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

Inledning

I denna laborationen så skall vi vidareutveckla det vi gjorde i laborationen "Transformationer i $\mathbb{R}^2$ och $\mathbb{R}^3$". Först skall vi studera hur man med hjälp av basbyte kan rotera kring en godtycklig axel, och inte bara längs med koordinataxlarna som i den tidigare laborationen. Sedan använder vi ortogonalitet och nollrum för att rita upp en cylinder, och med hjälp av detta ritar vi upp en tetraeder och en kub på ett nytt sätt.

Rotation runt sned axel i $\mathbb{R}^3$

Betrakta en punkt $\mathbf{x} = (x_1, x_2, x_3)$ i $\mathbb{R}^3$. Antag vi vill rotera pukten runt en axel, vars riktning ges av en vektor $\mathbf{v}$, med en vinkel ϕ. Rotationen skall göras moturs relativt riktningen på vektorn.

Vi kan då via ett basbyte återföra denna rotation till en rotation runt x_1-, x_2- eller x_3-axeln. För dessa har vi redan sett på standardmatriserna.

Säg att vi vill återföra till en rotation runt x_1-axeln. Vi normaliserar $\mathbf{v}$, dvs. vi bildar $\mathbf{v}_1 = \alpha\mathbf{v}$ där skalfaktorn α väljs så att $\mathbf{v}_1$ får enhetslängd. Därefter väljer vi två vektorer $\mathbf{v}_2$ och $\mathbf{v}_3$, båda av enhetslängd, så att $\mathbf{v}_2$ och $\mathbf{v}_3$ är vinkelräta mot $\mathbf{v}_1$ och vinkelräta mot varandra.

Vi skall helt enkelt se till att $\{\mathbf{v}_2, \mathbf{v}_3\}$ blir en ortogonalbas för nollrummet $\mathcal{N}(\mathbf{v}_1^\mathsf{T})$. Denna bas kan vi lätt beräkna i MATLAB med funktionen `null`.

Vi undersöker om $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ är en höger orienterad bas genom att beräkna determinanten

$$D = \det\left([\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]\right).$$

Om $D > 0$ så väljer vi $\mathbf{b}_1 = \mathbf{v}_1$, $\mathbf{b}_2 = \mathbf{v}_2$, $\mathbf{b}_3 = \mathbf{v}_3$ annars tar vi $\mathbf{b}_1 = \mathbf{v}_1$, $\mathbf{b}_2 = \mathbf{v}_3$, $\mathbf{b}_3 = \mathbf{v}_2$.

Nu bildar vi basbytesmatrisen $\mathbf{P} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$ och eftersom kolonnerna är ortogonala och normerade så gäller $\mathbf{P}^{-1} = \mathbf{P}^\mathsf{T}$.

Standardmatrisen för rotation runt axeln som ges av $\mathbf{v}$ blir

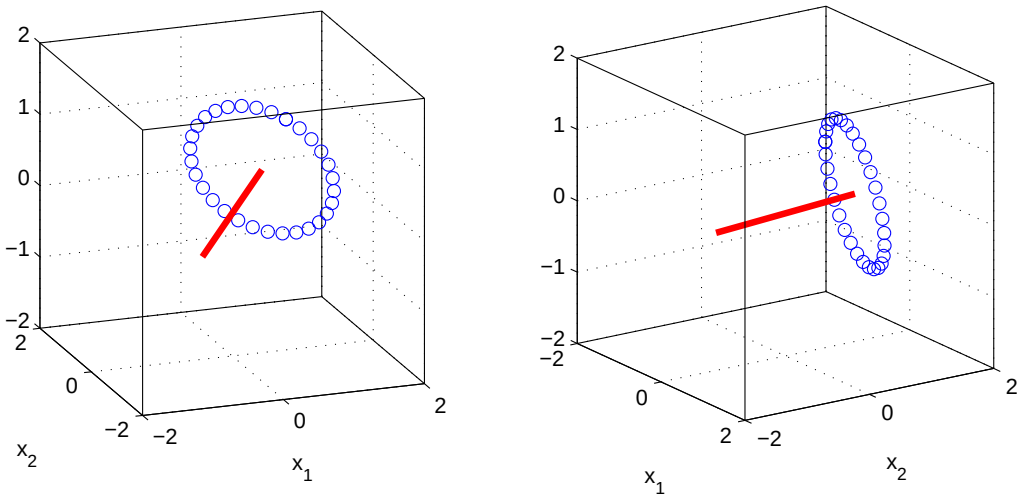
$$\mathbf{A}_\mathbf{v} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1} = \mathbf{P}\mathbf{A}\mathbf{P}^\mathsf{T}$$

där

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

är standardmatrisen för rotationen runt x_1-axeln.

