Geometriska transformationer

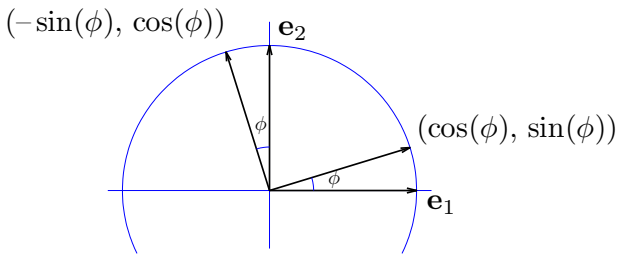
1 Inledning

Vi skall se på några geometriska transformationer; *rotation*, *skalning* och *translation*. Rotation och skalning är linjära avbildningar och kan beskrivas med standardmatriser, däremot är translation *inte* en linjär avbildning.

I nästa laboration skall vi se på ytterligare några transformationer; *ortogonal projektion* och *spegling*. När det gäller projektion och spegling får vi skilja på olika fall, t.ex. en ortogonal (vinkelrät) projektion i $\mathbb{R}^3$ på ett plan är en linjär avbildning om och endast om planet går genom origo.

2 Rotation, skalning och translation i $\mathbb{R}^2$

Som exempel på rotation tar vi: Låt $\mathbf{T} : \mathbb{R}^2 \to \mathbb{R}^2$ vara en rotation moturs 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 $\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x}$.

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

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

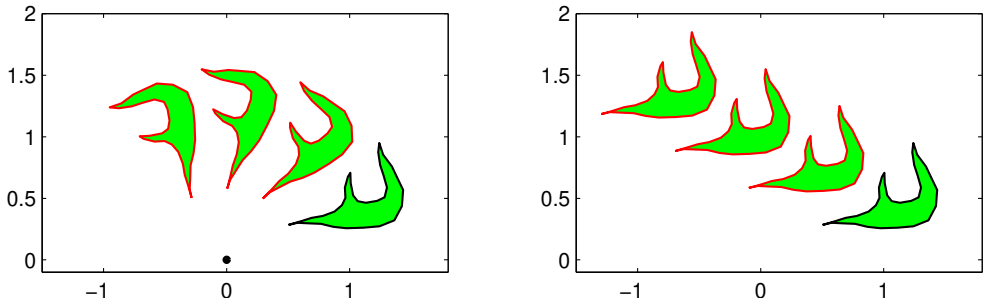
En translation med vektorn $\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$ ges av avbildningen $\mathbf{F} : \mathbb{R}^2 \to \mathbb{R}^2$,

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

men här finns ingen standardmatris.

Vi illustrerar 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 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.

3 Rotation, skalning och translation i $\mathbb{R}^3$

Betrakta en punkt $\mathbf{x} = (x_1, x_2, x_3)$ i $\mathbb{R}^3$. Rotation med vinkeln ϕ 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

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

Rotationen sker medurs sett i axelriktningarna.

Motsvarande för skalning och translation blir

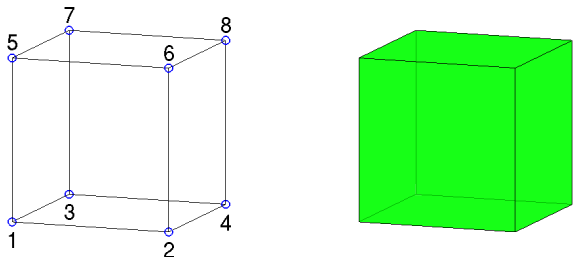
$$\mathbf{S}(\mathbf{x}) = \mathbf{B}\mathbf{x} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

respektive

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

I nästa laboration skall vi, förutom ortogonal projektion och spegling, även se på rotation runt en sned axel i $\mathbb{R}^3$.

Nu skall ni rita en tetraeder som ni sedan skall transformera på lite olika sätt. För att underlätta ritandet av tetraedern visar vi hur man kan rita en kub.



Vi ritar en enhetskub enligt

```
H=[0 1 0 1 0 1 0 1   % H(:,j), j:te kolonnen i H, ger koordinater för hörnpunkt j
   0 0 1 1 0 0 1 1   % size(H,2) ger antal hörnpunkter
   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
axis equal, axis tight, axis off, hold off, view(20,10)
```

Lägg lite tid på att tänka igenom det vi gjort. Hörnpunkternas koordinater ligger som kolonner i matrisen `H`. På raderna i matrisen `S` har vi numren på hörnpunkterna på sidorna, t.ex. första raden (`1 2 4 3`) ger numren på hörnpunkterna som ger botten på kuben.

