$\frac{dX}{dt}(t_i) = MX(t_i) + b$$

gilt. Vergleichen wir dies mit (9), dann sehen wir, dass die Euleriteration gerade die ersten beiden Summanden der Taylor-Reihe abbildet. In einem Iterationsschritt macht man also einen Fehler, der sich wie τ^2 verhält. Wenn wir τ also beliebig klein wählen, so wird auch der Fehler in einer Iteration beliebig klein. Diese Eigenschaft bezeichnet man als Konsistenz des numerischen Verfahrens. Dies bedeutet jedoch nicht, dass der Fehler zwischen der Näherung und der exakten Lösung an einem beliebigen Punkt quadratisch von τ abhängt, denn in der Iteration addieren sich all diese quadratischen Fehler auf und man „verliert“ eine Fehlerordnung. Der Fehler verhält sich letztendlich wie τ . Aus diesem Grund unterscheidet man in der Fachliteratur (z. B. Deuffhard 2002)


```

function p=compParams(p)

% Umrechnen in m/s
p.vb=p.vbkmh/3.6;

% v0 aus Abschusswinkel
% und Geschwindigkeit berechnen
p.v0=...
[p.vb*cos(d2r(p.theta))...
*cos(d2r(p.phi));...
 p.vb*cos(d2r(p.theta))...
 *sin(d2r(p.phi));...
 p.vb*sin(d2r(p.theta))];

% Ballradius aus Umfang
p.r=p.U*0.5/pi;

% Querschnittsflaeche
p.A=p.r*p.r*pi;

% Rotationsvektor normieren
p.rot=p.rot/norm(p.rot);

% Schrittweite ausrechnen
p.dt=p.tmax/p.nt;
end

function y=d2r(x)
y=x*pi/180;
end

```

Abb. 3 Fehlende Parameter ausrechnen

zunächst den Fehler in einem Zeitschritt, den *Konsistenzfehler*, vom *globalen Fehler*, der durch die Differenz der exakten und der Näherungslösung am selben Punkt dargestellt wird. Praktisch bedeutet das, dass sich bei einem Euler-Verfahren eine Halbierung der Schrittweite auch in einer Halbierung des Abstandes zur exakten Lösung in einem bestimmten Zeitpunkt niederschlägt. Es gibt zahlreiche weitere Verfahren mit einer besseren *Fehlerordnung*, bei denen sich der Fehler wie z. B. τ^4 verhält (klassisches Runge-Kutta Verfahren, Deuflhard 2002). Diese stellen eben die Taylor-Reihe bis zu einer höheren Potenz in τ korrekt dar.

Wir wissen nun, dass das Euler-Verfahren für genügend kleine Schrittweiten eine exakte Lösung von (2) approximiert. Da aufgrund der Gestalt von (2) sichergestellt ist, dass eine eindeutige exakte Lösung auch existiert, können wir davon ausgehen, dass dies auch die Lösung sein muss, an die wir uns mit dem Euler-Verfahren annähern.

```

function [X,V]=eulerM(p)
% Initialbedingung setzen
Xr=[p.x0;p.v0];
% Arrays zum Speichern der
% Ergebnisse
X=Xr(1:3);
V=Xr(4:6);
% Matriceintraege ausrechnen
CW=-p.rho*p.A*p.cw*0.5/p.m;
CM=-p.rho*p.A*p.cm*p.w*0.5/p.m;
% Teilmatrix
A=[...
 CW,-CM*p.rot(3),CM*p.rot(2);...
 CM*p.rot(3),CW,-CM*p.rot(1);...
 -CM*p.rot(2),CM*p.rot(1),CW];
% Systemmatrix
M=[zeros(3),eye(3);zeros(3),A];
% Gravitation
b=[0,0,0,0,0,-p.g]';
% So lange rechnen, bis Ball
% wieder auf dem Boden landet
while Xr(3)>=0
    % Iterationsvorschrift
    Xr=Xr+p.dt*(M*Xr+b);
    % Punkte abspeichern
    X=[X,Xr(1:3)];
    V=[V,Xr(4:6)];
end
end

```

Abb. 4 Funktion für Euler-Verfahren

Für ein gegebenes numerisch zu lösendes Problem müssen diese beiden Komponenten immer zusammen vorhanden sein, da sonst keine Aussagen über Korrektheit und Genauigkeit der numerischen Lösung möglich sind.

Im vorliegenden Fall können wir nun die Flugbahn eines Fußballes mit den hier diskutierten Effekten vorhersagen. Die Umsetzung des Euler-Verfahrens kann dann in einer beliebigen Programmiersprache (z. B. Python) erfolgen. Kurz diskutiert werden soll hier eine Umsetzung in MATLAB, da diese besonders kurz ausfällt und Matrizen und Vektoren, sowie die zugehörige Arithmetik natürlich damit umsetzbar sind. Der vorliegende Quelltext ist außerdem kompatibel zur freien Software *Octave* und läuft damit ohne jede Einschränkung.

