Linjär algebra, AT3 2011/2012
Matematiska vetenskaper

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

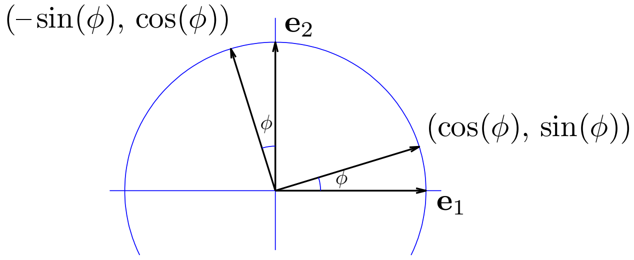
Inledning

Vi skall se på några geometriska transformationer; rotation, skalning, translation och projektion.

Rotation och skalning är linjära avbildningar och kan beskrivas med standardmatriser, däremot är translation *inte* en linjär avbildning. När det gäller projektion får vi skilja på olika fall, t.ex. en ortogonal projektion i $\mathbb{R}^3$ på ett plan är en linjär avbildning om och endast om planet går genom origo.

Rotation, skalning och translation

Som exempel på rotation tar vi: Låt $\mathbf{T} : \mathbb{R}^2 \to \mathbb{R}^2$ vara en rotation motsols med vinkeln ϕ runt origo.



Vi får

$$\mathbf{T}(\mathbf{e}_1) = \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix}, \quad \mathbf{T}(\mathbf{e}_2) = \begin{bmatrix} -\sin(\phi) \\ \cos(\phi) \end{bmatrix}$$

med standardmatrisen

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

Betraktar vi en punkt $\mathbf{x} = (x_1, x_2)$ i $\mathbb{R}^2$ som vi vill rotera runt origo så ges dess nya position av $\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x}$.

Betrakta en punkt $\mathbf{x} = (x_1, x_2, x_3)$ i $\mathbb{R}^3$. Rotation kring x_1-, x_2- och x_3-axlarna ges av $\mathbf{T} : \mathbb{R}^3 \to \mathbb{R}^3$, där $\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ med följande respektive standardmatriser

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

Åter till $\mathbb{R}^2$: Låt $\mathbf{S} : \mathbb{R}^2 \to \mathbb{R}^2$ vara en skalning med faktorerna s_1 och s_2 i respektive axelriktning. Vi får

$$\mathbf{S}(\mathbf{e}_1) = \begin{bmatrix} s_1 \\ 0 \end{bmatrix}, \quad \mathbf{S}(\mathbf{e}_2) = \begin{bmatrix} 0 \\ s_2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$$

En translation med vektorn $\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$ ges av avbildningen $\mathbf{F} : \mathbb{R}^2 \to \mathbb{R}^2$,

$$\mathbf{F}(\mathbf{x}) = \mathbf{x} + \mathbf{t} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

men här finns ingen standardmatris.

Motsvarande för $\mathbb{R}^3$ blir

$$\mathbf{S}(\mathbf{x}) = \mathbf{B}\mathbf{x} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

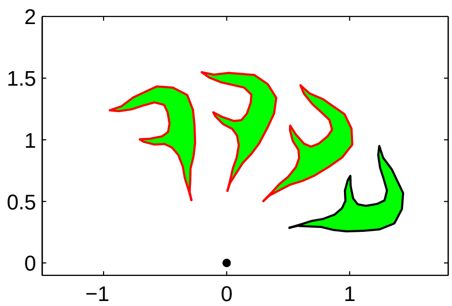
respektive

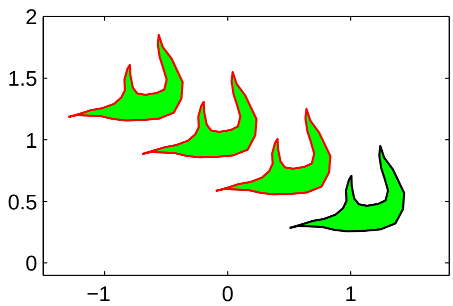
$$\mathbf{F}(\mathbf{x}) = \mathbf{x} + \mathbf{t} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Vi illustrerar rotation och translation i $\mathbb{R}^2$ med MATLAB. I figuren nedan till vänster ser vi ett polygonområde med svart rand som vi roterar några gånger med vinkeln $\frac{\pi}{6}$ och ritar de roterade områdena med röd rand.

