Grafritning – kurvor och ytor

1 Inledning

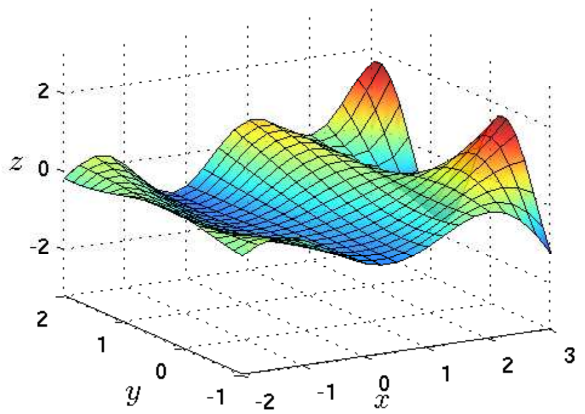
En graf till en funktion i en variabel $f : \mathbb{R} \to \mathbb{R}$ är mängden $\{(x, y) : y = f(x)\}$, dvs. en kurva i planet. En graf till en funktion i två variabler $f : \mathbb{R}^2 \to \mathbb{R}$ är mängden $\{(x, y, z) : z = f(x, y)\}$, dvs. en yta i rummet, en s.k. *funktionsyta*.

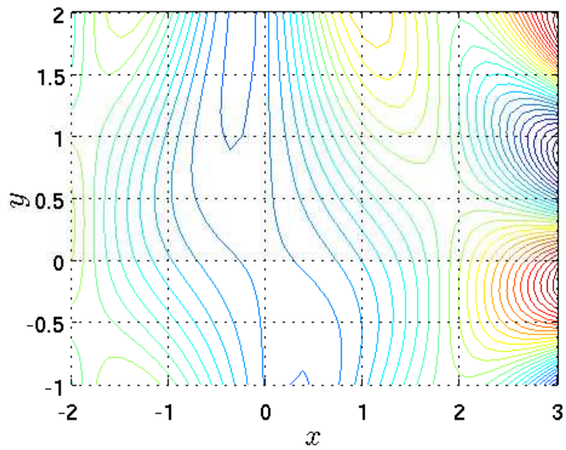
Som exempel tar vi

$$f(x, y) = (\frac{1}{3}x^2 - 1)\sin(1 - xy)$$

över området $-2 \leq x \leq 3$, $-1 \leq y \leq 2$.

Vi ser funktionsytan nedan till vänster.





Ett annat sätt att åskådliggöra en funktion i två variabler $f : \mathbb{R}^2 \to \mathbb{R}$ är att rita *nivåkurvor*, dvs. mängderna $\{(x, y) : f(x, y) = c\}$, där c är en konstant som anger nivån. Vi ser nivåkurvor till vår funktion ovan till höger.

Vi får en slags topografisk karta av funktionen om vi ser funktionsytan som ett landskap och då blir nivån helt enkelt höjden över havet.

En funktion i tre variabler $f : \mathbb{R}^3 \to \mathbb{R}$ kan vi inte rita en graf till. Det skulle vara en tredimensionellt objekt i ett fyrdimensionellt rum. Motsvarigheten till nivåkurvor blir *nivåytor*, dvs. mängderna $\{(x, y, z) : f(x, y, z) = c\}$ där c konstant. Dessa kan vi däremot rita upp.

Låt oss som exempel ta funktionen $f(x, y, z) = x^2 + y^2 + z^2$ i tre variabler. Nivåytorna blir då sfärer $\{(x, y, z) : x^2 + y^2 + z^2 = c\}$ med centrum i origo.

Vi skall också se på vektorvärda funktioner, dels parametrisering av kurvor i planet $\boldsymbol{r} : \mathbb{R} \to \mathbb{R}^2$ och i rummet $\boldsymbol{r} : \mathbb{R} \to \mathbb{R}^3$, dels parametrisering av ytor $\boldsymbol{r} : \mathbb{R}^2 \to \mathbb{R}^3$ i rummet.

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

Vi har redan ritat kurvor i $\mathbb{R}^2$ med kommandot `plot`, när vi skall rita kurvor i $\mathbb{R}^3$ kommer vi använda kommandot `plot3`.

