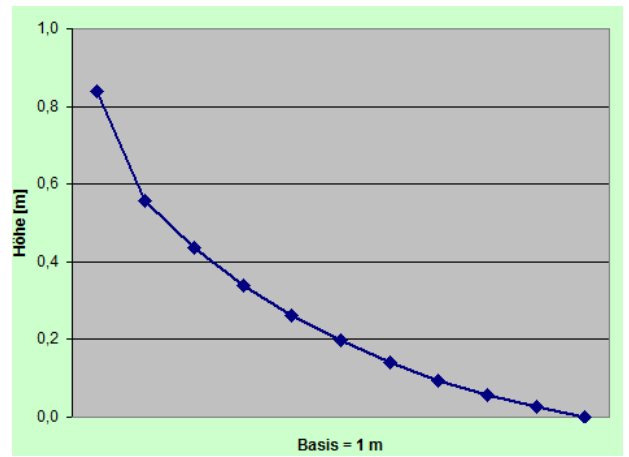
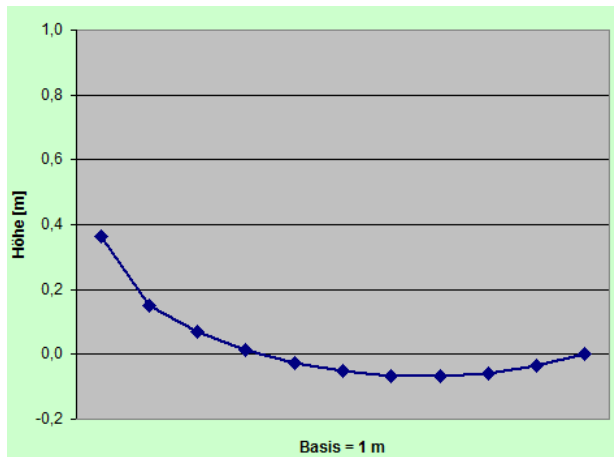


Brachistochrone (Schnellster Weg)

Die Brachistochrone (gr. brachistos kürzeste, chronos Zeit) ist eine reibungsfreie Bahn zwischen einem höher und einem tiefer gelegenen Punkt, auf der ein Massenpunkt unter dem Einfluss der Gravitationskraft am schnellsten zum tieferen Endpunkt gleitet. Der Tiefpunkt der Bahn kann tiefer liegen als der Endpunkt.

Der Körper gleitet auf einer solchen Bahn schneller zum Ziel als beispielsweise auf einer geradlinigen Bahn, obwohl diese kürzer ist.

Ist das Gefälle nicht kleiner als $1/\sqrt{2}$ (31,83 %), so ist B der tiefste Punkt der Kurve, bei kleinerem Gefälle liegt der Tiefpunkt zwischen A und B.



Links sehen wir das Ergebnis für ein Gefälle von 20° und rechts für 40° .

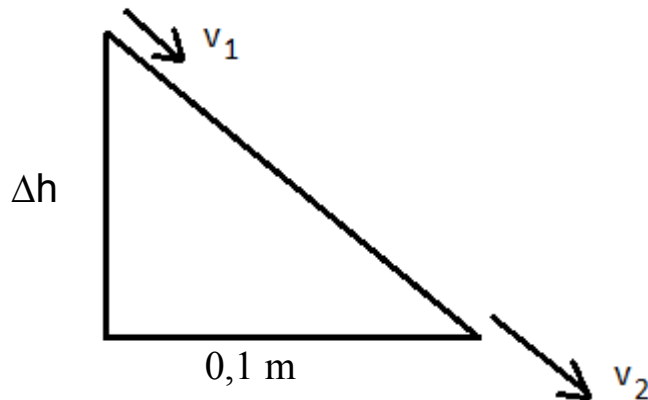


Die Wahlpflichtfachgruppe Biologie-Bionik der 6c-Klasse nahm sich unter der Leitung von Prof. Rouven Schipflinger dem Brachistochronen-Problem an, das schon den Schweizer Mathematiker Johann Bernoulli im 17. Jahrhundert beschäftigte.

Ich habe eine Simulation in Excel programmiert, wo ich die Bahn durch einen Polygonzug annähere. Ich verwende dabei 11 Punkte, deren horizontaler Abstand jeweils 0,1 Meter beträgt.

1) Die benötigten Formeln:

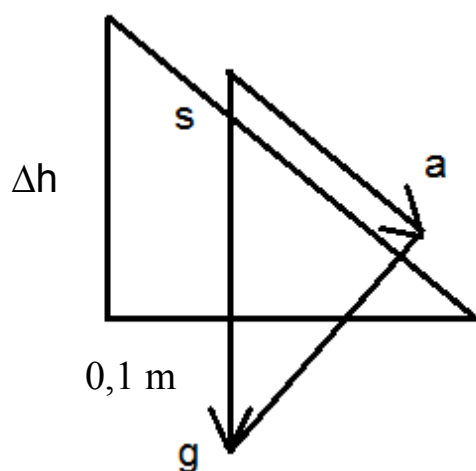
Über den Energiesatz kann man die jeweiligen Geschwindigkeiten berechnen.



