

Parametriserade kurvor

1 Inledning

Vi skall se hur man ritar parametriserade kurvor i planet samt hur man ritar tangenter och normaler i punkter längs kurvan. Därefter skall vi approximera en kurvas båglängd.

2 Rita kurvor i planet

Vi har tidigare ritat kurvor i planet med kommandot `plot`, t.ex. enhetscirkeln parametriserad av

$$(x(t), y(t)) = (\cos(t), \sin(t)), \quad 0 \leq t \leq 2\pi$$

som vi ritade upp med

```
>> t=linspace(0,2*pi);  
>> x=cos(t); y=sin(t);  
>> plot(x,y)  
>> axis equal
```

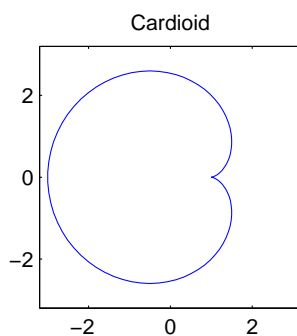
Med `axis equal` får vi rätt utseende (aspect ratio), utan detta skulle cirkeln se tillplattad ut.

Som ytterligare ett exempel tar vi cardioiden

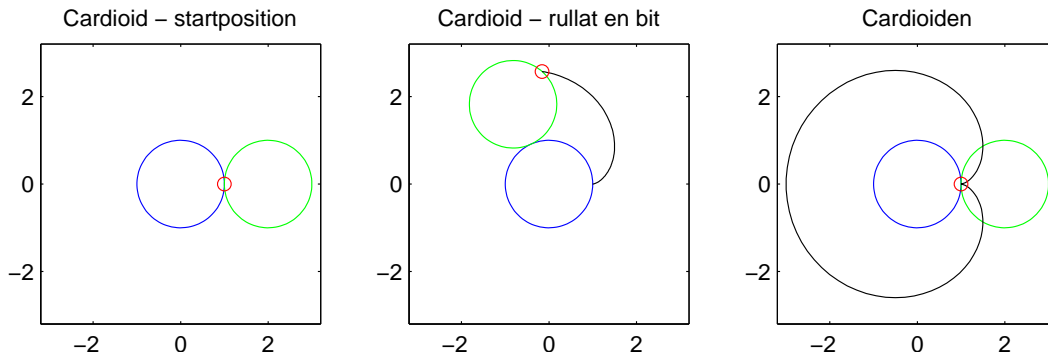
$$(x(t), y(t)) = (2 \cos(t) - \cos(2t), 2 \sin(t) - \sin(2t)), \quad 0 \leq t \leq 2\pi$$

som vi ritar enligt

```
>> t=linspace(0,2*pi);  
>> x=2*cos(t)-cos(2*t); y=2*sin(t)-sin(2*t);  
>> plot(x,y)  
>> axis equal, axis([-3.2 3.2 -3.2 3.2])  
>> title('Cardioid')
```



Kurvan kan konstrueras genom att man ritar av vägen för en punkt, markerad med röd ring, som sitter på kanten av den gröna cirkeln, då vi låter denna rulla, utan att glida, på den blå stillastående cirkeln.



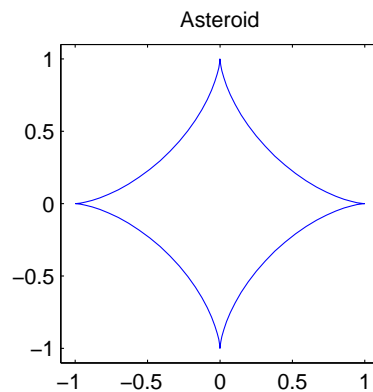
Om du vill kan du köra skriptet `cardioden`, som du finner på laborationshemsidan, så du ser hjulet rulla.

Som ännu ett exempel tar vi asteroiden

$$(x(t), y(t)) = (\cos^3(t), \sin^3(t)), \quad 0 \leq t \leq 2\pi$$

som vi ritar upp i MATLAB enligt

```
>> t=linspace(0,2*pi);
>> x=cos(t).^3; y=sin(t).^3;
>> plot(x,y)
>> axis equal, axis([-1.1 1.1 -1.1 1.1])
>> title('Asteroid')
```



Lägg märke till att vi använder `axis equal` för att få rätt utseende (aspect ratio) och för att få lite "luft" runt kurvan använder vi `axis(...)`. Vidare upphöjer vi med komponentvisa operationer (`t` är ju en vektor).

Uppgift 1. Rita upp cykloiden

$$(x(t), y(t)) = (t - \sin(t), 1 - \cos(t)), \quad 0 \leq t \leq 10\pi$$

Denna kurva kan ses som den väg en myra, som fastnat på ett hjul, färdas när hjulet rullar framåt på en horisontell linje.