Exempelvis enhetscirkeln parametriserad av $\boldsymbol{r} : \mathbb{R} \to \mathbb{R}^2$ där

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

ritar vi upp med

```
>> subplot(1,3,1)
>> t=linspace(0,2*pi);
>> plot(cos(t),sin(t))
```

Spiralen given av parametriseringen $\boldsymbol{r} : \mathbb{R} \to \mathbb{R}^3$ där

$$\boldsymbol{r}(t) = (x(t), y(t), z(t)) = (\cos(t), \sin(t), t), \quad 0 \leq t \leq 10\pi$$

får vi med (nu använder vi `plot3`)

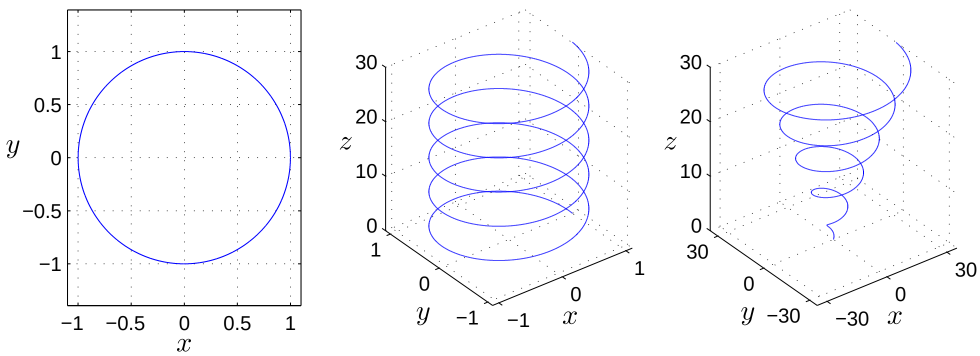
```
>> subplot(1,3,2)
>> t=linspace(0,10*pi);
>> plot3(cos(t),sin(t),t)
```

och den koniska spiralen given av

$$\boldsymbol{r}(t) = (x(t), y(t), z(t)) = (t\cos(t), t\sin(t), t), \quad 0 \leq t \leq 10\pi$$

ritar vi med (lägg märke till komponentvis multiplikation)

```
>> subplot(1,3,3)
>> plot3(t.*cos(t),t.*sin(t),t)
```



Uppgift 1. Rita upp följande kurvor i $\mathbb{R}^2$

(a). Asteroiden $\boldsymbol{r}(t) = (\cos^3(t), \sin^3(t)), \quad 0 \leq t \leq 2\pi$

(b). Cykloiden $\boldsymbol{r}(t) = (t - \sin(t), 1 - \cos(t)), \quad 0 \leq t \leq 10\pi$

Kurvan beskriver den väg en myra, som fastnat på ett hjul, färdas när hjulet rullar framåt. Snyggast ser kurvan ut om man ger kommandot `axis equal`.

Uppgift 2. Rita upp följande kurvor i $\mathbb{R}^3$ med `plot3`

(a). $\boldsymbol{r}(t) = (\cos(10t), \sin(30t), t), \quad 0 \leq t \leq 2\pi$

(b). $\boldsymbol{r}(t) = (\cos^3(10t), \sin^3(10t), t), \quad 0 \leq t \leq 2\pi$

3 Funktionsytor i $\mathbb{R}^3$

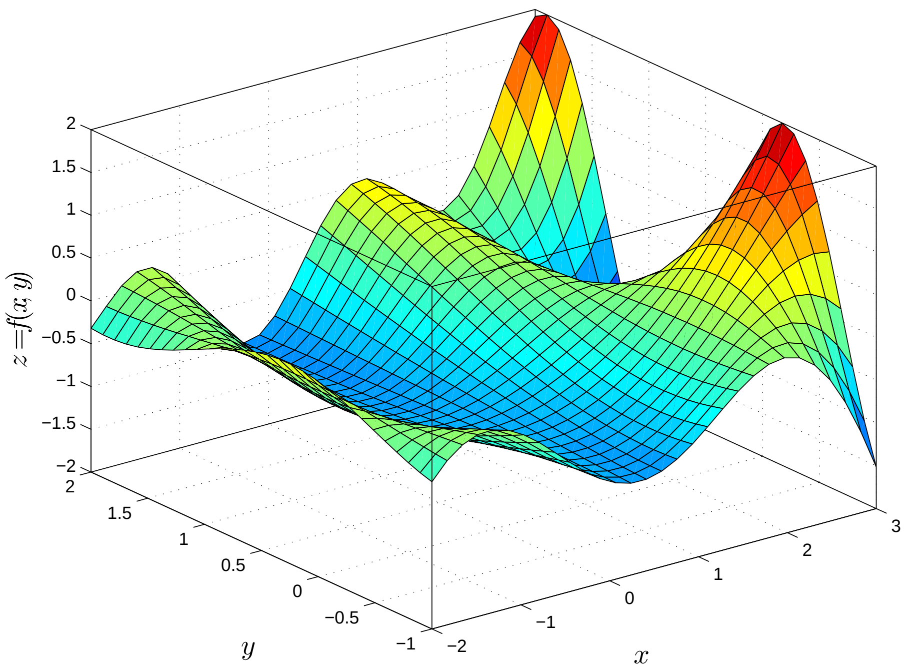
Vi ser på funktionsytan till funktionen från inleningen, dvs. $f : \mathbb{R}^2 \to \mathbb{R}$ där

