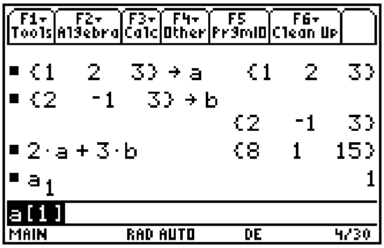
10 Vektorregning

Vektorer som lister

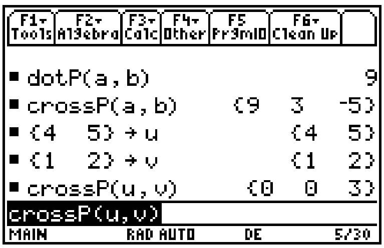
En vektor laves nemmest som en liste på TI-89 Titanium / Voyage 200. I nedenstående skærmbillede ser du, hvordan man definerer vektorer og laver en simpel udregning med dem. Husk at listens elementer adskilles med komma, altså {1,2,3}:



Som det ses, er det helt problemfrit at lægge vektorer sammen og gange dem med skalarer. Det er også ganske nemt at få adgang til koordinaterne i en vektor — fx er a[1] første koordinaten i a.

Mere problematisk er produktet. Man kan jo ikke gange to vektorer sammen, og forsøger man alligevel, så får man koordinaterne ganget sammen koordinat for koordinat — og det kan ikke bruges til noget her. Derimod er TI-89 Titanium / Voyage 200 udstyret med to kommandoer til udregning af prikprodukt og krydsprodukt. Kommandoerne hedder `DotP` og `CrossP`, hhv.

Tip:
Krydsproduktet er defineret i både 2 og 3 dimensioner.
I 2 dimensioner er krydsproduktet en 3-dimensional vektor, der peger op ad z-aksen.

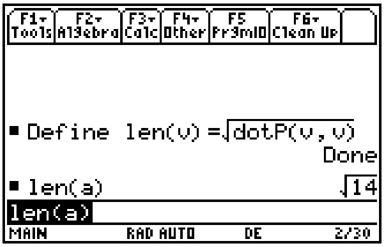


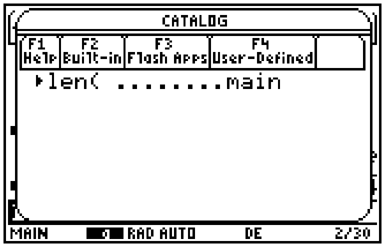
Dette er stort set, hvad TI-89 Titanium / Voyage 200 indeholder til vektorregning — resten må du selv lave.

Start med at definere længden af en vektor (`Define` skrives nemmest med [F4] 1):

```
Define len(v)=√(dotP(v,v))
```

Efter definitionen placeres `len` i `Catalog` under [F4] `User-Defined` (andet skærmbillede nedenfor):



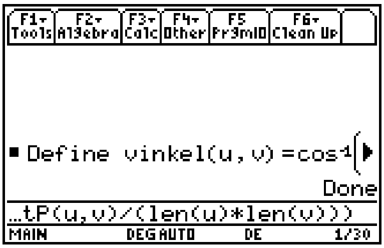


> **Tip**
> Ved at dividere med 1° er, at du får resultatet ud i grader, selvom indstillingen er `RADIAN`

> **Tip**
> Du kan benytte genvejstasterne:
> Voyage 200: [◆] F
> TI-89 Titanium: [◆][|]

Det kan også være nyttigt at have en kommando, der giver vinklen mellem to vektorer. Standardformlen er

```
Define vinkel(u,v)=
cos⁻¹(DotP(u,v)/(len(u)*len(v))/1°
```



Desværre er skærmen for lille til at vise denne definition. Værre bliver det, når projektionsformler, afstandsformler mv. skal indtastes. På side 100 får du en opskrift på, hvordan alle de gængse formler fra vektorregningen indtastes. Dette arbejde skal du kun gøre én gang. Herefter ligger de lagret i maskinens hukommelse.
Hvis du på et tidspunkt glemmer, hvordan en af formlerne er bygget op, så skal du ind i Var-Link for at undersøge dette. Se appendiks til dette afsnit, hvis du vil lave en skræddersyet menu til dine vektoroperationer.

