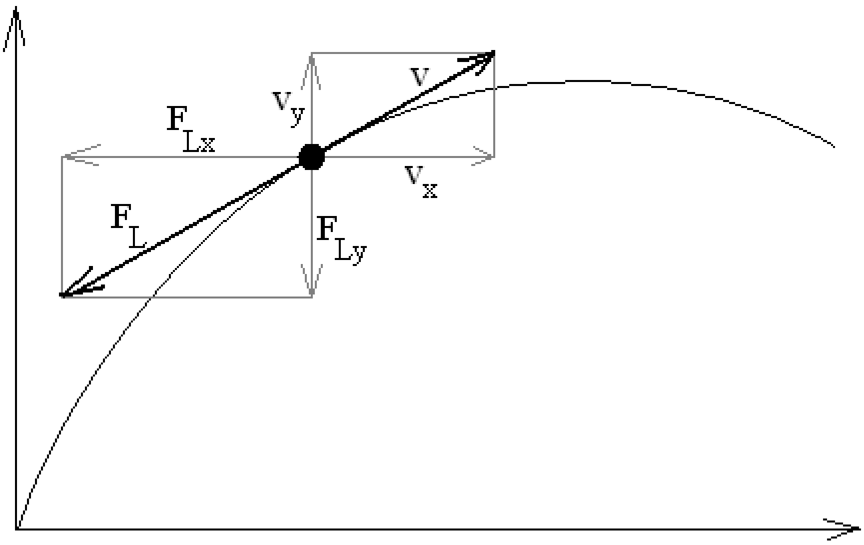
Flug einer Kugel mit Berücksichtigung des Luftwiderstands – eine Simulation

Die Simulation beruht auf einem bewussten, prinzipiellen Fehler: Man stellt sich die Flugbahn als einen Polygonzug (= Folge geradliniger Streckenabschnitte) vor. Die Kugel fliegt jeweils für eine Zeiteinheit Δt geradeaus und führt am Ende dieses Geradenstücks eine ruckartige Richtungsänderung durch. Die Richtungsänderung wird in jedem Schleifendurchgang einer entsprechenden Computer-Prozedur neu berechnet. Falls man den Zeitabschnitt Δt klein genug wählt, kann man von einer hinreichend guten Näherung an den tatsächlichen Kurvenverlauf ausgehen. Die Güte der Näherung kann nur mit Hilfe eines Experiments getestet werden, weil keine Formel für die Darstellung der Flugbahn zur Verfügung steht. Die Flugbahn einer Kugel kann als mathematische Formel nur für den Fall hergeleitet werden, dass der Einfluss der Luftreibung ausgeschlossen wird.

In der Skizze geht man von einer Momentaufnahme mit der Momentangeschwindigkeit v aus. Man stellt sich die Geschwindigkeit aufgespalten in die Komponenten v_x und v_y vor. (Man bestimmt umgekehrt v aus den Komponenten v_x und v_y mit dem Satz des Pythagoras!)

Die Bewegung der Kugel in Richtung von v hat die Luftreibungs-Bremskraft F_L zur Folge ($F_L = \frac{1}{2} \cdot c_W \cdot \rho \cdot A \cdot v^2 = C \cdot v^2$). Auch die Bremskraft denken wir uns in die Komponenten F_{Lx} und F_{Ly} zerlegt. Die beiden Kräfteparallelogramme (bzw. Rechtecke) sind zueinander ähnlich.

Daher gilt $\dfrac{F_{Lx}}{F_L} = \dfrac{v_x}{v} \iff F_{Lx} = \dfrac{v_x}{v} \cdot F_L = \dfrac{v_x}{v} \cdot C \cdot v^2 \iff F_{Lx} = v_x \cdot C \cdot v$



Indem wir die Entwicklung von v_x bzw. von v_y getrennt untersuchen, ersparen wir uns die Verwaltung des sich ändernden Winkels zwischen der Geschwindigkeit v und der Horizontalen. Dieser Winkel taucht in den Berechnungen überhaupt nicht auf.
Allerdings müssen wir nach jeder neuen Berechnung von v_x und v_y die Gesamtgeschwindigkeit v mit Pythagoras neu berechnen.