$$f(x, y) = (\frac{1}{3}x^2 - 1)\sin(1 - xy)$$

över området $-2 \leq x \leq 3$, $-1 \leq y \leq 2$.

Resultatet får vi med kommandot `surf`, vilket är motsvarigheten till `plot` då vi skall rita ytor.

```
>> f=@(x,y)(1/3*x.^2-1).*sin(1-x.*y);
>> x=linspace(-2,3,30); y=linspace(-1,2,30);
>> [X,Y]=meshgrid(x,y);
>> Z=f(X,Y);
>> surf(X,Y,Z)     % eller surf(x,y,Z)
>> grid on, box on
>> xlabel('x'), ylabel('y'), zlabel('z = f(x,y)')
```



Funktionen `meshgrid` får som indata två vektorer `x` och `y`, med x- och y-värden, och ger två matriser `X` och `Y` som utdata.

Dessa matriser är uppbyggda så att vi kan göra en matris `Z` med alla $f(x, y)$-värden på en gång genom att skriva av vår matematiska formel för $f(x, y)$ bara vi använder komponentvisa operationer och samtidigt ersätter x med `X` och y med `Y`.

Ritar vi en graf av en funktion i en variabel tar vi kanske några hundratal x-värden. När vi ritar en funktionsyta tar vi istället några tiotal x- respektive y-värden.

Ibland när man ritar en funktionsyta med `surf(X,Y,Z)` kanske man vill ha den lite genomskinlig och det kan man få med `surf(X,Y,Z,'FaceAlpha',0.7)`. Parametern `FaceAlpha` ges ett värde mellan 0 och 1, där 0 är helt transparent och 1 är helt solid.

Vill vi ha en enfärgad ytan, t.ex. blå, och ytan skall vara nästan helt genomskinlig så ger vi kommandot `surf(X,Y,Z,'FaceColor','b','FaceAlpha',0.1)`. Detta kan vara bra att använda när vi skall rita tangentplan i nästa studioövning.

Uppgift 3. Rita funktionsytan till funktionen $f : \mathbb{R}^2 \to \mathbb{R}$ där

$$f(x, y) = -xye^{-2(x^2+y^2)}$$

över området $-2 \leq x \leq 2,\ -2 \leq y \leq 2$.

Hur fungerar det?

Den yta vi skall rita består av alla punkter $(x, y, f(x, y))$, där $x_{\min} \leq x \leq x_{\max}$ och $y_{\min} \leq y \leq y_{\max}$. Då vi skall rita ytan med `surf` krävs att vi bildar en $m \times n$-matris `Z` med elementen

$$z_{ij} = f(x_j, y_i)$$

där $x_{\min} = x_1 < x_2 < \cdots < x_n = x_{\max}$ och $y_{\min} = y_1 < y_2 < \cdots < y_m = y_{\max}$.

Lägg märke till ordningen på indexen, element z_{ij} skall innehålla funktionsvärdet för $x = x_j$ och $y = y_i$. Stigande x-värden längs rader i `Z`, dvs. stigande kolonn-index, och stigande y-värden längs kolonner i `Z`, dvs. stigande rad-index.

Vi går igenom några alternativa lösningar och vi tänker oss att vi i Matlab redan skapat en funktion `f` och koordinat-vektorer `x` och `y` med `n` respektive `n` element.