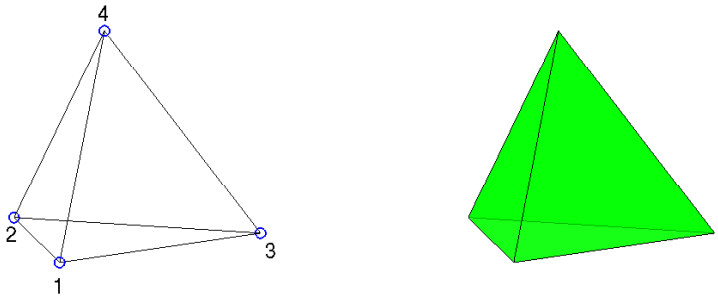
Antal kolonner i `H`, som ges av `size(H,2)`, är antalet hörnpunkter. Antalet rader i `S`, som ges av `size(S,1)`, är antalet sidor.

Med `for`-satsen ritar vi upp alla sidor. Vi plockar ut en sidas hörnpunkter med `Si=S(i,:)` och motsvarande kolonner i `H`, dvs. `H(:,Si)` ger koordinaterna för hörnpunkterna.

Vi ritar sidan med `fill3`, som fungerar som `fill` fast i rummet. Här måste vi separera x_1-, x_2- och x_3-koordinaterna. Med `H(1,Si)` får vi x_1-koordinaterna för hörnpunkterna på sidan, osv.

Vi får kuben lite lätt genomskinlig med `'facealpha',0.7`. För solid kub sätt `'facealpha',1` eller utelämna. Med `axis equal` får vi skalor på axlarna sådana att en kub ser ut som en kub och inte blir tillplattad. Vidare ger `axis tight` ett koordinatsystem *utan* luft runt kuben som vi ritat och `axis off` gör att axlarna och skalmarkeringar inte syns. Med `view(20,10)` ges betraktelsevinklar, se gärna hjälptexterna.

Uppgift 2. Rita en liksidig tetraeder.



Vi kan ta hörnpunkter på enhetssfären med hörnpunkternas koordinater som

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

Uppgift 3. Rotera tetraedern runt någon axel. Gör en translation av tetraedern bort från origo. Rotera den åter runt någon axel. Välj ett koordinatsystem så att tetraedern syns i alla lägen den hamnar i.

Uppgift 4. Nu skall vi göra om uppgift 3, fast som en *animation*. När ni roterade och translaterade tetraedern, så behöll ni i figuren resultatet efter varje rotation och translation. På så sätt fick ni en figur med många tetraedrar och det var bra för att förstå vad som hände. Nu vill vi bara se en enda tetraeder som roterar, blir translaterad för att sedan rotera igen. Rotationerna skall göras genom att upprepade gånger rotera med en liten vinkel så att vi får en jämn rörelse. För varje liten rotation skall vi sudda figuren (`clf`) och rita upp tetraedern i den senaste positionen samt lägga in en kort paus (`pause`) så att vi hinner se resultatet. På motsvarande sätt med translationen. Som hjälp ger vi följande början till ett script där vi ser på den första rotationen

```
ax=[...]; % Gränser för koordinataxlarna, ax=[xmin xmax ymin ymax zmin zmax].
          % Dessa sätts så att hela scenen får plats
v=...;    % Liten vinkel för upprepad rotation, t.ex. v=pi/8
A=[...];  % Rotationsmatrisen
P=H;      % H är matrisen med tetraederns koordinater
clf
hold on
for i=1:size(S,1)  % Rita tetraedern
    Si=S(i,:); fill3(P(1,Si),P(2,Si),P(3,Si),'g')
end
hold off
axis equal, axis(ax)  % Undvik deformation och använd samma skalor på axlarna
box on, grid on       % Förstärker tredimensionella känslan något
```

```
pause(0.5) % Paus så vi hinner se tetraedern innan den roteras
kmax=...;  % Antal små rotationer, t.ex. kmax=16; (kmax*v=2*pi, ett varv)
for k=1:kmax  % Rotera kmax gånger
    P=A*P;
    clf
    hold on
    for i=1:size(S,1)  % Rita tetraedern i nya positionen
        Si=S(i,:); fill3(P(1,Si),P(2,Si),P(3,Si),'g')
    end
    hold off
    axis equal, axis(ax), box on, grid on
    pause(0.1)
end
```

1 Målsättning

Avsikten med denna laborationen är att ni skall få träning i att använda linjära avbildningar för att transformera enkla geometrisk objekt i MATLAB.