Die Bremskraft F_{Lx} in Richtung der x-Achse hat nun eine Bremsbeschleunigung in x-Richtung **entgegen** v_x zur Folge: $a_x = \dfrac{F_{Lx}}{m} = v_x \cdot C_1 \cdot v \quad (mit\ \ C_1 = \dfrac{C}{m})$. Die Geschwindigkeit in x-Richtung wird also im nächsten Zeitintervall kleiner, und zwar sinkt sie auf

$$v_x^{(i+1)} = v_x^{(i)} - a_x^{(i)} \cdot \Delta t = v_x^{(i)} - v_x^{(i)} \cdot C_1 \cdot v^{(i)} \cdot \Delta t = v_x^{(i)} \cdot (1 - C_1 \cdot v^{(i)} \cdot \Delta t) \ .$$

Ganz analog folgt derselbe Term für die Änderung der Geschwindigkeit in y-Richtung, wobei hier noch der Effekt der Erdbeschleunigung (stets nach unten, d.h. in negative Richtung) berücksichtigt werden muss:

$$v_y^{(i+1)} = v_y^{(i)} - a_y^{(i)} \cdot \Delta t - g \cdot \Delta t = v_y^{(i)} - v_y^{(i)} \cdot C_1 \cdot v^{(i)} \cdot \Delta t - g \cdot \Delta t, \quad \text{also}$$

$$v_y^{(i+1)} = v_y^{(i)} \cdot (1 - C_1 \cdot v^{(i)} \cdot \Delta t) - g \cdot \Delta t$$

Man überzeuge sich, dass bei $v_{y\ neu}$ der Bremsbeschleunigungsanteil mal subtrahiert und mal addiert wird, je nach Vorzeichen (= Richtung) von v_y.

Im Folgenden wird die Simulation in MuPAD programmiert. Auch Ungeübten sollte die Interpretation der einzelnen Textzeilen unschwer gelingen.