Alternativ 1. Vi bildar matrisen `Z` i Matlab med

```
Z=zeros(m,n);
for i=1:m
    for j=1:n
        Z(i,j)=f(x(j),y(i));
    end
end
```

Alternativ 2. I första alternativet fick hålla reda på var det skulle vara `i` respektive `j`. Med funktionen `meshgrid` skapas två matriser `X` och `Y` så att `X(i,j)` har värdet `x(j)` och `Y(i,j)` har värdet `y(i)`, dvs. indexproblemet är borta.

```
[X,Y]=meshgrid(x,y);
Z=zeros(m,n);
for i=1:m
    for j=1:n
        Z(i,j)=f(X(i,j),Y(i,j));
    end
end
```

Alternativ 3. Den allra smidigaste lösningen får vi då vi låter de nästlade repetitionssatserna i tidigare alternativ ersättas av komponentvisa operationer. Dvs. vår funktion `f` måste använda komponetvisa operationer. Vi slipper även initieringen av `Z`.

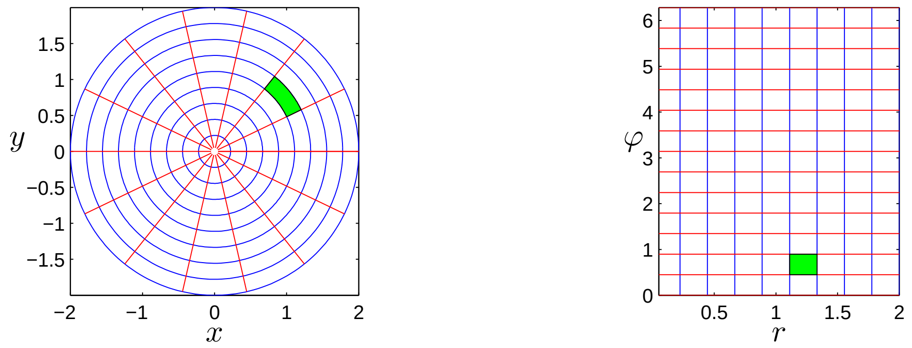
```
[X,Y]=meshgrid(x,y);
Z=f(X,Y);
```

Polära koordinater

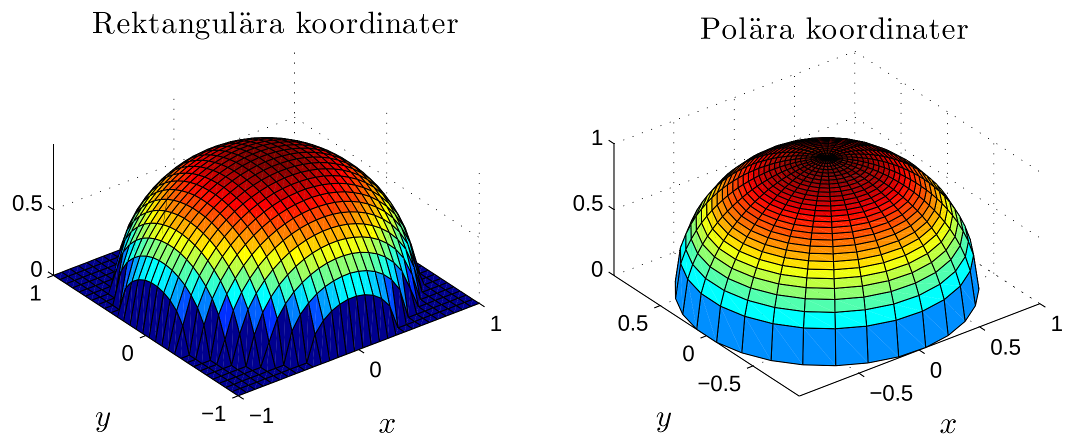
Ibland behöver man byta variabler för att på ett lättare eller bättre sätt behandla ett problem. Ett exempel på variabel- eller koordinatbyte är *polära koordinater*

$$\begin{cases} x = r\cos(\varphi) \\ y = r\sin(\varphi) \end{cases}$$

där $(x, y) \neq (0, 0)$ samt $r > 0$ och $0 \leq \varphi < 2\pi$.