Vi tänker oss att vi redan skapat koordinater i radvektorerna `X` och `Y` som beskriver det ursprungliga området.

```
fill(X,Y,'g','edgecolor','k','linewidth',1), hold on
axis equal, axis([-1.5 2 -0.1 2]), pause(1)
v=pi/6; A=[cos(v) -sin(v); sin(v) cos(v)];
P=[X;Y];
for i=1:3
    P=A*P;   % Varje koordinatpar roteras med vinkeln pi/6
    fill(P(1,:),P(2,:),'g','edgecolor','r','linewidth',1), pause(1)
end
plot(0,0,'ko','linewidth',2,'markersize',2)  % origo
hold off
```





Till höger ser vi samma område i ursprungsläget (svart kant) samt några upprepade translationer (röd kant) med vektorn t.

```
fill(X,Y,'g','edgecolor','k','linewidth',1), hold on
axis equal, axis([-1.5 2 -0.1 2]), pause(1)
t=[-0.6;0.3];
```

```
P=[X;Y];
for i=1:3
    P=P+t*ones(size(X));
    fill(P(1,:),P(2,:),'g','edgecolor','r','linewidth',1), pause(1)
end
hold off
```

Uppgift 1. Rotera och translatera ett polygonområde ni genererar själva, t.ex. en triangel.

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 $\mathcal{N}(\mathbf{v}_1^T)$. Denna bas kan vi lätt beräkna i MATLAB med funktionen `null`.

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

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

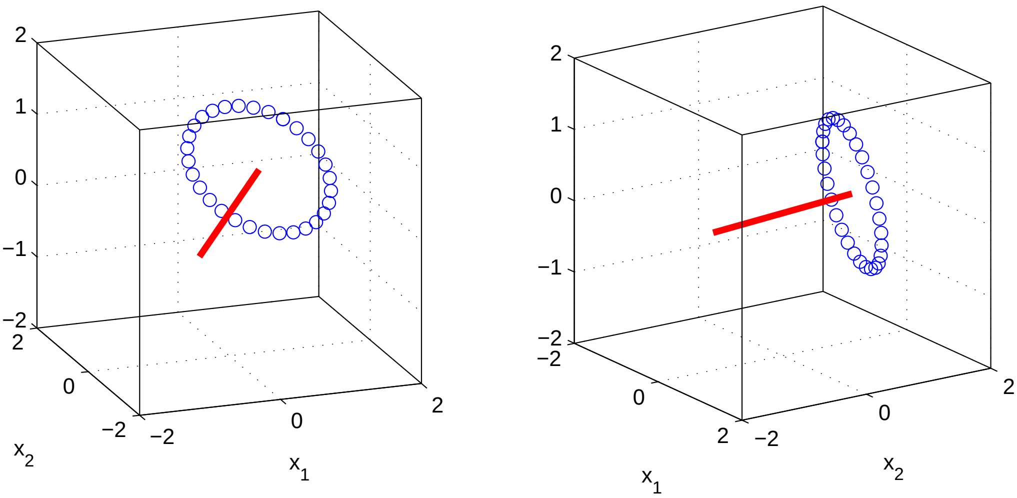
$$\mathbf{A}_\mathbf{v} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1} = \mathbf{P}\mathbf{A}\mathbf{P}^T$$

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.

Detaljerna ovan förstår ni nog först lite längre in i kursen. Det hindrar inte att vi kan rotera i MATLAB redan nu. Här ser vi en rotation av en punkt runt en viss axel, upprepad några gånger, från två olika betraktelse vinklar.

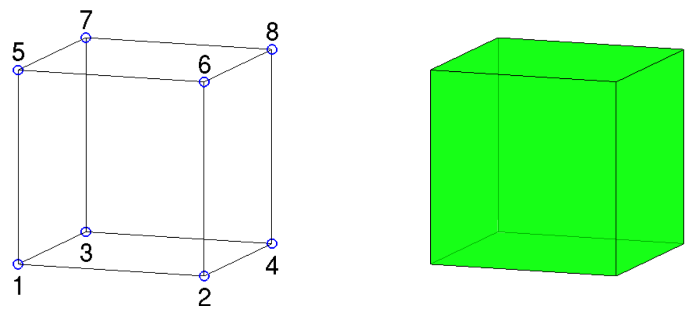


Så här gjorde vi i MATLAB

```
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];
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 2. Rotera en punkt runt någon sned axel som ni själva väljer.

Nu ritar vi en kub som vi sedan skall transformera på lite olika sätt.



Vi ritar kuben enligt

