

DATORÖVNING 2 — GEOMETRI

Allmänt. Dokumentera ditt arbete i ett pdf-dokument. Spara detta så att du kan läsa på inför duggor och tentamen. Detta ger träning i att skriva matematik. Det är viktigt i sig. Men du ska se att skrivandet också hjälper dig att lära matematik. Detta är ännu viktigare. När man arbetar med att förbättra resonemanget och texten ökar man samtidigt förståelsen.

Datorövningarna examineras endast vid duggor (2–3) samt vid tentamen. Detta sker genom ett flertal frågor av typen: “Hur gjorde du det och det i MATLAB i Datorövning 3?” “Skissa den graf som du gjorde då.” “Vad är syftet med denna beräkning?”

Datorövningarna är alltså inga vanliga “laborationer” som man gör och redovisar i datorsal. Det är material som du jobbar med, dokumenterar och lär in, på Chalmers och hemma. Samarbete uppmuntras, men detta är inget grupparbete. Varje student måste göra sina egna datorprogram och sina egna dokument.

Spara alla filer och textdokument i en separat filkatalog för varje övning. Om ni sitter två vid samma dator under den schemalagda datorövningstiden, tänk på att ni båda sparar filerna i er egen filkatalog så att ni kan läsa på inför duggor och tentamen, där material från datorövningarna kommer att utgöra en väsentlig del.

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.

Instruktioner. Skriv funktionsfiler (m-filer) för de nämnda funktionerna. Dokumentera allt ditt arbete i ett ordbehandlingsdokument. Exportera slutligen dokumentet till pdf-format. (Både MS Word och OpenOffice Writer stödjer detta.) Slutprodukten skall alltså vara pdf.

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

Uppgifter.

1. Linjärkombination 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 $s\mathbf{u} + t\mathbf{v} + r\mathbf{w}$.

Kom ihåg. Du skapar en radvektor så här i MATLAB:

```
>> u = [3 -2 -1]    % Radvektor.  
% >> är MATLABs prompt och ska inte skrivas.
```

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. Skalarprodukt. Det är viktigt att förstå att skalarprodukten 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, värdemängden 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 skalarprodukten i MATLAB, dvs skriva ett datorprogram som beräknar skalarprodukten. Detta måste göras i form av en funktion, vi ger den namnet `prick`. Funktionsfilen med namnet `prick.m` kan t.ex. se ut så här:

```
function s=prick(u,v)
% Scalar product.
%
% Syntax: s=prick(u,v)
%
% Input:   u,v - two 1x3 vectors
% Output:  s   - a scalar
s = u(1)*v(1)+u(2)*v(2)+u(3)*v(3);
```

Obs "help"-texten i deklarationen (de första raderna i filen) som visas om man skriver

```
>> help prick
```

Den ska vara med.

Hitta nu på några exempel för att testa programmet. Hitta på flera par av vektorer och beräkna skalarprodukten. 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 skalarprodukt. Läs om den med kommandot

```
>> help dot
```

```
>> help prick    % Jämför.
```

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 590 (sid 565, 6e uppl) (utgå hellre från projektion än att använda 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. (Detta kan göras även i **10.4** 28, även om Exempel 10.4.8 sid 590–591 (sid 565–566, 6e uppl) använder kryssprodukt. I detta exempel är $\overrightarrow{P_1P_0} = \overrightarrow{P_1P_2} + \overrightarrow{P_2P_0}$, där $\overrightarrow{P_1P_2}$ är vektorprojektion av $\overrightarrow{P_1P_0}$ på $\mathbf{v}$, och det sökta avståndet $s = |\overrightarrow{P_2P_0}|$.)

4. Kryssprodukt. (Om du hinner.) Implementera nu **kryssprodukten** $\mathbf{w} = \mathbf{u} \times \mathbf{v}$ av vektorerna $\mathbf{u}$ och $\mathbf{v}$. 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 10.3.14–15.

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