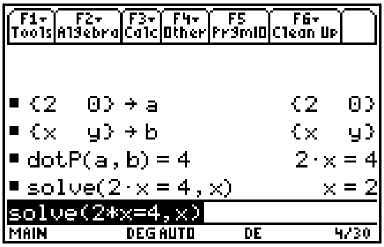
I en plan er givet to vektorer $\vec{a}$ og $\vec{b}$, hvor

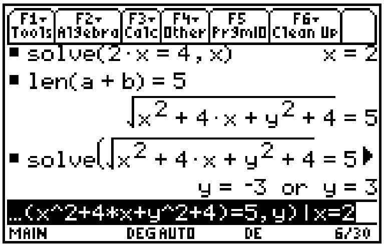
$$\left|\vec{a}\right| = 2, \quad \vec{a}\cdot\vec{b} = 4 \quad \text{og} \quad \left|\vec{a}+\vec{b}\right| = 2$$

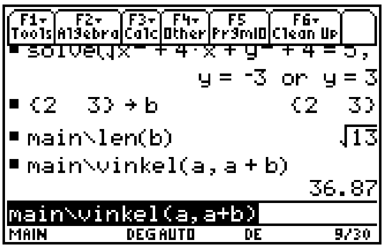
Find længden af $\vec{b}$

Bestem et gradtal for vinklen mellem $\vec{a}$ og $\vec{a}+\vec{b}$.

Opgaven regnes igennem med koordinater. Da koordinatsystemet kan vælges frit, kan kordinaterne til $\vec{a}$ vælges til at være {2,0}. Koordinaterne til $\vec{b}$ sættes til at være $\{x, y\}$, hvor x og y skal bestemmes ud fra de givne oplysninger.
Når først x og y er bestemt, er det en triviel sag at finde længden af $\vec{b}$ og vinklen mellem $\vec{a}$ og $\vec{a}+\vec{b}$.







TI-89 Titanium / Voyage 200 kommando	Formel
Længden af en vektor v `Define len(v)=√(dotP(v,v))`	$\sqrt{\vec{v}\cdot\vec{v}}$

Vinklen mellem to vektorer u og v
`Define vinkel(u,v)=`
`cos-1(DotP(u,v)/(len(u)*len(v))`
	$\cos^{-1}\left(\frac{\vec{u}\cdot\vec{v}}{\lvert\vec{u}\rvert\lvert\vec{v}\rvert}\right)$

Projektionen af en vektor u på en vektor v `Define proj(u,v)=DotP(u,v)/DotP(v,v)*v`	$\frac{\vec{u}\cdot\vec{v}}{\vec{v}\cdot\vec{v}}\cdot\vec{v}$
Afstand fra et punkt P til et punkt Q `Define dist(P,Q)=len(Q-P)`	$\lvert\overrightarrow{PQ}\rvert$
Afstand fra et punkt P til en plan p (med ankerpunkt P0 og normalvektor n) `Define distp(P,P0,n)=abs(dotP(n,P-P0))/len(n)`	$\frac{\lvert\vec{n}\cdot\overrightarrow{P_0P}\rvert}{\lvert\vec{n}\rvert}$

Afstand fra et punkt P til en linje *l*
(med ankerpunkt P0 og retningsvektor u)
`Define distl(P,P0,u)=`
`len(crossP(u,P-P0))/len(u)`
	$\frac{\lvert\vec{u}\times\overrightarrow{P_0P}\rvert}{\lvert\vec{u}\rvert}$

Afstand fra en linje l (med ankerpunkt P0 og retningsvektor u)
til en linje m (med ankerpunkt Q0 og retningsvektor v)
`Define distll(P0,u,Q0,v)=`
`abs(dotP(crossP(u,v),Q0-P0))/len(crossP(u,v))`
	$\frac{\lvert(\vec{u}\times\vec{v})\cdot\overrightarrow{P_0Q_0}\rvert}{\lvert\vec{u}\times\vec{v}\rvert}$

Areal af parallelogram udspændt af vektorerne u og v `Define areal(u,v)=len(crossP(u,v))`	$\lvert\vec{u}\times\vec{v}\rvert$

I et koordinatsystem i rummet er givet tre vektorer

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 2 \\ -1 \\ 2 \end{pmatrix}, \quad \vec{c} = \begin{pmatrix} -1 \\ 2 \\ 2 \end{pmatrix}$$

Bestem et gradtal for vinklen mellem $\vec{a}$ og $\vec{b}$.

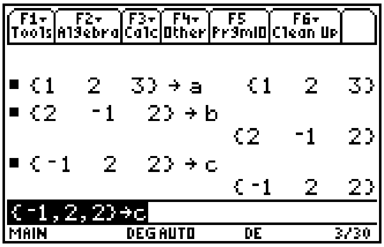
Bestem koordinatsættet til projektionen af $\vec{a}$ på $\vec{b}$.