```
H=[0 1 0 1 0 1 0 1      % H(:,j), j:te kolonnen i H, ger koordinater för punkt j
   0 0 1 1 0 0 1 1
   0 0 0 0 1 1 1 1];
S=[1 2 4 3              % S(i,:), i:te raden i S, ger nr på hörnpunkter på sida i
   1 2 6 5
   1 3 7 5
   3 4 8 7
   2 4 8 6
   5 6 8 7];            % size(S,1) ger antal sidor
figure(1), clf
hold on
for i=1:size(S,1)
    Si=S(i,:); fill3(H(1,Si),H(2,Si),H(3,Si),'g','facealpha',0.7)
end
hold off
axis equal, axis tight, axis off, view(20,10)
```

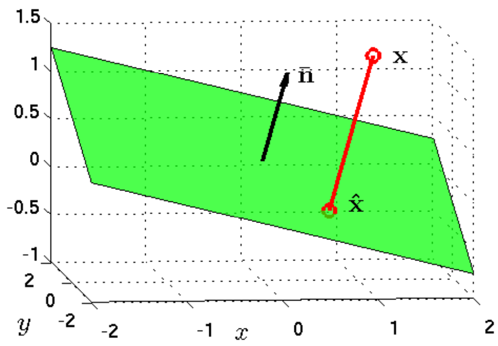
Uppgift 3. Rotera kuben runt någon axel. Gör en translation av kuben bort från origo. Rotera den åter runt någon axel.

Ortogonal projektion

Vi skall bestämma ortogonala projektionen på planet

$$ax + by + cz = d$$

Planet har normalvektorn $\mathbf{n} = (a, b, c)$. I bilden har vi ritat enhetsnormal $\bar{\mathbf{n}}$ och ortogonala projektionen $\hat{\mathbf{x}}$ längs enhetsnormalen av en punkt $\mathbf{x}$.



Vi gör ansatsen $\hat{\mathbf{x}} = \mathbf{x} + \alpha\,\mathbf{n}$, där α skall bestämmas så att $\hat{\mathbf{x}}$ ligger på planet. Ekvationen för planet kan skrivas

$$\mathbf{n} \cdot \hat{\mathbf{x}} = d$$

och sätter vi in ansatsen får vi

$$\mathbf{n} \cdot (\mathbf{x} + \alpha\mathbf{n}) = \mathbf{n} \cdot \mathbf{x} + \alpha\,\mathbf{n} \cdot \mathbf{n} = d$$

och därmed

$$\alpha = \frac{d - \mathbf{n} \cdot \mathbf{x}}{\mathbf{n} \cdot \mathbf{n}}$$

Nu skall vi i MATLAB rita planet $ax + by + cz = d$, för $a = 1, b = -1, c = 4$ och $d = 1$. Eftersom $c \neq 0$ så kan vi lösa ut z, i annat fall får vi modifiera koden

```
xmin=-2; xmax=2; ymin=-2; ymax=2;
a=1; b=-1; c=4; d=1;
X=[xmin xmax xmax xmin]; Y=[ymin ymin ymax ymax];
Z=(d-a*X-b*Y)/c;
figure(1), clf
fill3(X,Y,Z,'g','facealpha',0.7)
xlabel('x'), ylabel('y')
grid on
```

Resultatet blir planet i figuren ovan.

Uppgift 4. Rita planet vi just tittade på. Bestäm normalvektorn och rita ut den som ett streck från en punkt på planet. Välj en punkt $\mathbf{x}$, rita ut den, bestäm dess ortogonala projektion $\hat{\mathbf{x}}$ på planet och rita ut även denna punkt.

Vad blir formeln för spegling i planet? Rita även ut speglingen $\mathbf{x}_r$ av $\mathbf{x}$.

Uppgift 5. Om $d = 0$ i ekvationen för planet så går planet genom origo. Då blir ortogonala projektionen en linjär avbildning. Vad blir standardmatrisen?

Uppgift 6. Rita planet från uppgift 4. I samma bild skall ni rita en translaterad kub. Translationen skall vara sådan att kuben inte skär planet. Rita därefter speglingen av kuben i planet.