

MATLAB - Flere laster om bord og automatisering fribordsberegning med `if else`

Oppgave 1

I denne oppgaven skal vi legge til flere laster på samme kasseformede geometri som er brukt i Oppgave 1, Øv1-Øv3, ($L=24$, $B=5$, $D=5$, $KG_0 = 1.5$ m). Vi kommer å editere dette Matlabskript, men husk og spare med nytt navn for å ha som sikkerhetskopi. Lekteren flyter uten trim og krengning i sjøvann før vi legger på lastene. Det plasseres flere kasseformede containere om bord med respektive last og plassering som gitt i Tabellen under. Koordinatene er gitt i skipets koordinatsystem. Kontainer 1, ligger på samme sted som i Øv1-3, Oppgave 1 og kan brukes for å verifisere svaren gjennom å sette alle andre laster til 0kg til å begynne med. Følg anvisningene som begynner på side 2 for hjelp i MATLAB.

Last type	W_i [tonn]	x_i [m]	y_i [m]	z_i [m]
Kontainer 1	8	10	-1	4
Kontainer 2	5	4	2	1
Kontainer 3	12	18	0.5	5

- Beregne ny tyngdepunkt, GM_L og GM_T med alle containere om bord.
(SVAR: $x_G=12.059$ m, $y_G=0.029$ m, $z_G=1.72$ m, $T=2.203$ m, $GM_T=0.328$ m, $GM_L=21.168$ m)
- Beregne krengevinkel, trimvinkel og minste fribord.
(SVAR: $\phi = -5.148$ deg (mot babord), $\theta = 0.16$ deg (baug ned), $f_{\min}=2.538$ m ved babord baug)
- Automatisere minste fribordberegning ved bruk av `if else` i MATLAB og skriv ut vad og hvor minste fribord er (hekk, baug, styrbord, babord?). Sett alle massene for Kontainer 2 og 3 til 0 og verifiser ved å flytte Kontainer 1 at tegn på vinkler og at automatiseringen blir rett.
- Anta at du har veldig mange containere. Då er det vanlig at importere data fra en text-fil. Følg anvisningene for å importere dataen fra tabellen fra filen istedenfor å ha de i m-skriptet.

Anvisninger MATLAB

a-c) Begynn med å lagre skriptet fra utregningen i Øving 3, Oppgave 1 med nytt navn, f.eks.: Ov4. Eller bruk ferdig skript Ov3_oppg1 på It's Learning, kopier og lagre det med nytt navn. Fjern siste oppgaven med lasteflytt og kjør programmet (F5) for at se at alt virker.

Nå begynner oppgaven. Erstatt W_k , x_{Gk} , y_{Gk} og z_{Gk} med vektorer som inneholder verdiene for Kontainer 1 og sett foreløpig 0 for de andre verdiene frem til vi har verifisert resultatene mot de fra Øv 3:

```
Wk=[8e3 0 0]; % [kg] Vekt kontainer
xGk=[10 0 0]; % [m] Tyngdepunkt kontainer x koordinat
yGk=[-1 0 0]; % [m] Tyngdepunkt kontainer y koordinat
zGk=[4 0 0]; % [m] Tyngdepunkt kontainer z koordinat
```

Lengre ned i skriptet, etter at du har regnet ut lettskipsdeplasementet og lettskipstygdepunkt, legg inn lettskips tyngdepunkt og lettskipsvekt i ny vektor som inneholder alle masser og tyngdepunkter:

```
mv=[delta0 Wk]; % [kg] masse vektor inkludert lettskipsvekt
xgv=[xG0 xGk]; % [m] x-tyngdepunkt vektor for alle masser
ygv=[yG0 yGk]; % [m] y-tyngdepunkt vektor for alle masser
zgv=[zG0 zGk]; % [m] z-tyngdepunkt vektor for alle masser
```

Avsnittet `% Total tyngdepunkt` skal nå erstattes. Husk at elementvis multiplikasjon kan gjøres med `.*` for å få momentene. Også at tyngdepunktsatsen sier $\Sigma(\text{momenter})/\Sigma(\text{masser})$.