Vill vi t.ex. rita grafen av $f(x, y) = \sqrt{1 - x^2 - y^2}$ över området $x^2 + y^2 \leq 1$, dvs. en halvsfär, och använder rektangulära koordinater får vi den inte alltför snygga figuren nedan till vänster men använder vi polära koordinater får vi den lite snyggare till höger.




```
>> subplot(1,2,1)               % Rektangulära koordinater
>> x=linspace(-1,1,30); y=linspace(-1,1,30);
>> [X,Y]=meshgrid(x,y);
>> U=1-X.^2-Y.^2;
>> U(find(U<0))=0;              % Sätter negativa värden till noll,
>> Z=sqrt(U);                   % så att vi kan ta roten ur utan problem
>> surf(X,Y,Z)

>> subplot(1,2,2)               % Polära koordinater
>> r=linspace(0,1,30); t=linspace(0,2*pi,30);
>> [R,T]=meshgrid(r,t);
>> X=R.*cos(T); Y=R.*sin(T);
>> Z=sqrt(1-R.^2);              % Vi utnyttjar att X.^2+Y.^2=R.^2
>> surf(X,Y,Z)
```

Uppgift 4. Rita upp de ytor som är grafer till nedanstående funktioner med `surf`:

(a). $f(x,y) = x + 2y - 2, \quad (x,y) \in [1,2] \times [3,4]$

(b). $f(x,y) = x^2 - y^2, \quad (x,y) \in [-2,2] \times [-2,2]$

(c). $f(x,y) = (1 - y^2)^{1/2}, \quad (x,y) \in [0,3] \times [-1,1]$

(d). $f(x,y) = (9 - x^2 - y^2)^{1/2}, \quad (x,y) \in \{(x,y) : x^2 + y^2 \leq 9\}$ (Använd polära koordinater)

(e). $f(x,y) = e^{-\sqrt{x^2+y^2}}, \quad (x,y) \in \{(x,y) : x^2 + y^2 \leq 4\}$ (Använd polära koordinater)

Uppgift 5. Återskapa figur 12.4 i Adams. Dvs. rita grafen för funktionen

$$f(x,y) = -6x/(2 + x^2 + y^2)$$

över ett lämpligt område. Tänk på att använda komponentvisa operationer.

4 Nivåkurvor i $\mathbb{R}^2$

Som exempel ser vi på funktionen vars nivåkurvor vi visade en bild av i inledningen

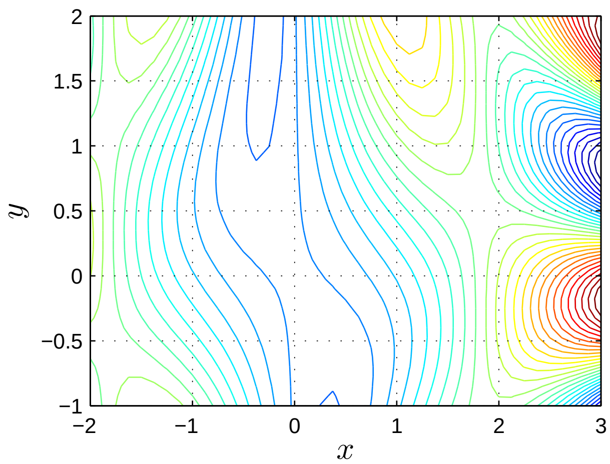
$$f(x,y) = (\frac{1}{3}x^2 - 1)\sin(1 - xy)$$

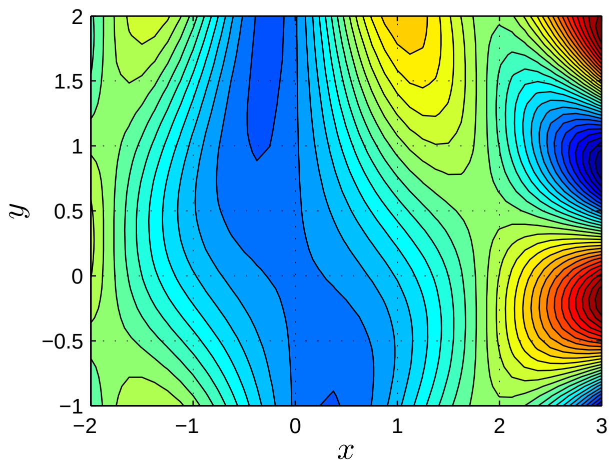
över området $-2 \leq x \leq 3, \ -1 \leq y \leq 2$.