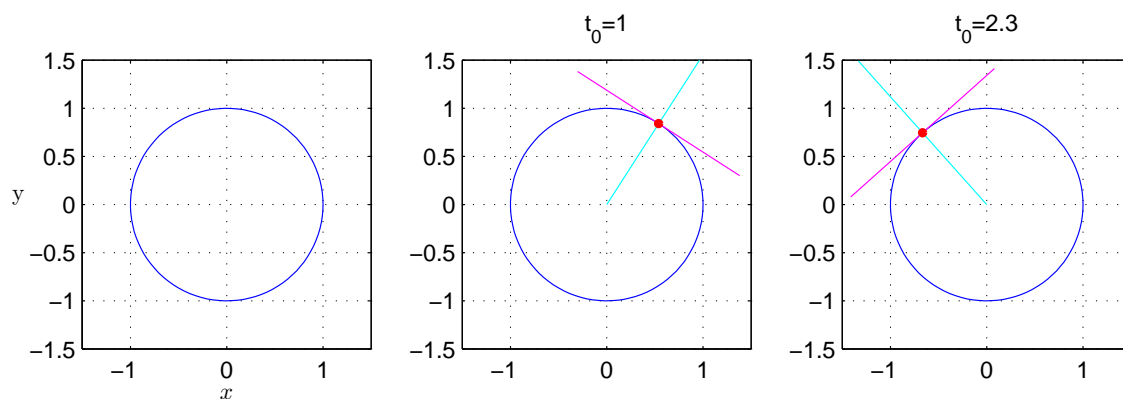
Uppgift 2. Skriv upp en parametrisering av en ellips med storaxel a (x -riktningen) och lillaxel b (y -riktningen) och medelpunkt i (p, q) . Låt $a = 1$, $b = 0.5$ och $p = 2$, $q = 1$. Rita en bild av kurvan. Använd `axis equal`.

Nu skall vi rita tangenter och normaler (Stewart kapitel 10.2). Som ett exempel tar vi cirkeln

$$(x(t), y(t)) = (\cos(t), \sin(t)), \quad 0 \leq t \leq 2\pi$$

```
>> t=linspace(0,2*pi,100);
>> x=cos(t); y=sin(t);
>> plot(x,y)
>> grid on
>> xlabel('x'), ylabel('y','rotation',0)
>> axis equal, axis([-1.5 1.5 -1.5 1.5])
```

Vi ser kurvan nedan till vänster.



För att rita tangenter och normaler får vi beräkna derivatan

$$(x'(t), y'(t)) = (-\sin(t), \cos(t))$$

för att sedan rita upp (en stycke av) de räta linjerna med

```
>> t0=1; s=[-1 1];
>> x0=cos(t0); y0=sin(t0);
>> dx0=-sin(t0); dy0=cos(t0);
>> hold on
>> plot(x0+s*dx0,y0+s*dy0,'m') % tangenten
>> plot(x0+s*dy0,y0-s*dx0,'c') % normalen
>> plot(x0,y0,'ro','markersize',2,'linewidth',2)
>> hold off
```

I bilden ovan till höger ser vi tangenterna och normalerna för ett par olika värden på parametern t och vi ser att allt ser ut som det skall.

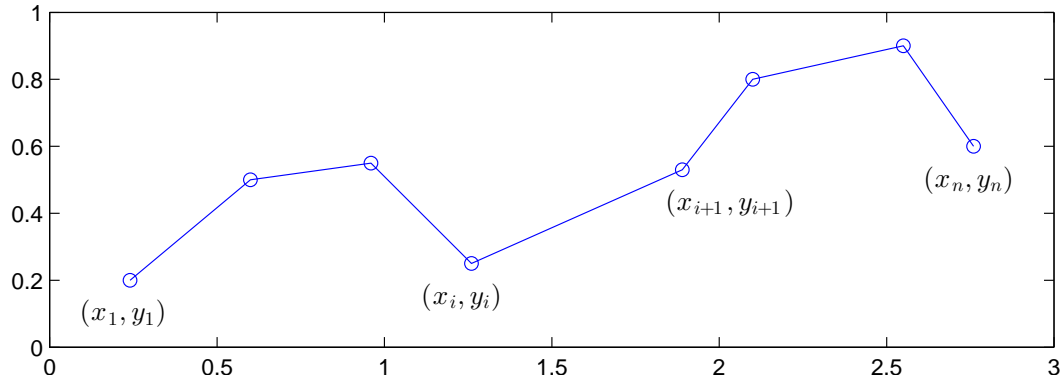
Uppgift 3. Rita upp ellipsen från uppgift 2 tillsammans med tangent och normal i några punkter.

