Linjär algebra, AT3 2011/2012
Matematiska vetenskaper

Några geometriska konstruktioner i $\mathbb{R}^3$

Inledning

Vi skall se på några Platonska kroppar. Dessa är konvexa tre-dimensionella polyedrar som har likformiga polygoner som sidor. Lika många sidor möts i varje hörn och alla hörn är lika. Det finns precis fem Platonska kroppar (tetraeder, kub, oktaeder, dodekaeder och ikosaeder). I sammanhanget vill vi även nämna Arkimediska kroppar. Dessa har lika hörn men sidorna kan vara olika polygoner. Det finns tretton sådana. Titta gärna tillbaka på Ture Westers lilla bok "Structural Order in Space" som ni nog läste under första årets vårtermin.

Några Platonska kroppar

Vi börjar med att rita en liksidig tetraeder med hörnpunkter på enhetssfären och hörnpunkternas koordinater som

$$\left(\frac{2\sqrt{2}}{3}, 0, -\frac{1}{3}\right), \quad \left(-\frac{\sqrt{2}}{3}, \pm\sqrt{\frac{2}{3}}, -\frac{1}{3}\right), \quad (0,0,1)$$

Först lagrar vi koordinaterna som kolonner i en matris `H` i Matlab enligt

```
a=2*sqrt(2)/3; b=-sqrt(2)/3; c=sqrt(2/3); d=-1/3;
H=[ a  b  b  0
    0  c -c  0
    d  d  d  1 ];
```

därefter tar vi och skriver ut de olika hörnens nummer på respektive plats i rummet, notera att `size(H,2)` ger antal kolonner i `H`, dvs. antal hörnpunkter

```
figure(1), clf
axis equal, axis([-2 2 -2 2 -2 2]), axis off, axis vis3d
hold on
for i=1:size(H,2)
    text(H(1,i),H(2,i),H(3,i),num2str(i))
end
```

Skriv in detta i Matlab så blir det begripligt. Vi kan vända och vrida så vi ser var hörnen är placerade. Nu skall vi bilda en matris `S` som skall hålla ordning på vilka hörnpunkter som är hörn på de olika sidorna i tetraedern. På rad 1 i `S`, dvs. `S(1,:)`, lagrar vi numren på hörnen på sidan 1, osv.

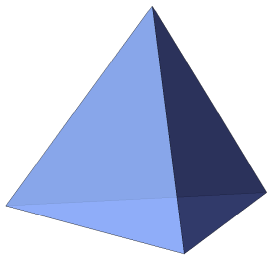
```
S=[ 1  2  3
    1  2  4
    1  3  4
    2  3  4 ];
```

Nu kan vi rita upp sidorna med `fill3`, notera att `size(S,1)` är antal rader i `S`, dvs. antal sidor på tetraedern

```
for i=1:size(S,1)
    Si=S(i,:); fill3(H(1,Si),H(2,Si),H(3,Si),[0.4 0.5 1],'facealpha',0.2)
end
hold off
```

Det är förståndigt att bygga upp `S` rad för rad, och använda koden ovan för att rita fler och fler av tetraederns sidor. På så sätt ser vi vilka sidor vi inte redan har beskrivit.

Så här ser tetraedern ut när vi är färdiga (vi har lagt på belysning också).



Vi kan på samma sätt göra en matris `K` som håller reda på alla kanter på tetraedern.

```
K=[ 1  2
    1  3
    2  3
    1  4
    2  4
    3  4 ];
```

Nu när vi kommit fram till hur `H` och `S` skall se ut kan vi samla ihop koden för att rita tetraedern och för att lägga på lite belysning och sådant

```
a=2*sqrt(2)/3; b=-sqrt(2)/3; c=sqrt(2/3); d=-1/3;
H=[ a  b  b  0
    0  c -c  0
    d  d  d  1 ];
S=[ 1  2  3
    1  2  4
    1  3  4
    2  3  4 ];
figure(1), clf
hold on
for i=1:size(S,1)
    Si=S(i,:); fill3(H(1,Si),H(2,Si),H(3,Si),[0.4 0.5 1],'facealpha',0.8)
end
hold off
axis equal, axis tight, axis off, axis vis3d
view(-30,20)
material shiny
camlight left, camlight head
```

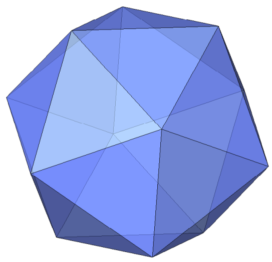
Uppgift 1. Nu är det dags för ikosaedern. Den består av 20 liksidiga trianglar. Koordinaterna för hörnpunkterna kan vi ta som

$$(0, \pm 1, \pm\varphi)\,, \quad (\pm 1, \pm\varphi, 0)\,, \quad (\pm\varphi, 0, \pm 1)$$

där $\varphi = \frac{1+\sqrt{5}}{2}$ (gyllene snittet).