Här ser vi en rotation av en punkt runt en viss axel, upprepad några gånger, från två olika betraktelsevinklar.



Så här gjorde vi i MATLAB

```
phi=pi/15;
A=[1 0 0; 0 cos(phi) -sin(phi); 0 sin(phi) cos(phi)];
v=[2;2;1]; v=v/norm(v); Z=null(v'); P=[v,Z];
if det(P)<0, P(:,[2 3])=P(:,[3 2]); end
Av=P*A*P';
x=[0.8; 0.1; 1.2];
plot3(x(1),x(2),x(3),'o'), hold on
for i=1:30
    x=Av*x;
    plot3(x(1),x(2),x(3),'o')
end
plot3([-v(1) v(1)],[-v(2) v(2)],[-v(3) v(3)],'r','linewidth',3)
box on, grid on, hold off
axis equal, axis([-2 2 -2 2 -2 2]), axis vis3d
```

Uppgift 1. Rotera en punkt runt någon sned axel som ni själva väljer (dock inte samma axel som i exemplet).

Uppgift 2. Rita en kub, precis som vi gjorde i laboration 2. Rotera kuben runt någon sned axel. Gör en translation av kuben bort från origo. Rotera den åter runt någon (annan) sned axel.

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 konstruerade 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 $\mathcal{N}(\mathbf{v}^\mathsf{T})$. Vi beräknar denna bas med funktionen `null`.

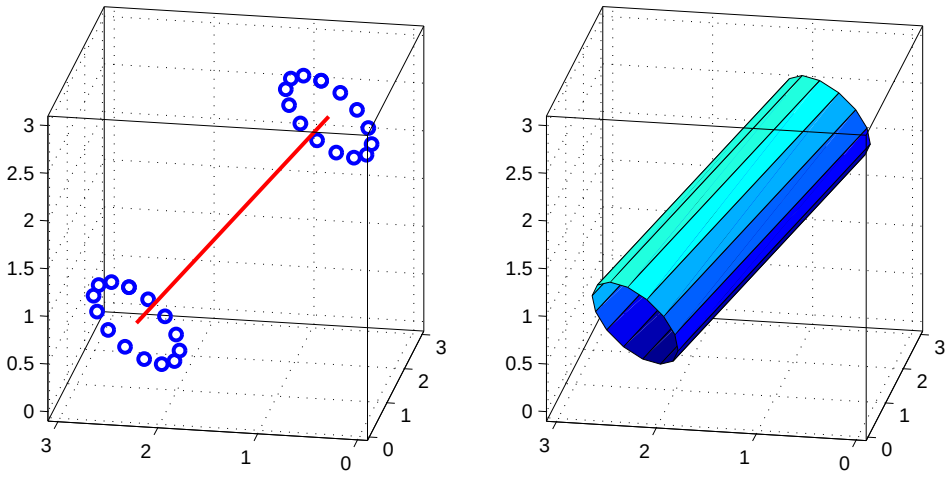
Kanterna på cylindern kommer då ges av cirklarna

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

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)
```

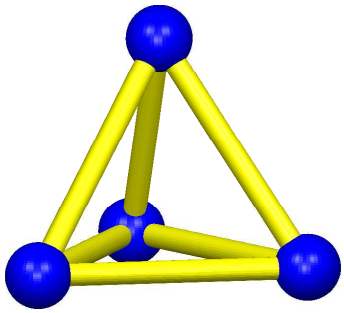
Vi gör en funktion där vi samlar ihop koden som genererar koordinat-matriserna för vår cylinder.

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

```
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 kub med stänger och noder.