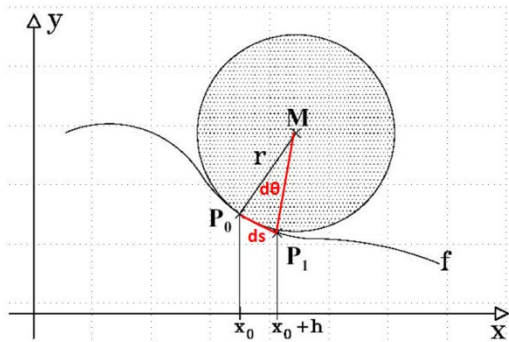




## Kurven mit vorgegebener Krümmung

Es werden Kurven mit vorgegebener Krümmung  $k(s)$  untersucht. Dabei ist  $k(s)$  eine Funktion der Bogenlänge  $s$ . Es ist nur in Einzelfällen möglich, Kurven mit vorgegebener Krümmung analytisch zu berechnen. Der Zugang zu solchen Kurven ist hingegen einfach mithilfe numerischer Integration nach dem Muster der „Turtlegraphik“. Einfach deshalb, weil die Richtungsänderung  $d\vartheta$  der Schildkröte für das nächste Streckenelement  $ds$  gleich dem Produkt  $k(s) \cdot ds$  ist.



$ds$  ist die Länge des Bogenelements. Die Krümmung  $k$  ist ja als Reziprokwert des Krümmungsradius  $r$  definiert und das Bogenelement  $ds$  schneidet aus dem Krümmungskreis einen Sektor mit Öffnung  $d\theta$  aus.  
 $r \cdot d\theta = ds$ .

Die Spur der Schildkröte berechnet sich somit durch folgende 4 Gleichungen :

Bogenlänge bis zum Schritt  $i$  :  $s = i \cdot ds$

Richtungswinkel im Schritt  $i$  :  $\theta_i = \theta_{i-1} + k(s) \cdot ds$

x-Koordinate nach Schritt  $i$  :  $xk_i = xk_{i-1} + ds \cdot \cos(\theta_i)$

y-Koordinate nach Schritt  $i$  :  $yk_i = yk_{i-1} + ds \cdot \sin(\theta_i)$

Diese Gleichungen werden im TI-nspire Programm **kurve()** umgesetzt und in einem Streudiagramm abgebildet.

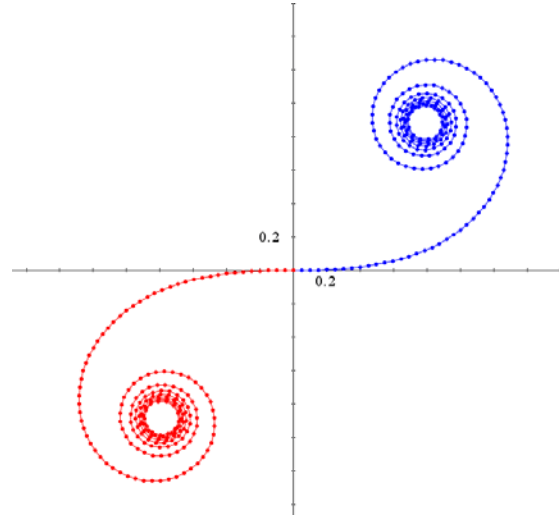


## Erstes Beispiel : Klothoide

Wenn in jedem Kurvenpunkt die Krümmung proportional zur Bogenlänge sein soll,  $k(s) = s$ , entsteht eine Klothoide.

```

kurve
Define kurve( )=
Prgm
xk:=newList(npt): yk:=newList(npt): θ:=newList(npt)
θ[1]:=ds/2 * f(ds): xk[1]:=ds/2 * cos(θ[1]): yk[1]:=ds/2 * sin(θ[1])
For i,2,npt
s:=i * ds
θ[i]:=θ[i-1]+ds * f(s)
xk[i]:=xk[i-1]+ds * cos(θ[i])
yk[i]:=yk[i-1]+ds * sin(θ[i])
EndFor
EndPrgm
  