Zunächst werden in einem Script alle notwendigen Parameter eingegeben (Abb. 2). Damit diese gebündelt an eine Funktion übergeben werden können, werden sie in der Struktur *p* zusammen-

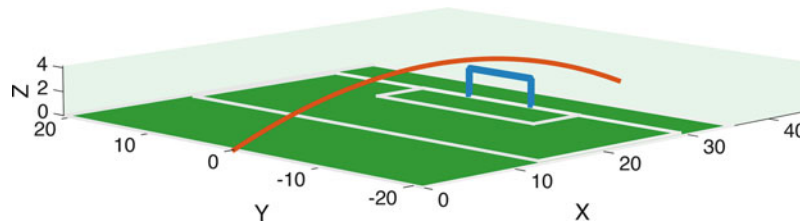


Abb. 5 Parameter aus 2. Ohne Eigenrotation geht der Ball über das Tor hinweg

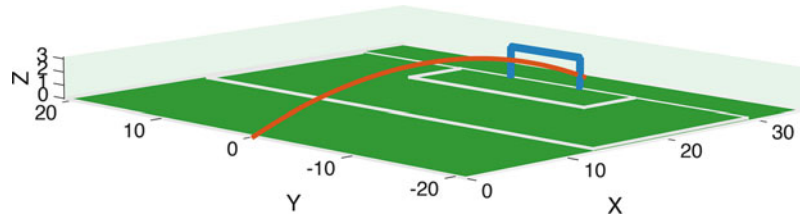


Abb. 6 Parameter aus 2, wobei die Drehgeschwindigkeit $p.w$ auf -3 geändert wurde. Aufgrund des Magnuseffektes geht der Ball mit sonst gleichen Anfangsdaten ins Tor

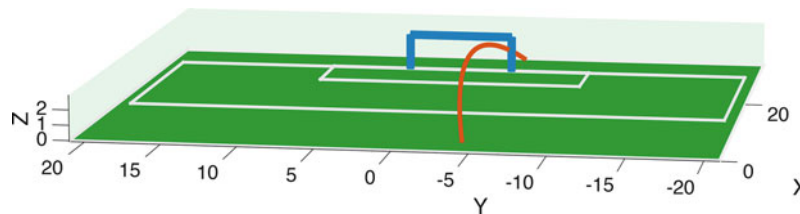


Abb. 7 Ändert man Abschussposition und -winkel, sowie die Drehachse, lassen sich auch kunstvollere Schüsse simulieren

gefasst. Diese stellt im Prinzip einen Container für mehrere Variablen unterschiedlicher Formen dar. Auf ihre Einträge (*Felder*) kann, wie in klassischen Programmiersprachen wie C, mit dem Punkt-Operator `.` zugegriffen werden. Abb. 3 zeigt eine Hilfsfunktion, mit der aus den gegebenen Parametern vor der eigentlichen Berechnung weitere Parameter vorberechnet werden.

Die so gewonnene Struktur `p` enthält nun alles, was für die Berechnung der Flugkurve gebraucht wird, sodass wir sie an die Funktion aus Abb. 4 übergeben können. Dort werden die Variablen aus (6), (7) und (8) besetzt und die Iteration (9) gerechnet. Nach dem Ausführen des Scripts aus Abb. 2 muss man dazu nur folgende Befehle verwenden:

```
% Fehlende Parameter berechnen
p=compParams(p);
% Euler Verfahren
[X,V]=eulerM(p);
```

Die Matrizen `X`, `V` enthalten danach die Punktkoordinaten bzw. Geschwindigkeitsvektoren an den diskreten Zeitpunkten als Spalten. Ihr Inhalt kann dann mit beliebigen Mitteln visualisiert werden (MATLAB: `plot3()`). Eine etwas aufwändigere Darstellung in MATLAB ist in Abb. 5 zu sehen. Wird bei gleichen Parametern der Ball in Rotation versetzt, so wird er dank Magnus-Effekt im Tor ankommen (s. Abb. 6). Ansonsten sind der Kreativität nun keine Grenzen mehr gesetzt, wenn man Abschussposition, -geschwindigkeit und Drehachse anders wählt (s. Abb. 7). Interessant kann auch das Variieren der eigentlich festen Parameter des Luftwiderstandes sein (c_w , Umfang, Ballmasse usw.).

