# Ausarbeitung Übung 1

Hannes Höttinger und Josua Kucher

*FH Technikum Wien, Game Engineering und Simulation, Wien, AUT*

## Bsp4

Ein Ball wird von der Stelle (x, y) = (0, 0) unter dem Winkel $\alpha \in (0, \frac{\pi}{2})$ nach rechts oben geworfen, mit Startgeschwindigkeit $v > 0$ (gemessen in m/s) zum Zeitpunkt t = 0. Dann folgt er unter dem Einfluss der Gravitation (mit Erdbeschleunigung $g = 9.81m/s$) einer Bahnkurve. Da in x-Richtung keine Beschleunigung wirkt, gilt für die horizontale Komponente die Differentialgleichung:

$$\ddot{x}(t) = 0$$

für alle t. In vertikaler Richtung wirkt nach unten die Erdbeschleunigung g, und daher gilt:

$$\ddot{y}(t) = -g$$

für alle t.

*a) Zu welchem Zeitpunkt t fällt der Ball wieder zu Boden?*

Folgende Ausgangssituation ist gegeben: x = 0 und y = 0 und die aus Bsp3 abgeleitete Funktion für $y(t) = -\frac{g}{2} \cdot t^2 + v_{0y} \cdot t + y_0$. Der Ball ist am Boden wenn seine $y$ Komponente zum Zeitpunkt $t \rightarrow 0$ wird. Somit kann die Gleichung auf 0 gesetzt werden und nach $t$ umgeformt werden ($y_0 = 0$. Es ergibt sich folgendes Ergebnis für den Aufprallpunkt (wobei $v_{0y} = v_0 \cdot \sin\alpha$):

$$0 = -\frac{g}{2} \cdot t^2 + v_{0y} \cdot t \tag{1}$$

$$t = \frac{2 \cdot v_{0y}}{g} = \frac{2 \cdot v_0 \cdot \sin\alpha}{g} \tag{2}$$

Es wurde ein $v_0 = 30m/s$ gewählt mit einem Abwurfwinkel von 45°. Die Parameter werden in die Gleichung eingesetzt und die Aufprallzeit von **t = 4.325s** berechnet.

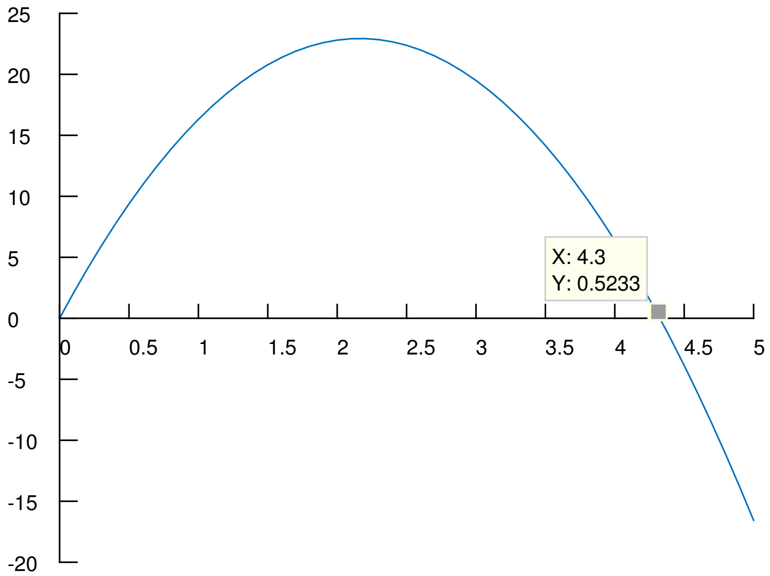



**Abbildung 1:** y(t) mit Aufprallzeit t = 4.325 auf x-Achse

*b) Der Ball springt wieder hoch (elastischer Stoß, Reflexionsgesetz) und fliegt weiter. Beschreiben Sie, wie die Lösungskurve fortzusetzen ist.*

Die weitere Flugbahn des Balles ist nur durch den Aufprallzeitpunkt verschoben. Die Abwurfgeschwindigkeit in x-Richtung bleibt konstant erhalten. Dies wird in Beispiel c) visualisiert. Die Gleichungen bleiben exakt bestehen, wir verschieben lediglich die Kurve vom Startzeitpunkt (0,0) an die Position des Aufprallpunktes. In der Abb. 1 wird $y(t)$ mit der berechneten Aufprallzeit gezeigt. Die Kurve würde somit auf diese Position gesetzt werden und den gleichen Funktionsverlauf haben (vgl. c).