```



$$k(s) = s, \quad npt = 200, \quad ds = .05$$

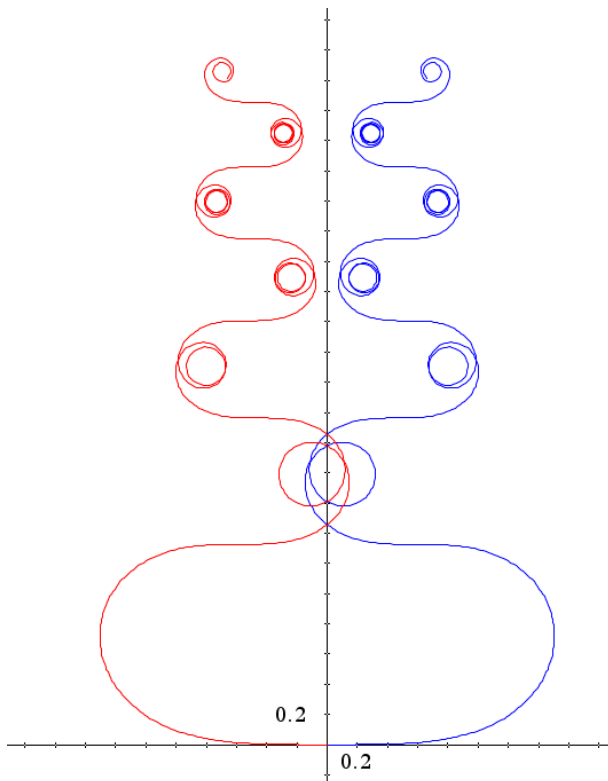
Der rote Kurventeil bildet die Listen  $-xk$  und  $-yk$  ab

Dieses kurze Programm erstellt die Koordinatenlisten  $xk$  und  $yk$



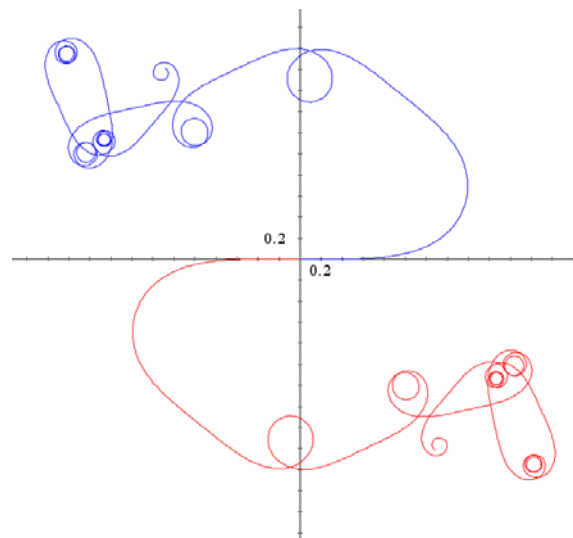
## Weitere Beispiele

Nun kann man verschiedene Krümmungsfunktionen  $k(s)$  ausprobieren, und es eröffnen sich unerwartet komplexe Figuren.



