

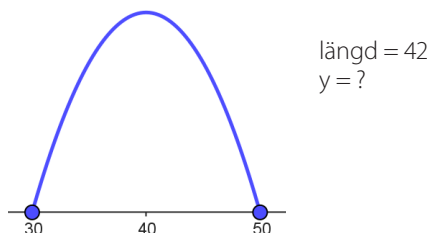
Problemlösning med digitala verktyg

Författaren hittade ett intressant problem i en Facebookgrupp. Här funderar han på hur elever skulle kunna lösa problemet med hjälp av digitala verktyg.

Är det möjligt att ha intressant och utmanande problemlösning i dagens värld av internet, Wolfram Alpha, GeoGebra och AI? Istället för ett hinder, kan de här digitala verktygen vara en del av lösningen? Jag säger ett bestämt "ja" på båda frågorna, särskilt om vi inkluderar programmering i verktygslådan.

Jag såg en spännande uppgift i en Facebookgrupp och här vill jag dela med mig av några tänkbara lösningar, de flesta kräver att eleverna är bekanta med derivata och integraler. Jag använder samma parametrar som i uppgiften. De kan väljas valfritt, men man bör helst undvika tillrättalagda val.

Hitta några "tillräckligt glatta" funktioner som går genom punkterna (30,0) och (50,0) och har kurvlängd 42 mellan punkterna. Vad blir y ?



Digitala verktyg

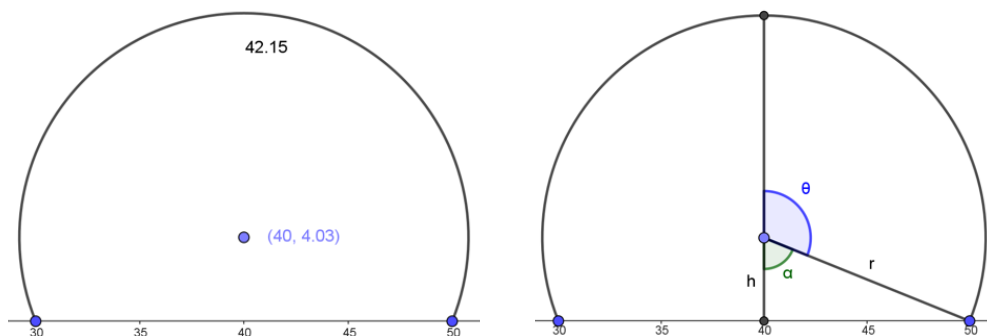
Det finns ingen lätt analytisk metod att lösa uppgiften med. Eleverna kan använda internet, läraren eller några gymnasieböcker och kanske hitta ett sätt för att beräkna kurvlängd:

$$\text{kurvlängd} = \int_a^b \sqrt{1 + (f'(x))^2} dx$$

Som vi snart kommer att se behöver många av integralerna beräknas numerisk, och dessutom behöver vi inkludera en obekant i $f(x)$ då vi siktar på en specifik kurvlängd. Detta gör att man *måste* använda digitala verktyg och det kommer ändå att vara svårt. Notera att ingen lösning kommer att landa på 42 exakt, men alla kan beräknas till godtycklig noggrannhet.

Cirkelbåge

Den lättaste lösningen, som inte kräver en integral, är kanske en cirkelbåge. Detta kräver dock Newton Raphson metod eller dylikt. Hur ska vi börja att resonera här? Det verkar rimligt att anta att cirkelns mittpunkt ligger på $x = 40$ någonstans. Man kan experimentera med GeoGebra för att hitta ungefär var den ska vara och få en bild man kan resonera utifrån:



Med lite algebra och grundläggande trigonometri kan man komma fram till följande ekvation:

$$\sin\left(\frac{21}{r}\right) = \frac{10}{r}$$

Med r och Pythagoras sats kan man få h och då har man ekvation till en cirkel. Ekvationen ovan kan inte lösas analytisk men kan lösas till godtycklig noggrannhet med numeriska metoder. Till exempel:

```
from math import sin, cos
f = lambda x : sin(21/x)-10/x
fp = lambda x : -21*cos(21/x)/x**2+10/x**2
def xn1(xn0):
    return xn0 - f(xn0)/fp(xn0)
x1 = 10
for i in range(10):
    x1 = xn1(x1)
    print(f'r = {x1}')
y = (x1**2 - 100)**0.5
print(f'y = {y}')
```