Summe forkortes med innebygd kommando `sum` i Matlab. Erstatt nå `delta`, `xG`, `yG` og `zG` med:

```
delta=sum(mv);
xG= sum(mv.*xgv)/delta;
yG= sum(mv.*ygv)/delta;
zG= sum(mv.*zgv)/delta;
```

Når dette er gjort, erstatt ny dypgang med å bruke summen av alle tilførte vektorer i vektoren W_k :
`deltaT=sum(Wk)/(B*L*rho);`

Kjør programmet og se at $f_{\min}$ blir lik som i Øv 3.

Før vi setter inn alle lastene, skal vi **automatisere** minste fribord beregningen. Hvis vi setter lasten lengre forover og baugen går ned (positiv trimvinkel), må ligningen for minste fribord endres, se Forelesningsnotater_GS.pdf. Vi vil og at programmet sier ifra at minste fribord da er ved baugen. Hvis vi flytter lasten sidledes, vil vi også at programmet skal si ifra at minste fribord er på babord eller styrbord side.

Vi skal begynne med å legge til et varsel for stabilitet, for å illustrere bruk av `if`-kommandoen, dvs «hvis». Hvis GM_T er mindre en null, skal vi skrive ut i *Command Window* at skipet er ustabil:

```
if GMT<0
    fprintf(1,'WARNING!!! Ship is unstable, GM = %.3f m. \n', GMT)
end
```

Man må alltid avslutte en `if`-sats med et `end` og det er standard at man øker innrykket innenfor en if-sats sånn at det er enkelt å se vad som tilhører if-satsen. Man ser at if automatisk blir blå farge, og vi vil se videre i kursen at all slik kode som blir blå, trenger et end. Sett z_{Gk} verdien for Kontainer 1 til 20 meter temporært og kjør programmet. Man ser at varselteksten skrives ut, samt at verdien på GMT skrives ut med tre desimaler. Notasjonen for dette er at verdien GMT som er skrevet sist i parentes skrives ut etter `%` tegnet der `.3f` står for *floating point* med tre desimaler. Du kan lese mer om andre sett å skrive ut verdien på under `>> doc fprintf`. `fprintf` er en kommando som seinere kan brukes for å skrive verdier til en tekstfil. Då kan man sette *field width* foran punktet `%8.3f`. Då tar verdien åtte plasser horisontalt i tekstfilen, 6 plasser

for selve verdien -0.041 og to plasser med tomrom etter. \n betyr at man skal skifte til ny rad etterpå i *Command Window*.

Nå skal vi automatisere fribord beregningen. Gå lengre ned i koden der du har regnet ut phi og theta. Dobbeltsjekk at krengevinkelen blir positiv når du har last på styrbord (-y) side og at trimvinkelen blir positiv når du har last foran midten på skipet, dvs endre litt frem og tilbake på x_{Gk} og y_{Gk} for Kontainer 1 for å verifisere dine svar. Nå vil vi at det både brukes ulike formler og at det skrives ut hvor minste fribord er. Vi vet at endringen i dypgang pga trim er $x_F \tan(\theta)$ hvis trimvinkelen er negativ (hekk ned). Og at endringen i dypgang er $(L - x_F) \tan(\theta)$ når det er positiv trim (baug ned). Vi velger at verdien på dypgangsendringen skal endres for disse to alternativer, for å unngå repetisjon av samme ligning for minste fribord, velger vi at bare oppdatere dypgangsendringen og skrive ligningen for minste fribord etterpå. Følgende syntax brukes:

```
if theta<0
    dT_trim=xF*tand(abs(theta));
else %theta>0
    dT_trim=(L-xF)*tand(abs(theta));
end
fmin=D-T-B/2*tand(abs(phi))-dT_trim;
```

Det skal og skrives ut i *Command Window* at baug eller hekk går ned, så legg til i koden oven:

```
if theta<0
    dT_trim=xF*tand(abs(theta));
    fprintf(1,'Minimum freeboard at stern. \n ')
else %theta>0
    dT_trim=(L-xF)*tand(abs(theta));
    fprintf(1,'Minimum freeboard at bow. \n ')
end
```

