

## DATORÖVNING 2 — GEOMETRI

**Allmänt.** Dokumentera ditt arbete i ett pdf-dokument. Spara alla filer så att du kan läsa på inför tentamen. Det räcker inte att din kompis har filerna. Datorövningarna examineras vid tentamen. Samarbete uppmuntras, men detta är inget grupparbete. Varje student måste göra sina egna datorprogram och sina egna dokument.

### Mål.

Att lära känna de geometriska funktionerna: skalärprodukt, projektion, kryssprodukt.  
Att träna på begreppet funktion, både matematiskt och programmeringstekniskt.

**Litteratur.** Adams, kap. 10.1–4. Jönsson, kap. 7.

**Instruktioner.** Skriv funktionsfiler (m-filer) för de nämnda funktionerna. Dokumentera allt ditt arbete i ett word-dokument. Exportera slutligen word-dokumentet till pdf-format. Det görs med 'print' och skrivare 'PDF995'. Slutprodukten skall alltså vara pdf. Visa upp för din lärare som prickar av att du gjort övningen.

Skapa en ny filkatalog ("directory") `studio-2` för denna övning.

**Demonstration.** Läraren börjar med att demonstrera Uppgift 2 nedan.

### Uppgifter.

**1. Linjär kombination av vektorer.** En vektor  $\mathbf{v} \in \mathbf{R}^3$  kan skrivas antingen som radvektor eller kolonnvektor i MATLAB. Men man måste vara konsekvent och inte växla mellan rad- och kolonnrepresentation. Senare, i linjär algebra (Lay), ska vi se att kolonnvektor är bäst. Men i denna övning föreslår jag att du skriver vektorer som radvektorer som i Adams.

Skapa nu tre vektorer  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{w}$  och tre skalärer  $s, t, r$  och bilda linjärkombinationen  $s\mathbf{u} + t\mathbf{v} + r\mathbf{w}$ .

Skapa de tre basvektorerna  $\mathbf{i}$ ,  $\mathbf{j}$ ,  $\mathbf{k}$ . Eftersom MATLAB inte har fetstil och  $\mathbf{i}$  och  $\mathbf{j}$  redan är upptagna (av vad då?) föreslår jag att du kallar dem `ii`, `jj`, `kk`. Bilda sedan linjärkombinationen  $5\mathbf{i} - 2\mathbf{j} + 3\mathbf{k}$ .

**2. Skalär produkt.** Det är viktigt att förstå att skalärprodukten är en funktion som till varje par av vektorer  $(\mathbf{u}, \mathbf{v})$  ordnar en skalär  $s = \mathbf{u} \cdot \mathbf{v}$ . Dvs en funktion av typen

$$f : \mathbf{R}^3 \times \mathbf{R}^3 \rightarrow \mathbf{R}.$$

Definitionsmängden är alltså produktmängden  $\mathbf{R}^3 \times \mathbf{R}^3$  som består av alla par av vektorer, målmängden är  $\mathbf{R}$ , och funktionens regel är  $f(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \mathbf{v}$ .

Detta är inte minst viktigt när vi ska implementera skalärprodukten i MATLAB, dvs skriva ett datorprogram som beräknar skalärprodukten. Detta måste göras i form av en funktion, vi ger den namnet `prick`. Skriv nu en funktionsfil med namnet `prick.m` och med deklarationen (de första raderna i filen)

```
function s=prick(u,v)
% Scalar product.
%
% Syntax: s=prick(u,v)
%
% Input:   u,v - two 1x3 vectors
% Output:  s   - a scalar
```

Obs “help”-texten. Den ska vara med. Skriv resten av filen själv.

Hitta nu på några exempel för att testa programmet. Hitta på flera par av vektorer och beräkna skalärprodukten. Beräkna den även för hand som kontroll att programmet är korrekt. Se till att du får med följande fall:  $\mathbf{u} \cdot \mathbf{v}$  positiv, negativ och noll. Ge vektorerna olika namn så att du kan använda dem igen senare. Beräkna även längden (normen) av vektorerna genom  $|\mathbf{u}|^2 = \mathbf{u} \cdot \mathbf{u}$ . (Vid brist på fantasi: Adams övn **10.2** 2, 3.)

MATLAB har redan en funktion `dot` för skalärprodukt. Läs om den med kommandot

```
>> help dot          (>> är Matlabs prompt och ska inte skrivas)
>> help prick
```

och prova den på dina exempel.

MATLAB har även en funktion för längd av vektor, den heter `norm`. Förväxla inte med `length` och `abs`. Läs om `norm`, `length`, `abs` och prova dem.

Normera en av dina vektorer, dvs bilda  $\hat{\mathbf{u}} = \mathbf{u}/|\mathbf{u}|$ .

**3. Projektion.** Implementera nu **skalära projektionen** av vektorn  $\mathbf{u}$  på vektorn  $\mathbf{v}$  enligt Adams Definition 10.2.4. (Jag skriver inte ned formeln här för att du ska träna att använda boken.) Obs att denna är också en funktion av typen

$$f : \mathbf{R}^3 \times \mathbf{R}^3 \rightarrow \mathbf{R}.$$

Skriv en funktionsfil med namnet `sprojection.m` och med deklarationen

```
function s=sprojection(u,v)
% Scalar projection of the vector u along the vector v.
%
% Syntax: s=sprojection(u,v)
%
% Input:  u,v - two 1x3 vectors
% Output: s   - a scalar
```

Obs “help”-texten. Skriv resten av filen själv. Använd funktionerna `prick` och `norm` i programmet.

Implementera nu **vektorprojektionen** av vektorn  $\mathbf{u}$  på vektorn  $\mathbf{v}$ . Obs att denna är en funktion av typen

$$f : \mathbf{R}^3 \times \mathbf{R}^3 \rightarrow \mathbf{R}^3.$$

Skriv en funktionsfil med namnet `vprojection.m` och deklarationen

```
function w=vprojection(u,v)
```

Skriv resten av filen själv. Glöm inte “help”-texten! Den måste vara med.

Prova dina program på några exempel. Projicera vektorn  $\mathbf{u}$  på  $\mathbf{i}$ ,  $\mathbf{j}$  och  $\mathbf{k}$ . Lös några övningar från Adams som involverar projektion, t ex: Adams **10.2** 31, Exempel 10.4.7 sid 565 (utgå från projektion, använd aldrig formeln  $s = |Ax_0 + By_0 + Cz_0 - D|/\sqrt{A^2 + B^2 + C^2}$ , den är inte värd att memorera), **10.4** 26–30.

Använd vektorprojektion för att göra en ortogonal uppdelning av en vektor.

**4. Kryssprodukt.** (Om du hinner.) Implementera nu **kryssprodukten** av vektorerna  $\mathbf{w} = \mathbf{u} \times \mathbf{v}$ . Obs att denna är en funktion av typen

$$f : \mathbf{R}^3 \times \mathbf{R}^3 \rightarrow \mathbf{R}^3.$$

Skriv en funktionsfil med namnet `kryss.m` och deklarationen

```
function w=kryss(u,v)
```

Skriv resten av filen själv. Glöm inte “help”-texten! Den måste vara med.

Hitta på några exempel. Beräkna till exempel  $\mathbf{i} \times \mathbf{j}$ . Lös några övningar från Adams 10.3.

Obs att MATLAB har en funktion `cross`. Prova den.

**5. Trippelprodukt.** Använd dina program `prick` och `kryss` för att beräkna skalära trippelprodukten och beräkna volym i övning 10.3.14.

Alternativt (om du hoppade över Uppgift 4): använd MATLABs program `dot` och `cross` istället.  
`/stig`