# Studio 1a: Kurvor och ytor i $\mathbb{R}^2$ och $\mathbb{R}^3$.

Analys och Linjär Algebra, del C, K1/Kf1/Bt1, vt09

14 januari 2009

## 1 Kurvor i $\mathbb{R}^2$

Matlab kan som bekant användas för att plotta kurvor i planet. Mest har detta skett genom att ni plottat en funktionsgraf, $y = f(x)$.

```
>> x=0:0.05:2;
>> y=x.^2;
>> plot(x,y)
```

En funktionsgraf är ett specialfall av den allmänna kurvan $g : I \to \mathbb{R}^2$, där $I \subset \mathbb{R}$, som för detta exempel skulle bli $g(t) = (t,t^2)$. I Matlab ser det nästan likadant ut.

```
>> t=0:0.05:2;
>> plot(t,t.^2)
```

Den viktiga skillnaden är att första koordinaten i `plot` nu kan vara en godtycklig funktion i den variabel vi använder.

### Uppgifter

Plotta följande kurvor

1. Asteroiden $g(t) = (\cos^3(t), \sin^3(t)), 0 \leq t \leq 2\pi$.
2. Cykloiden $g(t) = (t - \sin(t), 1 - \cos(t)), 0 \leq t \leq 10\pi$. Kurvan beskriver den väg en partikel på ett hjul färdas när hjulet rullar framåt. Denna ser snyggast ut om man anropar `axis equal`.
3. Kardoiden, eller Pascals snäcka, som definieras genom $r(\theta) = 1 - \cos(\theta), 0 \leq \theta \leq 2\pi$, i polära koordinater, det vill säga $x = r\cos(\theta)$ och $y = r\sin(\theta)$.

Polära koordinater innebär att vi istället för vanliga kartesiska koordinater $x$ och $y$ anger avståndet till origo, *radien* $r$, och vinkeln mot $x$-axeln, $\theta$. Varje punkt $(x, y) \neq (0, 0)$ svarar då mot en unik punkt $(r, \theta)$, om vi sätter $r \geq 0$ och $0 \leq \theta < 2\pi$. Prova också att använda Matlabs kommando `polar`, i detta fall med `polar(t, 1 - cos(t))`.

## 2 Kurvor i $\mathbb{R}^3$

En kurva i rummet definieras som $g : I \to \mathbb{R}^3$. Antag att vi har $g : [0{,}4] \to \mathbb{R}^3$ med $g(t) = (t^{1/2}\cos(\pi t), t^{1/2}\sin(\pi t), t)$. För att plotta denna kurva använder vi kommandot `plot3`:

```
>> t=0:0.05:4;
>> plot3(sqrt(t).*cos(pi*t),sqrt(t).*sin(pi*t),t)
```

### Uppgifter

Plotta följande kurvor.

1. $g(t) = (\cos(t), \sin(t), t), 0 \leq t \leq 2\pi$.
2. $g(t) = (\cos^3(t), \sin^3(t), t), 0 \leq t \leq 2\pi$.

## 3 Ytor

En yta skriver vi som $g : Q \to \mathbb{R}^3$, där $Q \subset \mathbb{R}^2$, eller så kan vi se den som grafen till en funktion $f : Q \to \mathbb{R}$ då vi får $g(x,y) = (x, y, f(x,y))$.

För att plotta ytor i Matlab är det enklast att börja med att skapa *matriser* `X` och `Y` som innehåller alla punkter $(x_{ij}, y_{ij})$ där vi vill beräkna vår funktion. Dessa svarar mot vektorn `t` ovan. Om området vi vill plotta vår funktion över är enhetskvadraten, kan vi göra detta med

```
>> x = 0:.1:1;
>> y = 0:.1:1;
>> [X,Y] = meshgrid(x,y);    % X innehåller alla x_{ij}
                             % Y innehåller alla y_{ij}
```

Vi kan sedan skapa en matris `Z` med motsvarande funktionsvärden.

```
>> Z = X.*Y;   % f(x,y) = xy. Obs! ‘‘Punktvisa’’ operationer!
```

Slutligen skriver vi

```
>> surf(X,Y,Z)          % Alt. >> mesh(X,Y,Z)
```

för att plotta ytan.

Om man som här har ett helt regelbundet rutnät, vilket är det vanliga, är ett tillåtet alternativ att bara ange vektorerna $\boldsymbol{x}$ och $\boldsymbol{y}$ som argument. På så sätt slipper man beräkna matriserna $X$ och $Y$, i alla fall om man klarar att beräkna $Z$ utan $\boldsymbol{x}$ och $\boldsymbol{y}$. I detta fall fungerar `surf(x, y, y'*x)`. Pröva gärna detta alternativ i uppgifterna nedan.

Istället för att rita upp ytan är det ibland bättre att visualisera en funktion genom en konturplot, det vill säga vi ritar nivåkurvor i planet som binder samman punkter där $f(x,y) = c$ för olika $c$. Ett konkret exempel alla har sett är höjdkurvorna på en karta. I Matlab gör vi detta med kommandot `contour`:

```
>> contour(X,Y,Z,v)
```

där $\boldsymbol{v}$ är en vektor som innehåller värdena på de nivåer vi vill rita. (Man kan också utelämna detta argument och låta Matlab bestämma.)

## Uppgifter

1. Plotta de ytor som är grafer till nedanstående funktioner med `surf`:

$$
\begin{array}{lll}
a) & f(x,y) = (x^2 + y^2)^{1/2}, & (x,y) \in [-1,1] \times [-1,1], \\
b) & f(x,y) = x + 2y - 2, & (x,y) \in [1,2] \times [3,4], \\
c) & f(x,y) = x^2 + y^2, & (x,y) \in [-1,1] \times [-1,1], \\
d) & f(x,y) = (1 - y^2)^{1/2}, & (x,y) \in [0,3] \times [-1,1], \\
e) & f(x,y) = x, & (x,y) \in [0,1] \times [0,1], \\
f) & f(x,y) = (1 - x^2 - y^2)^{1/2}, & (x,y) \in \{(x,y) : x^2 + y^2 \leq 1\}, \\
g) & f(x,y) = e^{-\sqrt{x^2+y^2}}, & (x,y) \in \{(x,y) : x^2 + y^2 \leq 4\}.
\end{array}
$$

*Tips:* I *f)* och *g)* kan du använda polära koordinater. Börja med att skapa två lämpliga vektorer $\boldsymbol{r}$ och $\boldsymbol{\theta}$ och därefter (med `meshgrid`) två motsvarande matriser $R$ och $T$. Sedan kan du beräkna `X = R.*cos(T)` och `Y = R.*sin(T)`. Prova att ge kommandona

```
>> subplot(1,2,1)
>> mesh(X,Y,zeros(size(X)))
>> view(2)
>> subplot(1,2,2)
>> mesh(R,T,zeros(size(X)))   % size(R) = size(X)!
>> view(2)
```

Vad ser du? Förklara! Nu när du har beräknat $X$ och $Y$ kan du fortsätta på samma sätt som förut.

2. Betrakta funktionen

$$f(x,y) = x^3 + 3xy^2 - 15x + y^2 - 15y.$$

Plotta ytan över ett lämpligt område samt skapa en konturplot. Rita några enstaka nivåkurvor. (Titta i Matlabs hjälp `help contour` för att ta reda på hur man ska rita en enstaka nivåkurva.)