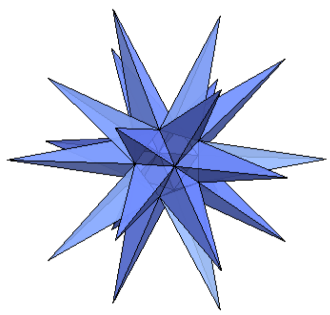
Om vi vill att hörnpunkterna skall ligga på enhetssfären skalar vi med faktorn $s = \frac{1}{\sqrt{1+\varphi^2}}$.

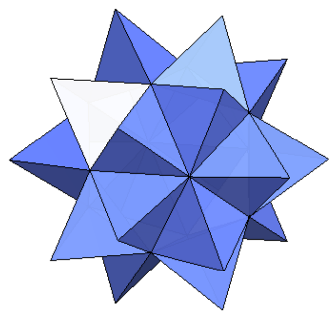
Rita nu upp ikosaedern genom att använda samma teknik som för tetraedern. Vi får jobba lite mer, vi har 12 hörn (istället för 4) och vi har 20 sidor (istället för 4). Något liknande följande bild bör ni komma fram till.



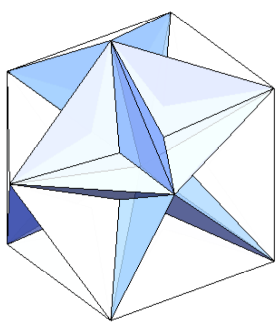
Stjärnformering

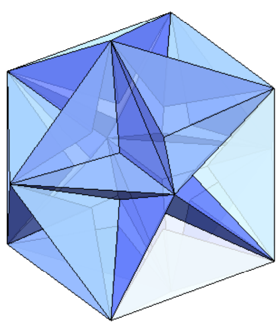
Nu kan vi göra en stjärna av vår ikosaeder. Vi tar ut mittpunkten på varje sida och ritar tre små triangel-flak istället för ett (för varje sida).





Här ovan skjuter vi ut mittpunkten och här nedan drar vi in den.





Koden för att rita sidorna ersätts med

```
for i=1:size(S,1)
    Si=S(i,:); Mi=(H(:,Si(1))+H(:,Si(2))+H(:,Si(3)))/3; % Mittpunkten på sida nr i
    Mi=Mi*5;  % Skalning, pröva olika faktorer. Med 5 får du en långarmad stjärna.
    fill3([H(1,Si(1:2)) Mi(1)],[H(2,Si(1:2)) Mi(2)],[H(3,Si(1:2)) Mi(3)],...
          [0.4 0.5 1],'facealpha',0.8)
    fill3([H(1,Si(2:3)) Mi(1)],[H(2,Si(2:3)) Mi(2)],[H(3,Si(2:3)) Mi(3)],...
          [0.4 0.5 1],'facealpha',0.8)
    fill3([H(1,Si([3,1])) Mi(1)],[H(2,Si([3,1])) Mi(2)],[H(3,Si([3,1])) Mi(3)],...
          [0.4 0.5 1],'facealpha',0.8)
end
```

Dessa stjärnformiga polyedrar är inte Platonska kroppar (varför?).

Uppgift 2. Modifiera nu er kod. Prova lite olika värden på skalfaktorn (även negativa värden är kul) och vänd och vrid.

Stänger och noder

Vi skall rita polyedrar med cirkulära cylindrar som stänger längs kanterna och små sfärer som hörnpunkter eller noder. En sfär gör man enklast med den i MATLAB inbyggda funktionen `sphere`. För att göra en cylinder kan man använda funktionen `cylinder` som ger en cylinder längs en bit av z-axeln och sedan rotera koordinat-matriserna och rita upp.

Vi skall bygga en egen funktion `min_cylinder` som kan generera en cylinder som går från en punkt $\mathbf{x}_1$ till en annan punkt $\mathbf{x}_2$. På liknande sätt som vi konstuerade en rotation runt en godtycklig axel gör vi följande:

Bilda $\mathbf{v} = \mathbf{x}_2 - \mathbf{x}_1$. Därefter väljer vi två vektorer $\mathbf{v}_1$ och $\mathbf{v}_2$, båda av enhetslängd, så att $\mathbf{v}_1$ och $\mathbf{v}_2$ är vinkelräta mot $\mathbf{v}$ och vinkelräta mot varandra. Dvs. $\{\mathbf{v}_1, \mathbf{v}_2\}$ skall vara en ortogonalbas för nollrummet $\mathcal{N}(\mathbf{v}^T)$. Vi beräknar denna bas med funktionen `null`.