Ekvationen blir då:

$$(x - 40)^2 + (y - 3.9902122841571694)^2 = 10.766698383099564^2$$

Om man vill ha högre noggrannhet kan man använda biblioteket `mpmath`.

Cosinuskurva

Vi kan försöka med en sinuskurva, men då behöver vi kurvylängdsintegralen. Resonemang med grundläggande trigonometri, eller att experimentera med GeoGebra, leder till följande ekvation:

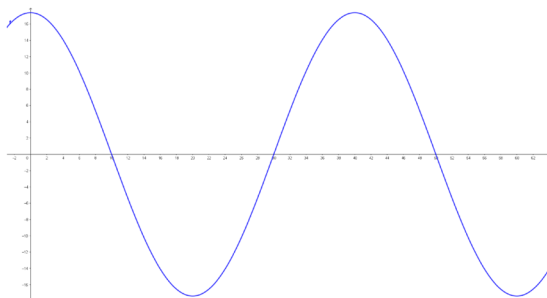
$$y = k \cos\left(\frac{\pi}{20} x\right)$$

Kurvan går genom (30,0) och (50,0) och då gäller det att hitta k så att kurvylängden blir 42. Vi behöver först derivata för att använda integralen:

$$y' = -\frac{k\pi}{20} \sin\left(\frac{\pi}{20} x\right)$$

Vår ekvation blir då:

$$42 = \int_{30}^{50} \sqrt{1 + k^2 \left(\frac{\pi}{20} \sin\left(\frac{\pi}{20} x\right)\right)^2} dx$$



Integralen kan inte lösas analytisk, bara numerisk (den som vill veta mer kan läsa om elliptiska integraler). Ett sätt att gå vidare är att prova sig fram med hjälp av Wolfram Alpha genom att prova olika värden på k . Python är bra även här. Båda bibliotek `scipy` och `mpmath` har funktioner som kan utföra numerisk integration och vi kan då prova olika värden på k automatisk i programmet. Följande är ett exempel som hittar k till 20 decimaler:

```
from numpy import pi, sin, sqrt
from mpmath import mp
mp.dps = 22 #noggrannhet för alla beräkningar
nog = 20 #noggrannhet för k
e = 0 # exponent på ändring
a = 30 # undre gräns
b = 50 # övre gräns
t = 42 # kurvylängd
k = mp.mpf(0)
f = lambda x : mp.sqrt(1+(k*mp.pi/20)**2*mp.sin(mp.pi*x/20)**2)
#optimize k
while e > -nog:
    adj = mp.mpf(10**e)
    while mp.quad(f,[a,b]) < 42:
        k += adj
    k -= adj
    e -= 1
k += adj
print(f'k = {k}')
print(f'längd = {mp.quad(f,[a,b])})')
```

Programmet ger $k = 17.69914781733849995077$. Det går att testa med exempelvis Wolfram Alpha och se hur nära 42 detta kommer.

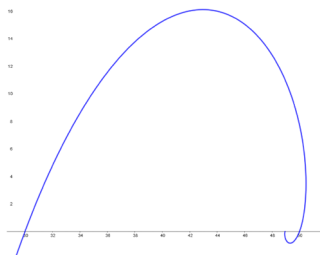
Parametriska ekvationer

Om vi även tillåter parametriska ekvationer har vi fler val. Kurvlängd för parametriska ekvationer beräknas på nästan samma sätt:

$$\text{kurvlängd} = \int_a^b \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt$$

Ett bra exempel att börja med här är baserad på en logaritmisk spiral:

$$x(t) = e^{\frac{t}{\pi} \ln(19)} \cos(t) + 49 \quad y(t) = k e^{\frac{t}{\pi} \ln(19)} \sin(t)$$