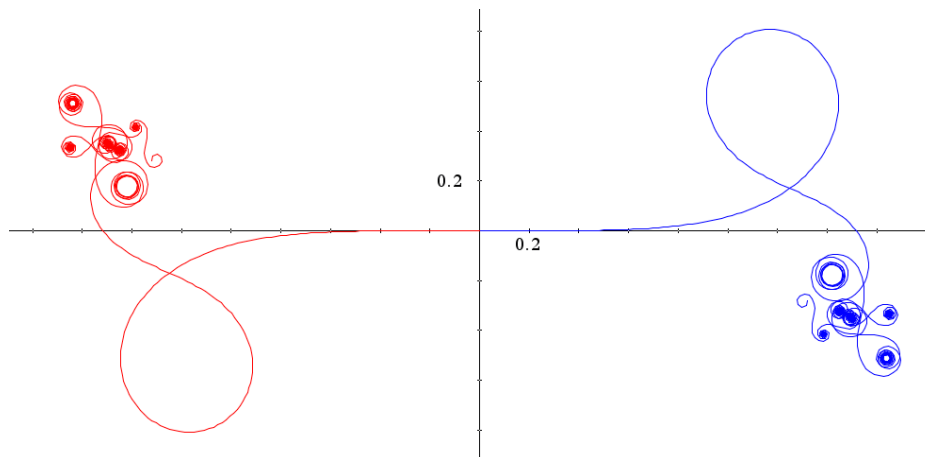
$$k(s) = s \cdot \sin(s), \quad npt = 1000, ds = 0.02$$

Der rote Kurventeil bildet die Listen  $-xk$  und  $yk$  ab



$$k(s) = s \cdot \sin(s)^2, \quad npt = 1000, ds = 0.02$$

Der rote Kurventeil bildet die Listen  $-xk$  und  $-yk$  ab



$$k(s) = s^2 \cdot \sin(s), \quad npt = 1000, ds = 0.02$$

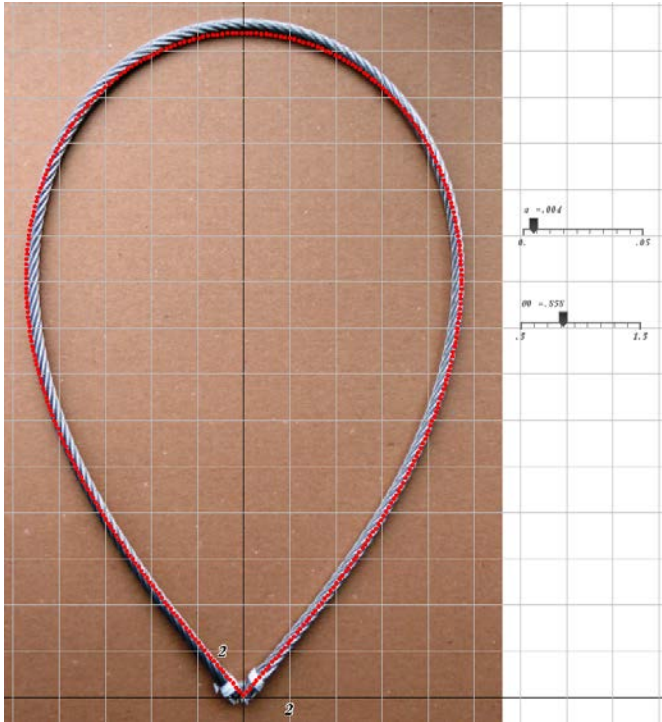
Der rote Kurventeil bildet die Listen  $-xk$  und  $-yk$  ab



Das soweit beschriebene Vorgehen kann man auch in physikalischen Beispielen anwenden.

## Gebogenes Drahtseil

Eine Drahtseil (Stahldraht, Stahlsaite) der Länge 70 cm wird so gebogen, dass sich die Enden berühren. Welche Figur entsteht ?



Praktischer Versuch.  
Länge 70 cm, Durchmesser,  $\varnothing = 4$  mm

Auf den ersten Blick scheint die Krümmung im Bild links an der Spitze unten Null zu sein und oben im Scheitel maximal. Wir berechnen deshalb eine Kurve mit Krümmung proportional zur Ordinate  $y$ .