4 Umsetzung mit Tabellenkalkulation

Die Umsetzung mit Tabellenkalkulationssoftware wie *Microsoft Excel* oder einer der freien Alternativen wie *LibreOffice* ist grundsätzlich möglich. Dabei könnte man so vorgehen, dass auf einem Tabellenblatt alle Parameter aus Abb. 2 in Zellen abgelegt werden. Auf einem weiteren Tabellenblatt werden alle Formeln aus Abb. 3 in Zellen hinterlegt, wobei die Variablenbezeichner durch Referenzen auf die entsprechenden vorher angelegten Parameterzellen zu ersetzen sind. Auf einem dritten Tabellenblatt könnte man dann die Euler-Iteration zeilenweise durchführen: Da es in der Tabellenkalkulation keine Strukturen für Vektoren, Matrizen und deren Arithmetik gibt, muss dies über Umwege realisiert werden: Die jeweils 3 Komponenten für Position und Geschwindigkeit können je einer Spalte zugeordnet werden. Die Matrixmultiplikationen und Vektoradditionen aus (9) sollten zuvor von Hand oder mit einem geeigneten Computeralgebrasystem ausgerechnet werden, damit man sieht, wie sich jede Komponente aus dem vorherigen Schritt ergibt. In diesen komponentenweisen Iterationsvorschriften sind dann alle Variablen durch Referenzen auf die vorherige Iterationszeile zu ersetzen. Die Iteration für jede Komponente von Ort und Geschwindigkeit kann dann als Formel in der entsprechenden Spalte hinterlegt werden. Hat man dies einmalig getan, so lässt sich die Formel durch klicken und Ziehen an der unteren rechten Ecke der Zelle auf die darunterliegenden Zellen ausdehnen, sodass jede Zeile einem Iterationsschritt entspricht. Es ist zu beachten, dass Zellreferenzen dabei zunächst als relativ angesehen werden. Das bedeutet, dass ein Bezug auf die vorherige Zeile immer ein Bezug auf die vorherige Zeile bleiben wird. Das gilt ohne Modifikation auch für Referenzen auf eigentlich feste Zellen wie die Parameterzellen. Jedoch lassen sich relative Bezüge durch Einfügen zweier \$ (Dollar-Zeichen), je eines vor dem Spaltenbezeichner und eines vor dem Zeilenbezeichner, in absolute Bezüge umwandeln. Ein relativer Bezug auf Zelle B2 kann z.B. durch \$B\$2 als absoluter Bezug geschrieben werden. Das Einsetzen der Formeln

muss mit größter Sorgfalt geschehen, da in der Standardansicht nur die berechneten Zahlen in den Zellen angezeigt werden. Eine etwaige Fehlersuche gestaltet sich dementsprechend schwierig.

Die Gesamtzahl der Iterationen ist in diesem Modus durch die Anzahl der Zeilen eines Tabellenblattes begrenzt. Je nach Feinheit des Zeitschrittes ist damit auch die Maximalzeit t_n begrenzt, was je nach Wahl der Parameter mehr oder weniger gravierend sein kann. Eine direkte Visualisierung der dreidimensionalen Flugkurve ist nicht möglich. Hier bietet sich der Export als csv-Datei und die Visualisierung mit z. B. *gnuplot* an.

5 Umsetzung, Probleme auf Schülerseite

Die Problemstellung bietet sich ab Klassenstufe 11 an, da hier sowohl die Bewegungsarten aus der Physik, sowie Differentialrechnung vorhanden sind. Um die Flugkurve zu modellieren, bietet sich eine Videoanalyse eines Freistoßes an. Hier kann man entweder auf Videos aus dem Internet zurückgreifen, oder selbst Videos erstellen. Letzteres hängt sicherlich davon ab, wieviel Zeit man für die Aufbereitung der Daten und die Datenerhebung aufbringen will. Die Analyse von Videos ist mit gängiger Videoanalyse-Software gut möglich. Ein Problem hierbei kann die exakte Vermessung des Abschusswinkels und der Abschussgeschwindigkeit, sowie der Längen sein. Hat man das Video nicht selbst erzeugt, so kann die Einbringung eines Maßstabes sehr schwierig sein. Oft muss man diesen anhand von Spielergrößen oder dem Wissen über die Längen der eingezeichneten Linien auf einem Fußballfeld errechnen. Die Messung der Zeit ist allerdings durch die Zeitangabe im Video genau genug gegeben. Über beide Größen lässt sich nun die Geschwindigkeit bestimmen. Gegebenenfalls ist es sinnvoll, neben der Berechnung der Größen auch eine Fehlerfortpflanzungsrechnung durchzuführen, um die Güte der Messungen abzuschätzen.

Schwieriger als die Messung der Geschwindigkeit ist die Abschätzung der Drehgeschwindigkeit und der Drehachse beim Magnuseffekt. Hier ist

es sinnvoll Werte zu schätzen. So kann man die Umdrehungen pro Sekunde aus den Videos durch zählen ermitteln. Der Effekt des Variierens der Drehachse hingegen kann durch Ausprobieren am Rechner beobachtet werden.