Es gilt:
$$\frac{m v_1^2}{2} + m g \Delta h = \frac{m v_2^2}{2}$$

Daraus ergibt sich:
$$v_2 = \sqrt{v_1^2 + 2 g \Delta h}$$

Dann berechnen wir die Beschleunigung auf so einem Abschnitt.



Die Hypotenuse s erhalten wir mit
$$s = \sqrt{0,1^2 + \Delta h^2}$$

und über ähnliche Dreiecke ergibt sich:

$$\frac{a}{g} = \frac{\Delta h}{s}$$

und damit: $a = \frac{\Delta h * g}{s}$

Jetzt kann man die Zeit für so ein Wegstückchen berechnen.

Mit $a = \frac{\Delta v}{\Delta t}$

ergibt sich:

$$\Delta t = \frac{\Delta v}{a}$$

Zusammengesetzt erhält man:

$$\Delta t = \frac{v_2 - v_1}{\Delta h * g} s = \frac{v_2 - v_1}{\Delta h * g} \sqrt{0,1^2 + \Delta h^2}$$

Diese Zeiten werden jetzt aufsummiert.

2) Die Berechnung in Excel:

	A	B	C	D
1	Brachistochrone (Schnellster Weg):			
2				
3	Änderung =	<input type="text" value="1"/>	Zentimeter	
4				
5	Gefälle =	<input type="text" value="20"/>	Grad	
6				
7	h	v	t	
8	0,363970	0,000000	0,000000	
9	0,327573	0,845050	0,251861	
10	0,291176	1,195081	0,104324	
11	0,254779	1,463670	0,080051	
12	0,218382	1,690100	0,067486	
13	0,181985	1,889589	0,059456	
14	0,145588	2,069941	0,053753	
15	0,109191	2,235792	0,049431	
16	0,072794	2,390163	0,046009	
17	0,036397	2,535150	0,043213	
18	0,000000	2,672283	0,040871	
19				
20			Gesamtzeit: 0,796456	

In den Spalten B wird die Geschwindigkeit bestimmt:

$$v_2 = \sqrt{v_1^2 + 2g \Delta h}$$

z.B. in Zelle B9: `=WURZEL(B8^2+2*9,81*(A8-A9))`

In den Spalten C wird die Zeit bestimmt:

$$\Delta t = \frac{v_2 - v_1}{\Delta h * g} \sqrt{0,1^2 + \Delta h^2}$$

z.B. in Zelle C9: `=(B9-B8)/(A8-A9)/9,81*WURZEL(0,1^2+(A8-A9)^2)`

3) Die evolutionäre Bestimmung:

Ausgehend von der geraden Verbindung der Endpunkte werden jetzt per Zufall die Höhen der inneren Punkte variiert (der Maximalwert ist einstellbar) und für jeden Weg wird die Gesamtzeit ermittelt. Der beste Weg wird übernommen und dann wird dieser Weg variiert u.s.w.

Die entsprechenden Programmteile werden als Makros in VBA geschrieben.

Damit die maximale Änderung nicht zu groß gewählt wird, wird dies überprüft und notfalls geändert. Auch Winkel unter 3° werden nicht zugelassen.

Anschließend werden die Punkte auf der geraden Verbindungslinie gesetzt.

```
Sub Ausgangslage()  
  
    If Cells(3, 2) > 3 Then Cells(3, 2) = 3  
    If Cells(5, 2) < 3 Then Cells(5, 2) = 3  
    If Cells(5, 2) < 10 Then Cells(3, 2) = 1  
  
    h = Tan(Cells(5, 2) * 3.141592654 / 180)  
  
    For k = 0 To 9  
  
        Cells(8 + k, 1) = h * (1 - k / 10)  
  
        Cells(27 + k, 1) = h * (1 - k / 10)  
        Cells(27 + k, 5) = h * (1 - k / 10)  
        Cells(27 + k, 9) = h * (1 - k / 10)  
  
    Next  
  
End Sub
```

Jetzt werden die besten Werte umkopiert und anschließend werden davon 3 Variationen berechnet.

```
Sub Schritt()  
  
    Minimum = Cells(42, 2)  
  
    If Cells(39, 3) = Minimum Then  
        For k = 0 To 8  
            Cells(9 + k, 1) = Cells(28 + k, 1)  
        Next  
    End If
```

```
If Cells(39, 7) = Minimum Then
  For k = 0 To 8
    Cells(9 + k, 1) = Cells(28 + k, 5)
  Next
End If
```

```
If Cells(39, 11) = Minimum Then
  For k = 0 To 8
    Cells(9 + k, 1) = Cells(28 + k, 9)
  Next
End If
```

```
For k = 0 To 8
  Cells(28 + k, 1) = Cells(9 + k, 1) + Cells(3, 2) * (Rnd() - 0.5) / 100
  Cells(28 + k, 5) = Cells(9 + k, 1) + Cells(3, 2) * (Rnd() - 0.5) / 100
  Cells(28 + k, 9) = Cells(9 + k, 1) + Cells(3, 2) * (Rnd() - 0.5) / 100
Next
```

```
End Sub
```

Um den Vorgang zu beschleunigen werden hier gleich 100 Schritte ausgeführt.

```
Sub Schritt100()
```

```
  For i = 1 To 100
    Schritt
  Next
```

```
End Sub
```