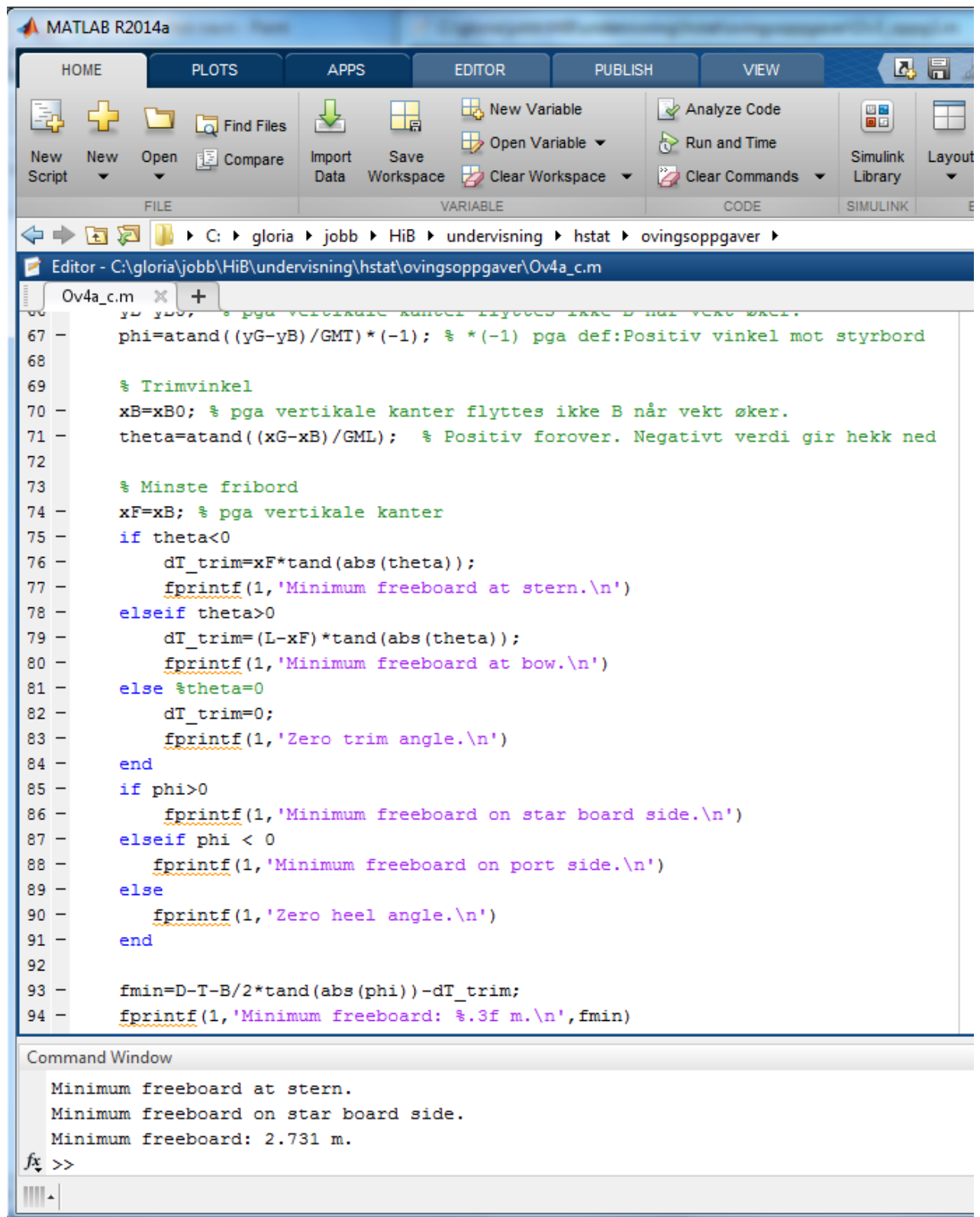
Når vi ser på hvis-satsen, så er alternativet at $\theta=0$, ikke er inkludert. For å unngå feilmelding når vi setter vekten eksakt på $L/2$, som ikke gir trim, så legger vi til en til hvis-sats med denne syntaxen *if-elseif-else-end*:

```
if theta<0
    dT_trim=xF*tand(abs(theta));
    fprintf(1,'Minimum freeboard at stern.\n ')
elseif theta>0
    dT_trim=(L-xF)*tand(abs(theta));
    fprintf(1,'Minimum freeboard at bow. \n ')
else %theta=0
    dT_trim=0;
    fprintf(1,'Zero trim angle.\n')
end
```