Die Modellierung als schiefer Wurf ist von Schülerseite oft sehr klar. Problematischer sind vielmehr die Modellerweiterungen, die im regulären Schulunterricht nicht behandelt werden, wie Luftwiderstand und Magnuseffekt. Diese werden oft nicht als Kräfte erkannt, bzw. fällt es den Schülerinnen und Schülern schwer, die Wirkung der Kraft auf die Bewegung, d. h. die Einbeziehung der resultierenden Beschleunigung in die Bewegungsgleichung zu beschreiben. Hier kann es sinnvoll sein, diesen Zusammenhang noch einmal ins Gedächtnis zu rufen.

Die Einbeziehung des Rechners von Anfang an ist für die Visualisierung sehr hilfreich. Dabei kann anstatt der Implementierung des Euler-Verfahrens in MATLAB oder einer vergleichbaren Script- (Python, Octave, FreeMat) oder Programmiersprache (C, Java) auch mit einem Tabellenkalkulationsprogramm gearbeitet werden. Für etwaige Änderungen von Parametern, sowie der Visualisierung der Trajektorien bietet sich MATLAB, Octave oder FreeMat allerdings an, da hier 3D-Diagramme einfach zu erstellen sind und man durch die 3D-Ansicht ein gutes Verständnis für die Flugbahnen bekommt. Weiterhin können Darstellungen für Tore oder Mauer leichter erzeugt werden.

Es hat sich gezeigt, dass die Schülerinnen und Schüler es als einfacher empfinden, wenn die Differentialgleichung als Differenzengleichung vorliegt. So zeigen sich oft Schwierigkeiten im Verständnis des Zusammenhangs einer Größe und ihrer Änderung zum selben Zeitpunkt. Ein Übergang von Differentialquotienten via Differenzenquotienten kann hier eine anschauliche Brücke zur Annahme einer konstanten Geschwindigkeit in einem Diskretisierungsschritt liefern. Sobald verstanden ist, dass man auf kleinen Zeitintervallen Größen konstant lassen kann, ist die Einbeziehung

des Magnuseffekts als solche keine große Hürde mehr.

Vielmehr birgt die Umrechnung der Winkelanteile der einzelnen Größen, gerade im 3D-Fall, große Probleme mit sich. Hierbei ist die angesprochene Visualisierung ein großer Vorteil. Man sollte die Schülergruppen auch dazu anhalten, Skizzen zu erstellen. Gerade, wenn die Drehachse des Balls ins Spiel kommt, müssen viele verschiedene Systeme beachtet werden. Da dies im Normalfall nicht in der Schule behandelt wird, können hierbei Probleme auftreten.

6 Fazit und Erweiterungen

Das Projekt ist für Schülerinnen und Schüler ab Klassenstufe 11 gut beherrschbar. Das Euler-Verfahren liefert dabei eine einfache aber leistungsstarke Methode zur numerischen Lösung von gewöhnlichen Differentialgleichungen. Hierdurch wird der Zugang zu physikalisch schwierigeren Phänomenen wie dem Magnus-Effekt und dem Luftwiderstand möglich. Alle betreuten Schülergruppen konnten das mathematische Modell und die zugehörige Programmierung gut umsetzen.

Eine Erweiterung zu dem beschriebenen Modell ist die Einbeziehung der Bewegung der Mauer. So kann aufbauend auf statistischen Daten etwa die Sprunghöhe und Sprungwahrscheinlichkeit ermittelt werden. Zusammen mit einem Stärkemodell für den Torwart können somit für verschiedene Szenarien optimale Parameter für einen Schuss, der zum Torerfolg führt, ermittelt werden.

Auch ein Modell für das Können des Schützen kann dabei von Vorteil sein. So kann man etwa optimale Parameter berechnen, wenn man weiß, dass der Schütze nicht sehr schnell bzw. mit wenig Effekt schießen kann. Die Platzierung des Freistoßes ist allerdings selbst ein eigenständiges Thema, bei dem viele verschiedene Faktoren berücksichtigt werden müssen.

Der Abstoß im Fußball wurde in einem ähnlichen Kontext in Maas (2009) behandelt.

Literatur

- Deuflhard, P., Bornemann, F.: Scientific Computing with Ordinary Differential Equations. Springer (2002)
- Maaß, J., Schlöglhofer, F.: Der Abstoß beim Fußball. ISTRON Band **13**, 125–143 (2009)
- Wirsching, G.: Gewöhnliche Differentialgleichungen. Springer (2006)

Neue Materialien für einen realitätsbezogenen
Mathematikunterricht 3

ISTRON-Schriftenreihe

Humenberger, H.; Bracke, M. (Hrsg.)

2017, XII, 190 S. 143 Abb., Softcover

ISBN: 978-3-658-11901-0