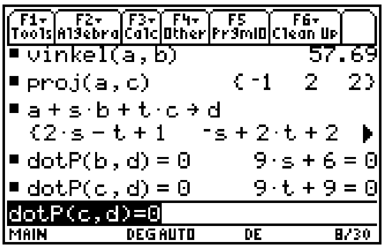
Bestem tallene s og t, således at vektoren

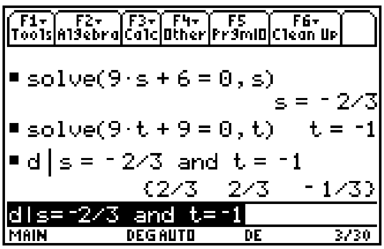
$$\vec{d} = \vec{a} + s\cdot\vec{b} + t\cdot\vec{c}$$

står vinkelret på både $\vec{b}$ og $\vec{c}$, og angiv koordinaterne for $\vec{d}$.

I nedenstående 3 skærmbilleder løses opgaven:







I et koordinatsystem i rummet er givet et punkt $P(5,4,3)$. To linjer *l* og *m* er bestemt ved:

$$l: \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 0 \\ 0 \end{pmatrix} + t \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad t \in R$$

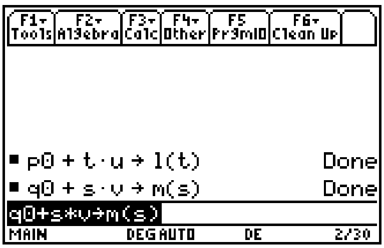
$$m: \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ -4 \\ 2 \end{pmatrix} + s \cdot \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \quad s \in R$$

Bestem en ligning for den plan α, der indeholder *P* og *l*.
Find koordinatsættet til *m*'s skæringspunkt med α.
Bestem et gradtal for den spidse vinkel, som *m* danner med α.
Bestem en parameterfremstilling for den linje, der går gennem *P* og skærer både *l* og *m*.

Først laves en række tildelinger (vises ikke på skærmbillederne nedenfor):

$\{5,4,3\} \rightarrow$ p
$\{8,0,0\} \rightarrow$ p0 $\qquad \{1,0,1\} \rightarrow$ u
$\{4,-4,2\} \rightarrow$ q0 $\qquad \{1,2,0\} \rightarrow$ v

Så kan de to parameterfremstillinger indtastes således:



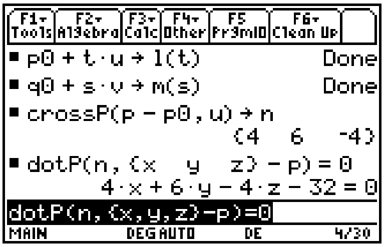
Ligning for den plan α, der indeholder *P* og *l* kan bestemmes således: Først finder du normalvektoren ved

$$\vec{n} = \overrightarrow{P_0P} \times \vec{u}$$

og ligningen for planen ved:

$$\alpha: \ \vec{n} \cdot \begin{pmatrix} x-5 \\ y-4 \\ z-3 \end{pmatrix} = 0$$

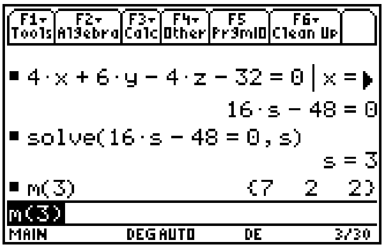
Dette laver du således på TI-89 Titanium / Voyage 200:

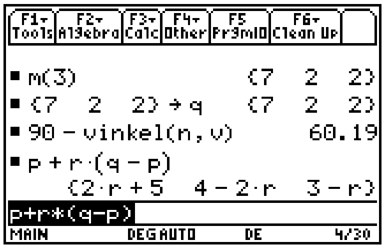


For at finde koordinatsættet til *m*'s skæringspunkt med α, skal du indsætte parameterfremstillingen for *m* i planens ligning. Du får fat i første koordinaten i *m*'s parameterfremstilling ved at skrive `m(s)[1]` og tilsvarende for de øvrige koordinater.
Gør således: Kopier planens ligning til indtastningslinjen og tilføjer

```
|x=m(s)[1] and y=m(s)[2] and z=m(s)[3]
```

efter ligningen. Så vil du få parameterfremstillingen for *m* indsat i planens ligning. Det en triviel sag at løse den ligning, der kommer frem:





Skæringspunktet mellem *m* og α er altså {7,2,2}. Dette punkt kaldes q.

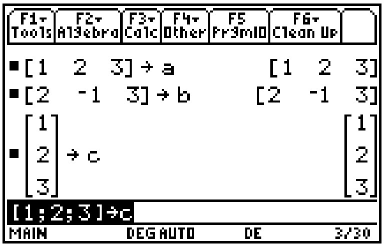
Vinklen, som *m* danner med α, findes let med den brugerdefinerede funktion `vinkel`.