*c) Analog wie unter b), aber mit Bremswirkung: Bei jedem Aufprall auf den Boden reduziert sich die momentane Geschwindigkeit durch einen Reibungsverlust um b Prozent. Wählen Sie konkrete Daten für die Parameter $\alpha$; g; v; b und produzieren Sie eine Animation der resultierenden Bahnkurve.*

In Matlab gibt es die Funktion *animatedline*, welche für die Animation der Bahnkurve verwendet wurde. Es werden folgende Anfangsparameter definiert:

$v = 30$; $\alpha = \frac{\pi}{4}$; $g = 9.81$ und $b =$coeff$= 0.90$

Für die Animation wird $y(x)$ berechnet. Dazu wird $t = \frac{x}{v_{0x}}$ in die Gleichung von $y(t)$ eingesetzt und erhalten:

$$y(x) = -\frac{g}{2} \cdot (\frac{x}{v_{0x}})^2 + v_{0y} \cdot (\frac{x}{v_{0x}}) + y_0 \tag{3}$$

Es wird nun für jeden x-Wert die aktuelle y-Position berechnet und in einem animierten Plot ausgegeben. Bei jedem Aufprall wird die y-Komponente der Geschwindigkeit mit dem Faktor 0.90 multipliziert und die x-Position zurückgesetzt (mit der Hilfsvariable $tt$), jedoch an der richtigen Position gezeichnet.

**Listing 1:** Animierter Plot für die Flugbahn eines schrägen Wurfs.

```
% animated plot
v = 30;
alpha = pi/4;
g=9.81;
% Reibungsverlust
coeff = 0.90;

figure;
h = animatedline;
axis([0,1000,0,25])

x = 0:0.5:1000;
vy = v*sin(alpha);
vx = v*cos(alpha);
tt = 0;
for k = 1:length(x)
    y = -(1/2)*g.*((x(k)-tt)/(vx)).^2 + vy.*((x(k)-tt)/(vx)↩
        );
    if (y < 0)
        % save x(k) to set back x coordinate
        tt = x(k);
        % slow down
        vy = vy*coeff;
    end
    % add current location point to the plot
    addpoints(h,x(k),y);
    drawnow
end
```

Ergibt folgenden Kurvenverlauf:

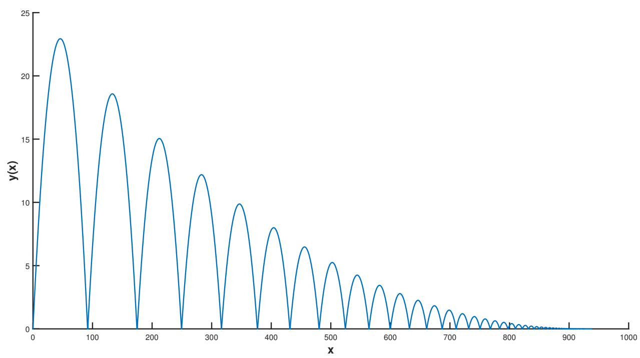
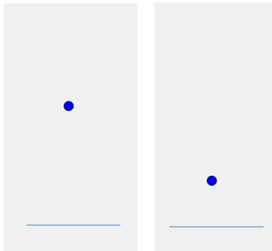



**Abbildung 2:** Kurvenverlauf von y(x) mit Reibungsverlust

Weiters wurde eine Animation erstellt, die einen lotrechten Wurf ohne x-Komponente beschreibt. Diese Animation zeigt ein besseres physikalisches Verhalten, da der Faktor dt mitberechnet wird. Abb. 3 zeigt einen Ausschnitt der Animation.



(a) (b)

**Abbildung 3:** Lotrechter Wurf mit Reibungsverlust auf Gerade

**Listing 2:** Animierter Ball

```
% animated ball
initpos=5;
initvel=0;
r_ball=0.25;
gravity=9.81;
coeff=0.90;
dt=0.0125;      %timestep

line([-5*r_ball,5*r_ball],...
     [0,0]);

pos=initpos-r_ball;
vel=initvel;

axis([-5*r_ball,5*r_ball,0,initpos+2*r_ball]);
    axis('equal');
    axis('off');
while 1
    t_loopstart=tic();
    pos=pos+(vel*dt) - 1/2 * gravity * (dt^2);
    vel=vel-(gravity*dt);
    if pos<0
        vel=-vel*coeff;      %current vertical velocity
    end

    ball=rectangle('Position',[-r_ball,pos,r_ball,↩
       r_ball],...
              'Curvature',[1,1],...
              'FaceColor','b');
    %Pause
    el_time=toc(t_loopstart);
    pause(dt-el_time);
    delete(ball)
end
```