Vi kan använda samma metoder som ovan för att hitta k , jag fick $k = 2.5675823041448$. Om vi vill utöka användandet av digitala verktyg går det att överlämna ännu fler steg till Python. Exempelvis räknar följande program ut derivator och integrerar:

```
from sympy import *
from math import e
# symboler i ekvationerna
t, c = symbols('t c')
k = 2 # konstant att justera, börja låg
p = 0 # tiopotens om lägg till k varje gång
nog=14 # antal decimaler att räkna ut
fx = e**((t/pi)*ln(19))*cos(t) # x(t)
dx = diff(fx, t) # dx/dt
fy = Lambda(c, c*e**((t/pi)*ln(19))*sin(t)) # y(t) med konstant c att hitta
dy = Lambda(c, diff(fy(c), t)) # dy/dy
while p>-nog:
    adj = 10**p
    arc = integrate(sqrt((dx**2) + (dy(k)**2)), (t, 0, pi)).
    doit().evalf()
    while arc < 42:
        k+=adj
        arc = integrate(sqrt((dx**2) + (dy(k)**2)), (t, 0, pi))
        .doit().evalf()
        print(k)
    k -=adj #we
    p+=1
print(k, arc)
```

Polynom

De sista två exempel jag kommer att diskutera är polynomekvationer: en andragradsekvation och en tredjegradssekvation. Man kan naturligtvis använda vilken grad som helst.

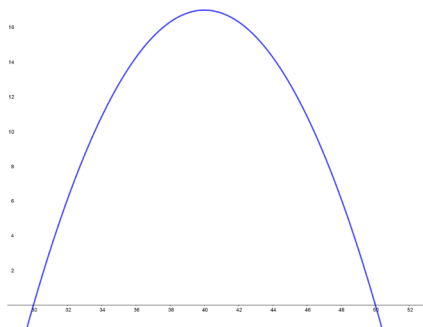
Så länge ekvationen har faktorerna $(x-30)(x-50)$ vet vi att kurvan kommer att gå genom $(30,0)$ och $(50,0)$. Då behöver vi bara lägga till en variabel som vi ska hitta ett bra värde till, exempelvis:

$$y = -k(x-30)(x-50)$$

Vi behöver derivatan igen: $y' = -2k(x-40)$

Kurvlängdsekvationen blir då:

$$42 = \int_{30}^{50} \sqrt{1 + 4k^2(x-40)^2} dx$$



Denna integral kan faktiskt lösas analytisk. Enligt Wolfram Alpha får vi:

$$\int_{30}^{50} \sqrt{1 + 4k^2(x-40)^2} dx = 10\sqrt{400k^2 + 1} + \frac{\sinh^{-1}(20k)}{2k}$$

Om vi använder Wolfram Alpha igen för att sätta den lika med 42, får vi:

$$k \approx 0.174903959182774262163224852$$

Vad händer med en tredjegradssekvation? Tredje roten är godtycklig, så vi kan välja någon lätt:

$$y = -kx(x-30)(x-50)$$

$$y' = -k(3x^2 - 160x + 1500)$$

$$42 = \int_{30}^{50} \sqrt{1 + k^2(3x^2 - 160x + 1500)^2} dx$$

Denna integral kan bara lösas numerisk och vi kan då välja att prova oss fram med exempelvis Wolfram Alpha eller använder Python på samma sätt som ovan. Om ni vill jämföra, får jag följande värde:

$$k = 0.004317157300463107698142$$

Andra lösningar

Det finns andra kurvor eleverna kan testa. Det finns till exempel oändligt många ellipser som skulle fungera.

Om vi släpper kravet att det ska vara tillräckligt glatt och är intresserade av att använda digitala verktyg, kan vi bygga kurvor styckvis vilket blir mycket enklare att beräkna. Ett enkelt exempel är en liksidig triangel med tredje spetsen någonstans på $x = 40$. Detta kräver bara Pythagoras sats.