Das Programm *saite()* unterscheidet sich von *kurve()* nur in der Zeile für  $\theta[i]$ .

```

saite
Define saite()=
Prgm
xk:=newList(npt): yk:=newList(npt): theta:=newList(npt)
theta[1]:=-theta: xk[1]:=0: yk[1]:=ds/2 * sin(theta)
For i,2,npt
theta[i]:=-theta[i-1]+a*ds*yk[i-1]
xk[i]:=xk[i-1]+ds*cos(theta[i])
yk[i]:=yk[i-1]+ds*sin(theta[i])
EndFor
EndPrgm
  
```

Mit  $npt = 350$  Schritten zu  $ds = .2$  cm und den mit den Schieberegler  $a$  und  $\theta_0$  eingestellten Werten ergibt sich eine sehr schöne Übereinstimmung.

Im Parameter  $a$  ist die Physik verpackt. Die Hauptaussage lautet : In stabiler Lage ist das Spannungsmoment auf den Querschnitt gleich dem Biegemoment, verursacht durch die Kraft, mit der die Enden zusammengehalten werden. Das Spannungsmoment ist proportional zur Krümmung und der Proportionalitätsfaktor ist das Produkt der Materialgrößen Flächenträgheitsmoment  $I$  und Elastizitätsmodul  $E$ . Das Biegemoment ist das Produkt aus Kraft  $F$  unten an der Spitze in  $x$ -Richtung mit dem Hebelarm  $y$ . Wir erhalten somit  $\frac{d\theta}{ds} \cdot I \cdot E = F \cdot y$  und die im Programm verwendete Grösse  $a$  wird zu  $a = \frac{F}{I \cdot E}$ .

Die analytische Lösung der Aufgabe führt zu aufwendig handhabbaren elliptischen Integralen. Schön, wie hier mit der einfachen numerischen Methode eine überzeugende Lösung gefunden wird. Für die Form wurde bisher kein Name gefunden. Das ist eigentlich erstaunlich, weil unabhängig von der Art und Länge der Saite immer dieselbe Form entsteht.  $\theta_0$  ist immer  $49^\circ$  und  $a$  ist nur ein Massstabsfaktor.



## Tropfenformen

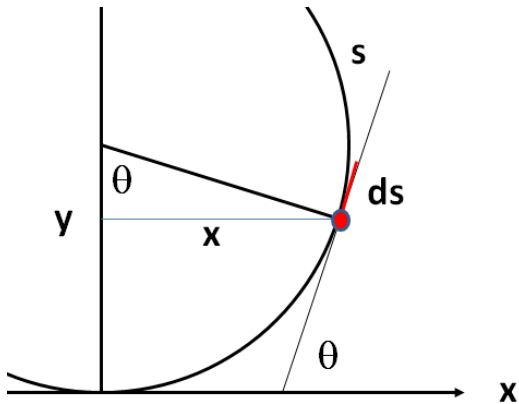


Quecksilbertropfen <http://de.dreamstime.com>



<http://www.sagen.at/fotos/showphoto.php/photo/3713/size/big>

Diese Bilder aus dem Internet faszinieren. Tropfenformen sind schön und daher ein attraktives Sujet in der Makrophotographie, aber auch für diese TI-nspire Anwendung. Es soll die Querschnittskontur eines auf einer Glasplatte liegenden Quecksilbertropfens berechnet werden.



Es geht darum, die physikalische Aussage „hydrostatischer Innendruck = Oberflächenspannung“ in ein System von Differentialgleichungen umzusetzen.

Dazu wird die Konturlänge  $s$  als unabhängige Variable gewählt. Ihr Nullpunkt ist unten.

Der hydrostatische Innendruck beträgt  $p = \rho \cdot g \cdot (h - y)$ .  $\rho$  ist die Dichte,  $g$  die Erdbeschleunigung und  $h$  die Höhe der Flüssigkeitssäule (im Fall eines liegenden Tropfens die Tropfenhöhe plus die Oberflächenspannung im Scheitel, im Fall eines an einer Pipette hängenden Tropfens die Tropfenlänge plus der Überschuss in der Pipette).