Vi får bilden nedan med

```
>> x=linspace(-2,3,40); y=linspace(-1,2,40);
>> f=@(x,y)(1/3*x.^2-1).*sin(1-x.*y);
>> [X,Y]=meshgrid(x,y);
>> Z=f(X,Y);
>> contour(X,Y,Z,30)     % eller contour(x,y,Z,30)
```

Den lite fylligare bilden nedan till höger fås med `contourf`.





När man ritar nivåkurvor med `contour(X,Y,Z)` får man som standard 20 nivåer jämnt fördelade i spannet av värdena i `Z`, vill vi ha ett annat antal använder vi `contour(X,Y,Z,nl)`, där `nl` är antalet önskade nivåer. Om vi istället vill ange exakt vilka nivåer som skall ritas upp ger vi dem i en vektor `lvec` och använder `contour(X,Y,Z,lvec)`.

I nästa studioövning vill vi rita upp endast nivån där funktionen är noll, dvs. rita noll-nivåkurvor. detta gör vi med `contour(X,Y,Z,[0 0])`. Fundera på varför vi måste ge `0` två gånger.

Uppgift 6. Rita nivåkurvor till funktionerna i uppgift 4. Gå in och redigera cellerna så att ni samtidigt ser både funktionsytan och nivåkurvorna. Använd `subplot`.

Uppgift 7. Betrakta funktionen

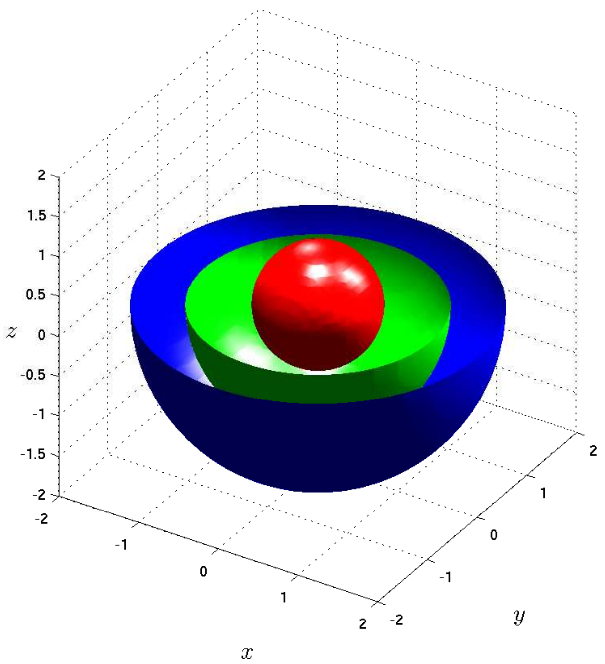
$$f(x,y) = x^3 + 3xy^2 - 15x + y^2 - 15y$$

Rita ytan över ett lämpligt område samt skapa en konturplot. Rita några enstaka nivåkurvor.

5 Nivåytor i $\mathbb{R}^3$

För att rita nivåytor använder vi `isosurface`. Vi tar exemplet från inledningen, dvs. funktionen $f(x,y,z) = x^2 + y^2 + z^2$ i tre variabler. Nivåytorna blir då sfärer $\{(x,y,z) : x^2 + y^2 + z^2 = c\}$ och vi ritar upp för $c = 0.5$.

```
>> x=linspace(-2,2,30); y=linspace(-2,2,30); z=linspace(-2,2,30);
>> [X,Y,Z]=meshgrid(x,y,z);
>> F=X.^2+Y.^2+Z.^2; c=0.5;
>> p=patch(isosurface(X,Y,Z,F,c));
>> set(p,'facecolor','r','edgecolor','none');
>> axis equal, axis([-2 2 -2 2 -2 2]), grid on
```



Vi har även ritat ytterligare två nivåer som vi skalat av överdelen. De tekniska detaljerna får vi skjuta på framtiden, men vi fick till ett rött klot i alla fall.

