Infoprojekt - Schräger Wurf

Lukas Kaufmann
Christina Meinert

Grundlegende Gleichung

Die Bahnkurve des schiefen Wurfes ohne Reibung ist parametrisiert durch

$$y(x) = x \tan(\theta) - \frac{g}{2 \cdot v^2 \cdot \cos^2(\theta)} x^2 \tag{1}$$

- θ Abwurfwinkel
- v Abwurfgeschwindigkeit

Grundlegende Gleichungen II

Der schiefe Wurf mit Reibung (*Stokesche Reibung*) folgt den beiden Differentialgleichungen

$$\ddot{x}(t) + \frac{k}{m}\dot{x}(t) = 0 \tag{2}$$

und

$$\ddot{y}(t) + \frac{k}{m}\dot{y}(t) + g = 0 \tag{3}$$

$k = 6\pi\eta r$

$\Rightarrow$ Bewegungsgleichungen sind entkoppelt!

Grundlegende Gleichungen III

Mit den Randbedingungen $\dot{x}(0) = v_x, x(0) = 0, \dot{y}(0) = v_y$ und $y(0) = 0$ erhält man die Lösungen

$$x(t) = \frac{m \cdot v_x}{k} \left(1 - e^{-\frac{k}{m}t}\right)$$

und

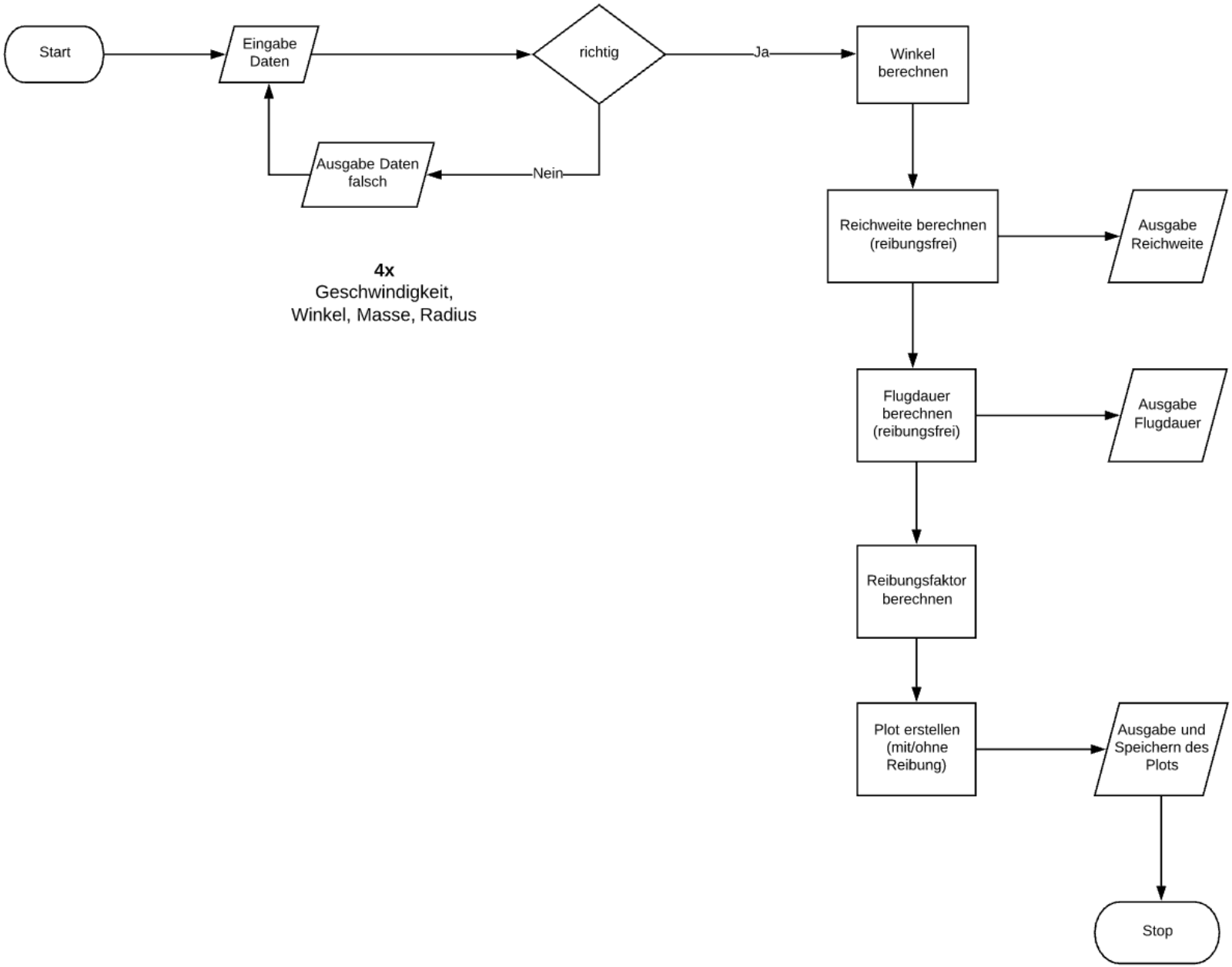
$$y(t) = -\frac{mg}{k}t + \left(v_y + \frac{mg}{k}\right)\left(1 - e^{-\frac{k}{m}t}\right)$$