3 Båglängd och polygontåg

Vi tänker oss att vi har ett polygontåg

$$(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), (x_{i+1}, y_{i+1}), \dots, (x_n, y_n)$$

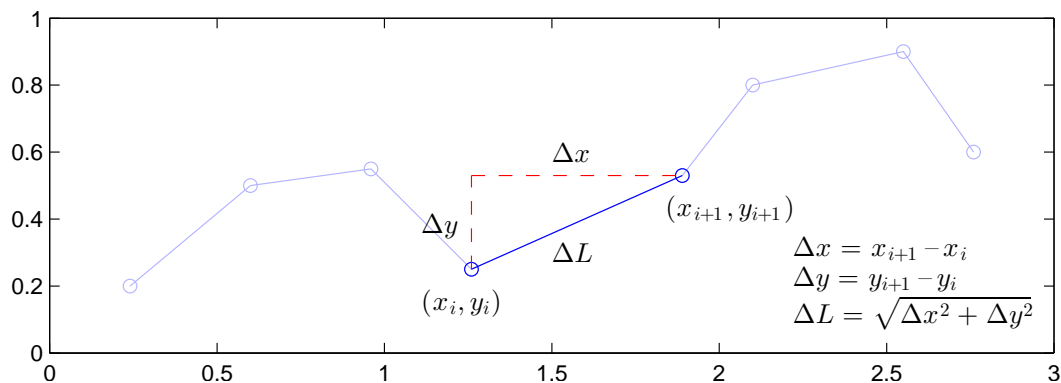
som vi ritat en figur av



Vill vi beräkna polygontågets längd kan vi göra det med formeln

$$L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Denna formel, som vi såg på redan i "Kontrollstrukturer i MATLAB" i den inledande matematik-kursen, fås genom att använda Pytagoras sats på varje segment i polygontåget.



Antag att koordinaterna samlade i två vektorer $\mathbf{x} = (x_1, x_2, \dots, x_n)$ och $\mathbf{y} = (y_1, y_2, \dots, y_n)$, då beräknar vi längden enligt

```
>> n=length(x);
>> L=0;
>> for i=1:n-1
    L=L+sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2);
end
>> L
```

eller lite kortare med vektorisering

```
>> L=sum(sqrt((x(2:end)-x(1:end-1)).^2+(y(2:end)-y(1:end-1)).^2))
```

alternativt

```
>> L=sum(sqrt(diff(x).^2+diff(y).^2))
```

Om vi nu har en parametriserad kurva $(x(t), y(t))$, $a \leq t \leq b$ så kan vi beräkna en approximation av båglängden om vi tar en antal punkter längs kurvan och beräkna längden av motsvarande polygontåg.

Låt $a = t_1 < t_2 < \dots < t_n = b$, med $\Delta t_i = t_{i+1} - t_i$, vara en indelning av parameterintervallet och låt

$$s_n = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Detta är en approximation av båglängden. Ju fler punkter, desto noggrannare resultat.

Vi har

$$s_n = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} = \sum_{i=1}^{n-1} \sqrt{\left(\frac{x_{i+1} - x_i}{\Delta t_i}\right)^2 + \left(\frac{y_{i+1} - y_i}{\Delta t_i}\right)^2} \Delta t_i$$

så om $x'(t)$ och $y'(t)$ kontinuerliga (och inte noll samtidigt) får vi kurvans längd

$$s = \lim_{\substack{n \rightarrow \infty \\ \max \Delta t_i \rightarrow 0}} \sum_{i=1}^{n-1} \sqrt{\left(\frac{x_{i+1} - x_i}{\Delta t_i}\right)^2 + \left(\frac{y_{i+1} - y_i}{\Delta t_i}\right)^2} \Delta t_i = \int_a^b \sqrt{(x'(t))^2 + (y'(t))^2} dt$$

som vi känner igen från Stewart kapitel 10.2.

Uppgift 4. Bestäm en approximation av omkretsen av ellipsen i uppgift 2. Det var ett polygontåg ni ritade upp. Börja med ett fåtal punkter (10 kanske) i polygontåget och öka successivt antalet (ända upp till 100 säg) och se hur approximationen av omkretsen förbättras.

Jämför med resultatet man får då man beräknar båglängden med hjälp av integralen ovan. Använd integral.

1 Målsättning

Avsikten med denna laboration är att vi skall med MATLAB grafiskt studera parametriserade kurvor.

2 Kommentarer och förklaringar

Vi kommer se mer på kurvor, både i planet och rummet, i flervariabelkursen.

3 Lärandemål

Efter denna laboration skall du kunna

- rita parametriserade kurvor i planet med `plot`
- rita tangent och normal till en kurva i en punkt
- beräkna approximationer av båglängden av en kurva