6 Allmänna ytor i $\mathbb{R}^3$

Som exempel på en allmän yta i rummet tar vi en cylinder med radien ρ och höjden h som beskrivs av ekvationen

$$\{(x,y,z):\ x^2+y^2=\rho^2,\ 0\le z\le h\}$$

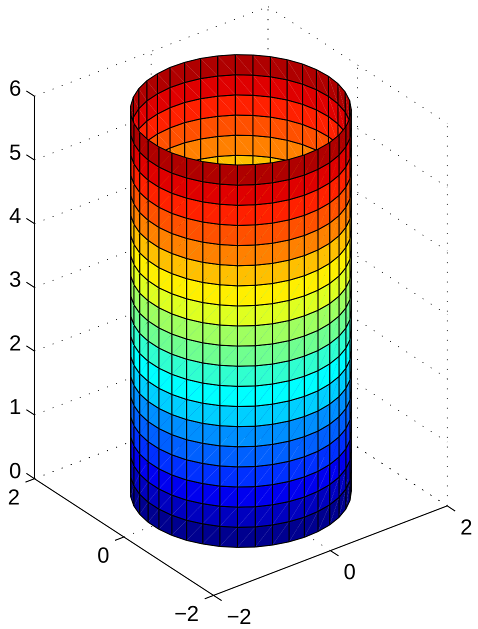
Ytan kan parametriseras $\boldsymbol{r}:\mathbb{R}^2\to\mathbb{R}^3$ med

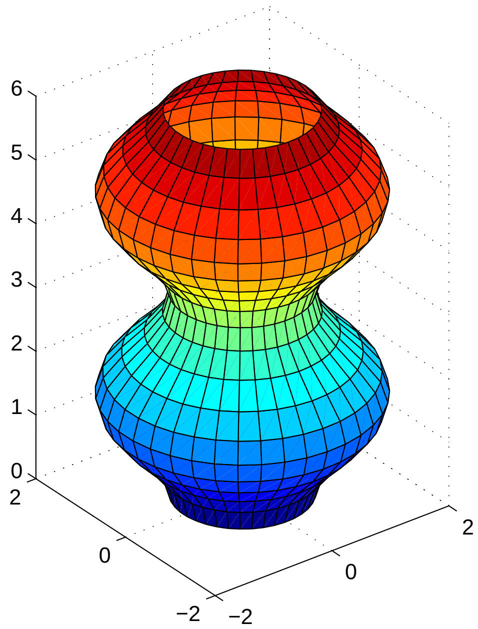
$$\boldsymbol{r}(s,t)=(x(s,t),y(s,t),z(s,t))=(\rho\cos(t),\rho\sin(t),s)$$

där $0\le s\le h$ och $0\le t\le 2\pi$.

Detta passar bra när vi skall rita bilden i MATLAB.

```
>> rho=1.5; h=6; n=20; m=30;
>> s=linspace(0,h,n); t=linspace(0,2*pi,m);
>> [S,T]=meshgrid(s,t);
>> X=rho*cos(T); Y=rho*sin(T); Z=S;
>> surf(X,Y,Z)
>> axis equal, axis([-2 2 -2 2 0 6])
```





För cylindern hade vi $\rho(s)=\rho_0$, dvs. ett konstat värde (samma radie). Om vi istället låter $\rho(s)=1+\sin(s)^2$ (varierande radie) så blir det lite roligare. Detta är ett exempel på en *rotationsyta*. Sådana pratade vi om i ALA-B i samband med volymberäkning (Adams kap 7.1).

Nu ritar vi upp rotationsytan. Var skiljer sig koden nedan från den för cylindern?

```
>> h=6; n=20; m=30;
>> s=linspace(0,h,n); t=linspace(0,2*pi,m);
>> [S,T]=meshgrid(s,t);
>> R=1+sin(S).^2;
>> X=R.*cos(T); Y=R.*sin(T); Z=S;
>> surf(X,Y,Z)
>> axis equal, axis([-2 2 -2 2 0 6])
```

En sfär med radien r och centrum i origo ges av ekvationen

$$x^2 + y^2 + z^2 = \rho^2$$

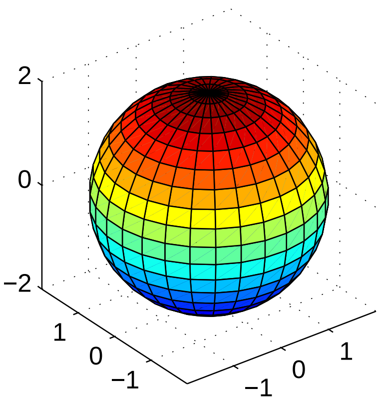
och kan parametriseras $\boldsymbol{r} : \mathbb{R}^2 \to \mathbb{R}^3$ med