Die Oberflächenspannung ist proportional zur Oberflächenkrümmung und diese ist die Summe der Hauptkrümmungen eines Oberflächenelements. Die Krümmungsradien sind mit den Bezeichnungen in der Figur  $R_1 = \frac{ds}{d\theta}$  und  $R_2 = \frac{x}{\sin \theta}$ . Somit erhalten wir

$$\rho \cdot g \cdot (h - y) = \sigma \cdot \left( \frac{1}{R_1} + \frac{1}{R_2} \right) = \sigma \cdot \left( \frac{d\theta}{ds} + \frac{\sin \theta}{x} \right) \text{ und daraus}$$

$$\frac{d\theta}{ds} = \frac{\rho \cdot g \cdot (h - y)}{\sigma} - \frac{\sin \theta}{x} \quad (1)$$

$$\text{Aus der Geometrie folgen } dx = ds \cdot \cos \theta \quad (2) \text{ und } dy = ds \cdot \sin \theta \quad (3)$$



Schliesslich ergibt sich noch eine vierte Gleichung für das Tropfenvolumen  
 $dv = ds \cdot \pi \cdot x^2 \cdot \sin \theta$  (4).

Diese 4 Beziehungen werden nach bisherigem Muster numerisch im TI-Programm **tropfen()** ausgewertet.

```

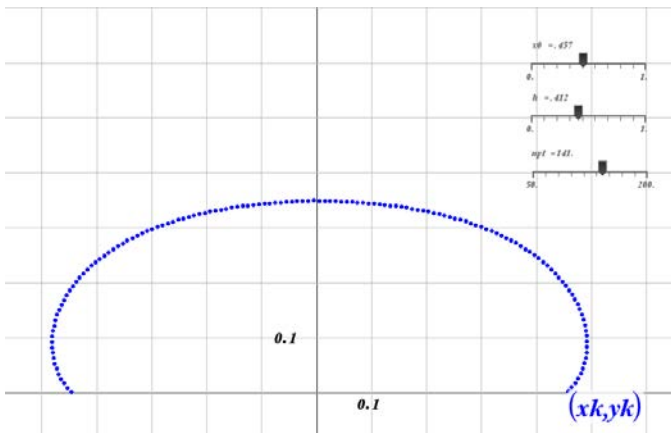
tropfen
Define tropfen()=
Prgm
xk:=newList(npt): yk:=newList(npt): theta:=newList(npt)
theta[1]:=theta0: xk[1]:=x0: yk[1]:=-ds/2 * sin(theta0): vol:=0
For i,2,npt
If |xk[i-1]|>0.1*ds Then
theta[i]:=theta[i-1]+ds * (rho*sigma / (h-yk[i-1]) - sin(theta[i-1]) / xk[i-1])
Else
theta[i]:=0
EndIf
xk[i]:=xk[i-1]+ds*cos(theta[i])
yk[i]:=yk[i-1]+ds*sin(theta[i])
If theta[i]<=0
vol:=vol+ds*pi*xk[i]^2*sin(theta[i])
EndFor
EndPrgm

```

Weil für  $x_k \rightarrow 0$  der Term  $\sin(\theta)/x_k$  unbestimmt wird, ist die Bedingung If-Else-Endif eingefügt.

Mit den Zeilen  
 If  $\theta[i] < \pi$   
 $vol := vol + ds \cdot \pi \cdot xk[i]^2 \cdot \sin(\theta[i])$   
 wird das Tropfenvolumen aufintegriert.

Für Quecksilber sind mit den Einheiten (cm, g, s) :  $\rho = 13.6$ ,  $\sigma = 470$ ,  $\theta_0 = .7$  ( $40^\circ$ )



Die Schieberegler für die Anzahl Schritte npt und den Anfangswert  $x_0$  stellt man so ein, dass das Zielvolumen (hier  $0.2 \text{ cm}^3$ ) erreicht wird. Anschliessend wird h so gesucht, dass die Krümmung im Scheitelpunkt Null ist.

A. Roulier  
[a.roulier@bluewin.ch](mailto:a.roulier@bluewin.ch)

Beilagen :  
 Krümmung.tns