Damit ergibt sich

$$y(x) = \frac{m^2 g}{k^2} + \ln\left[1 - \frac{k}{m \cdot v_x} \cdot x\right] + \left(v_y + \frac{mg}{k}\right)\frac{x}{v_x}$$

als Flugbahn des Wurfes unter Einwirkung der Reibung.

Flussdiagramm



Code I

```
1  int main rootWurf () {
2
3    double v, w, wRad, m, r;
4
5    ///Eingabe der erforderlichen Daten
6    cout << "Gib eine Abwurfgeschwindigkeit [m/s] ein:  " << endl;
7    cin >> v;
8    while (v <= 0) {
9      cout << "Du wirfst deine Kugel nach hinten. Hast du vorher geschaut, ob da jemand
         steht? Bitte korrigiere deine Eingabe der Geschwindigkeit!" << endl;
10     cin >> v;
11   }
```

mainWurf.cpp

Code II

```
1  #include "statischeDaten.hpp"
2  #include "math.h"
3
4  ///Konstruktor definieren:
5  statischeDaten::statischeDaten(){
6  }
7
8  ///Destruktor definieren:
9  statischeDaten::~statischeDaten(){
10 }
11
12 ///Wurfweite berechnen:
13 double statischeDaten::reichweite(double v, double w){
14         return ( (v*v)/9.81 * sin(2*w) );
15 }
16
17 ///Flugdauer berechnen:
18 double statischeDaten::flugdauer(double v, double w){
19         return (2/9.81 * v * sin(w));
```

statischeDaten.cpp

Code III

```
1   ///Berechnung der Reichweite des reibungsfreien Wurfes
2   statischeDaten d ;
3   cout << "Reichweite des Wurfes (ohne Reibung):  ";
4   cout << d.reichweite(v, wRad) << endl;
5
6   ///Berechnung der Flugdauer des reibungsfreien Wurfes
7   statischeDaten t;
8   cout << "Dauer des Wurfes(ohne Reibung):  ";
9   cout << t.flugdauer(v, wRad) << endl;
10
11  ///Plotten der Bahnkurve mit den eingegebenen Parametern
12  bahnkurve s;
13  s.plot(v, wRad, m, r);
```

mainWurf.cpp

Code IV

```
1  TApplication *a1 = new TApplication("a1", 0, 0);
2  TCanvas *c1 = new TCanvas("c1", "Bahnkurven", 750, 500);
3
4  TF1 *f1 = new TF1("f1", "x * tan([1]) - [2]/(2*[0]*[0]*cos([1])*cos([1])) *x*x", 0., d);
5  f1->SetParameter(0, v); f1->SetParameter(1, w); f1->SetParameter(2, g);
6
7  double k = (6 * PI * 17.1 * 0.000001 * r/m);
8
9  TF1 *f2 = new TF1("f2","[2]/([3]*[3])*log(1 - [3]/[0] *x)+([1] + [2]/[3])*x/[0]", 0., d);
10 f2->SetParameter(0, vx); f2->SetParameter(1, vy);
11 f2->SetParameter(2, g); f2->SetParameter(3, k);
12 f2->SetLineColor(4);
13 f1->SetTitle("Bahnkurven");
14 f1->Draw("c1");
15 f2->Draw("same");
16
17 auto legend = new TLegend(0.1, 0.7, 0.48, 0.9);
18 legend->SetHeader("LEGENDE", "C");
19 legend->AddEntry(f1, "Bahnkurve ohne Reibung", "L");
20 legend->AddEntry(f2, "Bahnkurve mit Reibung", "L");
21 legend->Draw("same");
22
23 c1->Update();
24 c1->Draw();
25 c1->SaveAs("plot.png");
26
27 a1->Run();
```