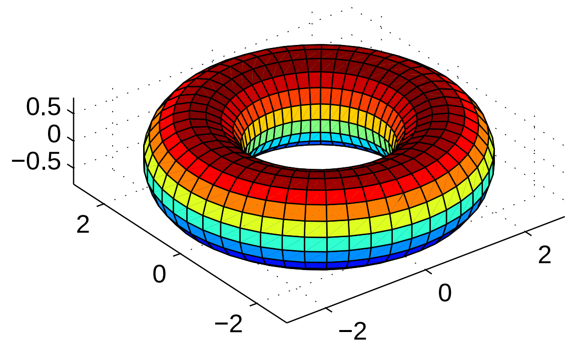
$$\boldsymbol{r}(s,t) = (x(s,t), y(s,t), z(s,t)) = (\rho \sin(s) \cos(t), \rho \sin(s) \sin(t), \rho \cos(s))$$

där $0 \leq s \leq \pi$ och $0 \leq t \leq 2\pi$.

Vi ritar en sfär med radien $\rho = 2$ enligt

```
>> rho=2; n=20; m=30;
>> s=linspace(0,pi,n); t=linspace(0,2*pi,m);
>> [S,T]=meshgrid(s,t);
>> X=rho*sin(S).*cos(T); Y=rho*sin(S).*sin(T); Z=rho*cos(S);
>> surf(X,Y,Z)
```





En torus med lateralradien ρ och centralradien a samt centrum i origo ges av

$$(x^2 + y^2 + z^2 + a^2 - \rho^2)^2 = 4a^2(x^2 + y^2)$$

och kan parametriseras $\boldsymbol{r} : \mathbb{R}^2 \to \mathbb{R}^3$ med

$$\boldsymbol{r}(s,t) = (x(s,t), y(s,t), z(s,t)) =$$

$$= ((a + \rho \cos(s)) \cos(t), (a + \rho \cos(s)) \sin(t), \rho \sin(s))$$

där $-\pi \leq s \leq \pi$ och $0 \leq t \leq 2\pi$.

Vi ritar en torus med lateralradien $\rho = 0.8$ och centralradien $a = 2$ enligt

```
>> rho=0.8; a=2; n=20; m=50;
>> s=linspace(-pi,pi,n); t=linspace(0,2*pi,m);
>> [S,T]=meshgrid(s,t);
>> X=(a+rho*cos(S)).*cos(T); Y=(a+rho*cos(S)).*sin(T); Z=rho*sin(S);
>> surf(X,Y,Z)
>> axis equal
```

I en av de sista studioövningarna i ALA-B ritade vi en kul-och-pinnmodell (ball-and-stick) för några molekylstrukturer. Vi använde två funktioner `klot` och `stav` från studiohemsidan, för att rita ett klot (eller kula) respektive en stav (eller pinne). Dessa funktioner bygger på de parametriseringar för en sfär respektive en cylinder som vi just sett på.

Så här såg funktionen `klot` ut. Nu hade ni förhoppningsvis själva kunnat skriva den.

```
function []=klot(a,r,n,col,alpha)
% klotyta - ritar ett klot runt en punkt i rummet.
%   Syntax:
%           []=klot(a,r,n,col,alpha)
%   Argument:
%           a - en vektor med medelpunkten för klotet.
%           r - klotets radie.
%           n - heltal som anger antal paneler på klotet.
%           col - textsträng som anger färg på klotet.
%           alpha - transparens, alpha=1 solid, alpha=0 transparent.
%   Returnerar:
%           -
%   Beskrivning:
%           -
%   Exempel:
%           r=0.3; n=30; col='g'; alpha=0.7;
%           a=[0; 0; 0];
%           klot(a,r,n,col,alpha)

    [S,T]=meshgrid(linspace(0,pi,n),linspace(0,2*pi,2*n));
    X=a(1)+r*sin(S).*cos(T);
    Y=a(2)+r*sin(S).*sin(T);
    Z=a(3)+r*cos(S);
    surf(X,Y,Z,'facecolor',col,'edgecolor','none','facealpha',alpha)
```

Funktionen `stav` är lite mer komplicerad. Den cylinder som ritas har vanligtvis inte z-axeln som symmetriaxel utan har en sned symmetriaxel (annars skulle molekylerna se konstiga ut). Med hjälp av den linjära algebra ni just lärt kan man klara av denna komplikation, men vi får skjuta det på framtiden.