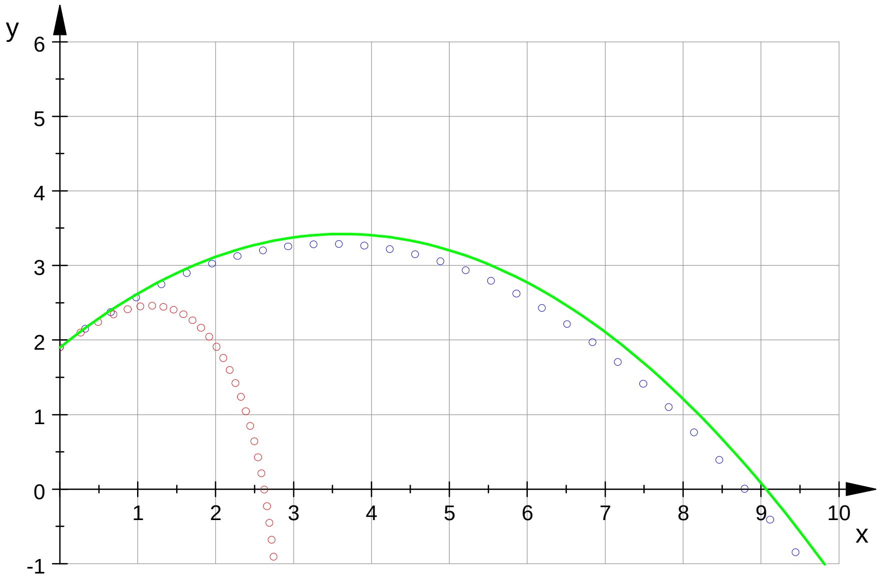
- ```
  m:=0.005:;  //Masse in kg
  w_beta:=40:; //Abwurf-Winkel in Grad
  v0:=8.5:;  // Abwurf-Geschwindigkeit in m/s
  d:=0.1:;  // Durchmesser der Kugel in m
  y0:=1.9:;  // Abwurfhöhe in m
  dt:=0.05:; // Dauer einer Zeiteinheit in s
  n:=50:;  // Anzahl der Zeiteinheiten

  g:=9.81:; // Erdbeschleunigung
  ro:=1.225:;  //  Dichte der Luft in kg/m³
  cw:=0.45:;  // Luftwiderstansbeiwert
  A:=PI*d^2/4:; // Querschnittsfläche in m²
  C:=0.5*cw*ro*A:;  // Konstante (s. Text oben)
  C1:=float(C/m):;  //     --- " ---
  vx0:=float(v0*cos(w_beta*PI/180)):; // Anfangskomponenten
  vy0:=float(v0*sin(w_beta*PI/180)):; // der Geschwindigkeit
  ```
- ```
  Bahn_ohne_Luftwid:=proc(n)
  local vx,vy;
  begin
  sx[0]:=0;sy[0]:=h;
  vx:=vx0;vy:=vy0;
  for i from 0 to n do
    vx:=vx;
    vy:=vy-g*dt;
    sx[i+1]:=sx[i]+vx*dt;
    sy[i+1]:=sy[i]+vy*dt;
  end_for;
  end_proc:
  ```
- ```
  Bahn_mit_Luftwid:=proc(n)
  local vx,vy,v;
  begin
  smx[0]:=0;smy[0]:=h;
  vx:=vx0;vy:=vy0;
  for i from 0 to n do
    v:=float(sqrt(vx^2+vy^2)); // s. Text oben im Kasten
    vx:=vx*(1- C1*v*dt); //
    vy:=vy*(1- C1*v*dt)-g*dt;
    smx[i+1]:=smx[i]+vx*dt;
    smy[i+1]:=smy[i]+vy*dt;
  end_for;
  end_proc:
  ```
- ```
  Bahn_mit_Luftwid(n):;
  Bahn_ohne_Luftwid(n):;
  ```
- ```
  Punkte_mit:=plot::PointList2d([[smx[i],smy[i]] $ i=0..n+1],PointSize=1,
  PointStyle=Circles,Color=RGB::Red):;
  Punkte_ohne:=plot::PointList2d([[sx[i],sy[i]] $ i=0..n+1],PointSize=1,
  PointStyle=Circles,Color=RGB::Blue):;
  ```
- ```
  x:=t-->vx0*t:;//mathematisch ohne Luftwid.
  y:=t-->vy0*t-0.5*g*t^2+y0:;
  Kurve_o:=plot::Curve2d([x(t),y(t)] ,t=0..n*dt,PointSize=1,
           PointStyle=Circles,Color=RGB::Green):;
  ```
- ```
  plot(Punkte_ohne,Punkte_mit,Kurve_o,
       ViewingBox = [0..10, -1..6], Scaling=Constrained,
       GridVisible = TRUE);
  ```
  ```

Zwei Simulationsbeispiele zum Vergleich.
Im zweiten Beispiel wird eine schwerere Kugel benutzt und die Zeiteinheit verkleinert.
Man sieht, wie die Simulation für den Fall „ohne Luftwiderstand“ näher an der theoretischen Kurve liegt. Würde man die Zeiteinheit noch weiter verkleinern, bekäme man die beiden Kurven im Rahmen der Zeichengenauigkeit zur Deckung.

Beispiel 1:

```
m:=0.005:;  //Masse in kg
w_beta:=40:; //Abwurf-Winkel in Grad
v0:=8.5:;  // Abwurf-Geschwindigkeit in m/s
d:=0.1:;  // Durchmesser der Kugel in m
y0:=1.9:;  // Abwurfhöhe in m
dt:=0.05:; // Dauer einer Zeiteinheit in s
n:=50:;  // Anzahl der Zeiteinheiten
```



Beispiel 2:

```
m:=0.05:;  //Masse in kg
w_beta:=60:; //Abwurf-Winkel in Grad
v0:=8.5:;  // Abwurf-Geschwindigkeit in m/s
d:=0.1:;  // Durchmesser der Kugel in m
y0:=1.9:;  // Abwurfhöhe in m
dt:=0.02:; // Dauer einer Zeiteinheit in s
n:=100:;  // Anzahl der Zeiteinheiten
```