Den linje, der går gennem *P* og skærer både *l* og *m*, må gå igennem *m*'s skæringspunkt med α — altså q. Dvs., du skal blot finde parameterfremstillingen for linjen gennem p og q. Se skærmbilledet til højre ovenfor, hvor r er benyttet som parameter.

Bemærkning:

Man kunne lige så vel have benyttet matricer til vektorer — enten som række-vektorer eller som søjlevektorer. Men der vindes ikke ret meget derved.

Rækkevektorer indtastes med kantede parenteser og komma som separator. Søjlevektorer indtastes også med kantede parenteser, men med semikolon som separator.



Addition og skalarmultiplikation fungerer som forventet, i kraft af at vektorerne her opfattes som matricer. Derimod fungerer multiplikation ikke, da spillereglerne for matrixmultiplikation ikke overholdes. Her er man altså også nødt til henvise til dotP for skalarprodukt og crossP for krydsprodukt.

Hvis en søjlevektor kopieres til indtastningslinjen fra historikområdet, vil søjlevektoren ikke vise sig med semikolon som separator, men på formen [[1] [2] [3]], som er standardindtastningen af en matrix.

Gevinsten ved at benytte matricer er, at så har vi automatisk længden af en vektor til rådighed, nemlig

`Norm( )`

Denne virker imidlertid kun på matricer — ikke på lister. Den norm, `len()`, vi har defineret, virker på såvel lister som på matrix-vektorer. Det samme gælder i øvrigt for alle de regneoperationer, der er beskrevet på side 100.

Endelig findes kommandoen

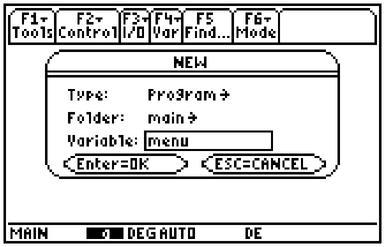
`UnitV( )`

der returnerer en enhedsvektor med samme retning som en given matrix-vektor. Skulle der opstå behov for denne i forbindelse med liste-vektorer, er det ingen sag at definere denne.

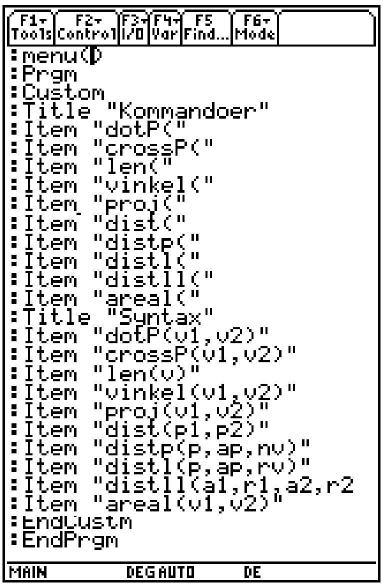
Appendiks — menu til vektorregning

Ved at skrive et simpelt program menu() kan du oprette din egen oversigt over kommandoer med tilhørende syntaks.

Du starter programeditoren fra skrivebordet eller med [APPS] `7`, hvis du ikke har skrivebordet slået til. Vælg `New..` og kald programmet `menu` (skærmbilledet til højre).

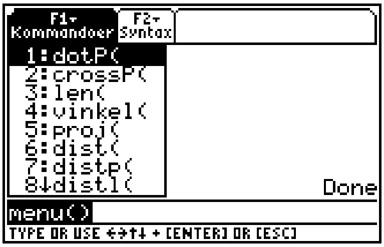


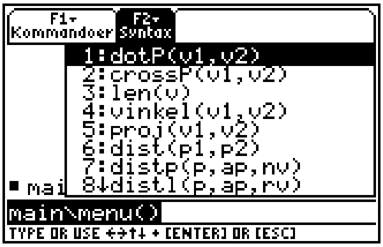
Afslut med [ENTER], og en næsten tom programeditor kommer frem. Denne udfylder du som vist nedenfor (en række sammenklippede skærmbilleder):



Start nu programmet `menu()` (husk parenteserne). Der sker tilsyneladende ingenting.

Tryk på [CUSTOM] , og straks kommer din nye menulinje frem. Tast [F1] for at se kommandoerne og [F2] for at se syntaksen:





Kommandoerne fra [F1] kan hentes til indtastningslinje på sædvanlig vis.

[CUSTOM]-tasten virker som en vippe-kontakt. Et tryk på [CUSTOM], og menuerne skifter til de brugerdefinerede menuer [F1] Kommandoer og [F2] Syntax. Endnu et tryk på [CUSTOM]-tasten, og de brugerdefinerede menuer fjernes igen.