bahnkurve.cpp

Plot

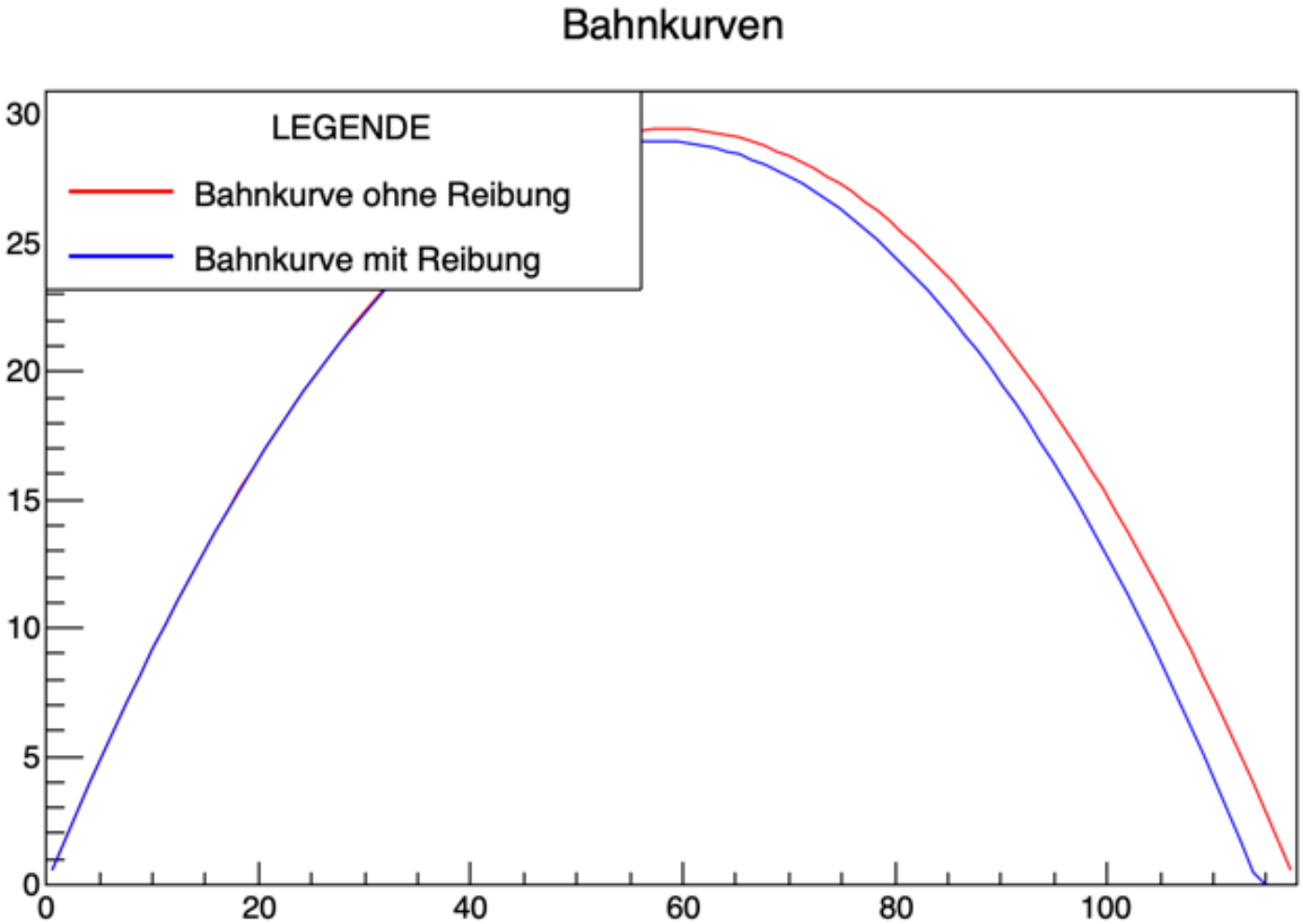


Figure: v=34m/s, w=45Grad, m=0.01kg, r=0.3m

Fazit

- (Eigenständiges) Lösen von Programmierfehlern
- teamorientierte Arbeitsweise beim Coden (z.B. Umgang mit GitHub)
- Einblick in die Verwendung von Programmen in der Physik
  (Was ist möglich/sinnvoll?)
- Probleme über das reine Programmieren hinaus