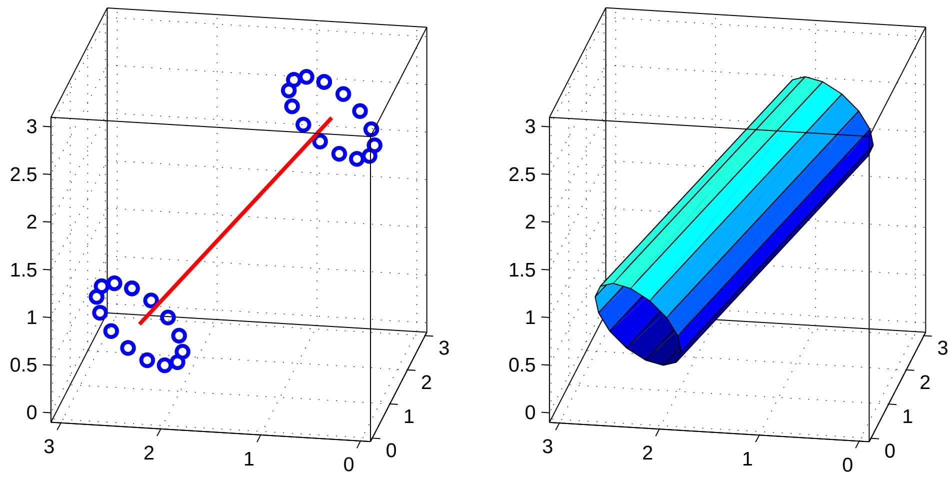
Kanterna på cylindern kommer då ges av cirklarna

$$\mathbf{x} = \mathbf{x}_i + r\cos(t)\mathbf{v}_1 + r\sin(t)\mathbf{v}_2, \quad 0 \le t \le 2\pi, \quad i = 1, 2$$

Vi ritar upp några punkter på dessa cirklar och en röd linje som förbinder $\mathbf{x}_1$ och $\mathbf{x}_2$.

```
figure(1), clf
hold on
x1=[0.4;2.3;0.8]; x2=[2.8;0.8;2.2];
plot3([x1(1) x2(1)],[x1(2) x2(2)],[x1(3) x2(3)],'r','linewidth',2)
v=x2-x1; A=v'; W=null(A); v1=W(:,1); v2=W(:,2);
r=0.5; t=linspace(0,2*pi,15);
C1=x1*ones(size(t))+v1*(r.*cos(t))+v2*(r.*sin(t));
plot3(C1(1,:),C1(2,:),C1(3,:),'ob','linewidth',2)
C2=x2*ones(size(t))+v1*(r.*cos(t))+v2*(r.*sin(t));
plot3(C2(1,:),C2(2,:),C2(3,:),'ob','linewidth',2)
hold off
```

Vi får bilden nedan till vänster



Vi ritar sedan upp vår cylinder med

```
X=[C1(1,:);C2(1,:)]; Y=[C1(2,:);C2(2,:)]; Z=[C1(3,:);C2(3,:)];
surf(X,Y,Z)
```

Vi gör en funktion där vi samlar ihop koden som genererar koordinat-matriserna för vår cylinder.

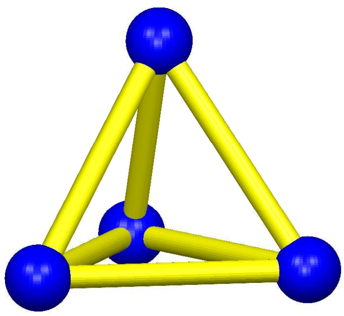
```
function [X,Y,Z]=min_cylinder(x1,x2,r,n)
A=(x2-x1)'; W=null(A); v1=W(:,1); v2=W(:,2);
t=linspace(0,2*pi,n);
C1=x1*ones(size(t))+v1*(r.*cos(t))+v2*(r.*sin(t));
C2=x2*ones(size(t))+v1*(r.*cos(t))+v2*(r.*sin(t));
X=[C1(1,:);C2(1,:)]; Y=[C1(2,:);C2(2,:)]; Z=[C1(3,:);C2(3,:)];
```

Vi ritar tetraeder med blå klot som noder och gula cylindrar som kanter (eller stänger mellan noderna).

```
% Skriptfil
a=2*sqrt(2)/3; b=-sqrt(2)/3; c=sqrt(2/3); d=-1/3;
H=[ a  b  b  0      % H(:,j), j:te kolonnen i H, ger koordinaterna
    0  c -c  0      % för hörnpunkt j.
    d  d  d  1 ];   % size(H,2) ger antal hörn på kuben.
S=[ 1  2  3         % S(i,:), i:te raden i S, ger nr på hörnpunkter
    1  2  4         % på sidan i.
    1  3  4         % size(S,1) ger antal sidor på kuben.
    2  3  4 ];
K=[ 1  2            % K(i,:), i:te raden i K, ger nr på hörnpunkter
    1  3            % på kanten i.
    2  3            % size(K,1) ger antal kanter på kuben.
    1  4
    2  4
    3  4 ];
figure(1), clf
```

```
hold on
[X,Y,Z]=sphere(50); s=0.2;
for j=1:size(H,2)
    surf(H(1,j)+s*X,H(2,j)+s*Y,H(3,j)+s*Z,'facecolor','b','edgecolor','none')
end
r=0.07; m=15;
for i=1:size(K,1)
    Ki=K(i,:);
    [XR,YR,ZR]=min_cylinder(H(:,Ki(1)),H(:,Ki(2)),r,m);
    surf(XR,YR,ZR,'facecolor','y','edgecolor','none')
end
hold off
axis equal, axis tight, axis off, view(20,10), axis vis3d
material metal
camlight left, camlight head
```

Och så här ser det ut



Uppgift 3. Gör nu en ikosaedern med stänger och noder.