2 Kommentarer och förklaringar

När vi roterar och translaterar i planet med MATLAB antar vi att vi redan har två radvektorer med koordinater för polygonområde. Vi placerar dem i en matris med `P=[X;Y]` så att koordinaterna för punkterna ligger som kolonner.

$$\mathbf{P} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix} = [\mathbf{P}_1 \ \ \mathbf{P}_2 \ \cdots \ \mathbf{P}_n], \text{ med } \mathbf{P}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

I MATLAB roterar vi med `P=A*P` så att varje punkt roteras $\hat{\mathbf{P}}_i = \mathbf{A}\mathbf{P}_i$, fast alla samtidigt.

$$\hat{\mathbf{P}} = \left[\hat{\mathbf{P}}_1 \ \ \hat{\mathbf{P}}_2 \ \cdots \ \hat{\mathbf{P}}_n\right] = [\mathbf{A}\mathbf{P}_1 \ \ \mathbf{A}\mathbf{P}_2 \ \cdots \ \mathbf{A}\mathbf{P}_n] = \mathbf{A}\,[\mathbf{P}_1 \ \ \mathbf{P}_2 \ \cdots \ \mathbf{P}_n] = \mathbf{A}\mathbf{P}$$

Vi translaterar i MATLAB med `P=P+t*ones(size(X))` så att varje punkt translateras $\hat{\mathbf{P}}_i = \mathbf{P}_i + \mathbf{t}$, fast alla samtidigt.

$$\begin{aligned} \hat{\mathbf{P}} &= \left[\hat{\mathbf{P}}_1 \ \ \hat{\mathbf{P}}_2 \ \cdots \ \hat{\mathbf{P}}_n\right] = [\mathbf{P}_1 + \mathbf{t} \ \ \mathbf{P}_2 + \mathbf{t} \ \cdots \ \mathbf{P}_n + \mathbf{t}] = \\ &= \mathbf{P} + [\mathbf{t} \ \ \mathbf{t} \ \cdots \ \mathbf{t}] = \mathbf{P} + \begin{bmatrix} t_1 & t_1 & \cdots & t_1 \\ t_2 & t_2 & \cdots & t_2 \end{bmatrix} = \mathbf{P} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \end{aligned}$$

Rotation och skalning är linjära avbildningar och kan beskrivas med standardmatriser $\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, däremot är translation *inte* en linjär avbildning utan en s.k. affin avbildning $\mathbf{F}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$.

En affin avbildning $\mathbf{F}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ kan beskrivas med matrismultiplikation om vi inför en extra koordinat w enligt

$$\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}$$

där $\mathbf{0}^\mathsf{T}$ är en rad med nollor.

Vi kommer då ha

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}w \\ \hat{w} &= w \end{aligned}$$

Så om låter vi $w = 1$ så får vi $\hat{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}$ och $\hat{w} = 1$.

Detta kallas *homogena koordinater* och man kan läsa om detta i Lay kap. 2.7.

Translation med en vektor $\mathbf{t}$ ges av

$$\mathbf{F}(\mathbf{x}) = \mathbf{x} + \mathbf{t}$$

Vi har $\mathbf{A} = \mathbf{I}$, dvs. enhetsmatrisen, och $\mathbf{b} = \mathbf{t}$, dvs. translationsvektorn, och vi kan beskriva translationen med matrismultiplikation

$$\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}$$

som vi också kan skriva

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{x} + \mathbf{t}w \\ \hat{w} &= w \end{aligned}$$

Tar vi $w = 1$ så får vi $\hat{\mathbf{x}} = \mathbf{x} + \mathbf{t}$.

Vill vi använda oss av homogena koordinater är det praktiskt att använda lika stora matriser även för linjära avbildningar.

En linjär avbildning $\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ kan bäddas in med homogena koordinater enligt

$$\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}$$

och vi ser att $\hat{\mathbf{x}} = \mathbf{A}\mathbf{x}$.

3 Lärandemål

Efter denna laboration skall du

- kunna rotera, skala och translatera i planet och rummet
- ha kännedom om homogena koordinater och hur de kan användas för att beskriva både linjära och affina avbildningar med matrismultiplikation