Nå skal vi også skrive ut hvis minste fribord er på babord eller styrbord side, eller hvis krengevinkeln er null, legg till:

```
if phi>0
    fprintf(1,'Minimum freeboard on star board side.\n')
elseif phi < 0
    fprintf(1,'Minimum freeboard on port side.\n\n')
else
    fprintf(1,'Zero heel angle.\n\n')
end
```

Koden skal nå se ca slik ut etter at phi og theta har regnets ut og resultatene i Command Window vises fortsatt bare med Kontainer 1 om bord. Verifier koden med å flytte Kontainer 1 forover og mot babord, og kjør programmet for å se at rett tekst skrives ut i *Command Window*.



The screenshot shows the MATLAB R2014a interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. Below the menu bar is a toolbar with icons for New Script, New, Open, Compare, Import Data, Save Workspace, Open Variable, Clear Workspace, Analyze Code, Run and Time, Clear Commands, Simulink Library, and Layout. The current file path is C:\gloria\jobb\HiB\undervisning\hstat\ovingsoppgaver\Ov4a_c.m. The script editor displays the following code:

```

67 - phi=atand((yG-yB)/GMT)*(-1); % *(-1) pga def:Positiv vinkel mot styrbord
68
69 % Trimvinkel
70 - xB=xB0; % pga vertikale kanter flyttes ikke B når vekt øker.
71 - theta=atand((xG-xB)/GML); % Positiv forover. Negativt verdi gir hekk ned
72
73 % Minste fribord
74 - xF=xB; % pga vertikale kanter
75 - if theta<0
76 -     dT_trim=xF*tand(abs(theta));
77 -     fprintf(1,'Minimum freeboard at stern.\n')
78 - elseif theta>0
79 -     dT_trim=(L-xF)*tand(abs(theta));
80 -     fprintf(1,'Minimum freeboard at bow.\n')
81 - else %theta=0
82 -     dT_trim=0;
83 -     fprintf(1,'Zero trim angle.\n')
84 - end
85 - if phi>0
86 -     fprintf(1,'Minimum freeboard on star board side.\n')
87 - elseif phi < 0
88 -     fprintf(1,'Minimum freeboard on port side.\n')
89 - else
90 -     fprintf(1,'Zero heel angle.\n')
91 - end
92
93 - fmin=D-T-B/2*tand(abs(phi))-dT_trim;
94 - fprintf(1,'Minimum freeboard: %.3f m.\n',fmin)

```

The Command Window at the bottom shows the output of the script:

```

Minimum freeboard at stern.
Minimum freeboard on star board side.
Minimum freeboard: 2.731 m.
fx >>

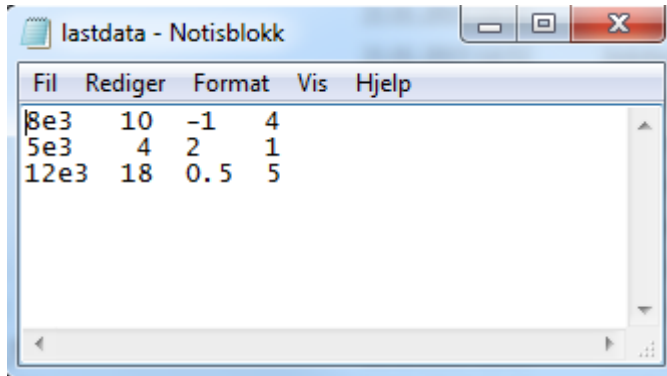
```

Legg till de andre lastene og koordinatene for Kontainer 2 og 3 der du tidligere har skrevet 0. Resultatet i Command Window skal vise:
 Minimum freeboard at bow.
 Minimum freeboard on port side.
 Minimum freeboard: 2.538 m.

d)

Anta att vi har mange laster og vil importere data slik den er gitt i Tabellen, fra en datafil. Spare ditt skript som for eksempel Ov4d.m.

Kopier filen: lastdata.txt til samme mappe som ditt m-skript eller lag en egen i Notepad, eller last ned en bedre versjon fra nett som heter Notepad++. Skriv inn data fra tabellen slik og lagre som .txt-fil:



I ditt skript, gå til der du har dine verdier for W_k , x_{Gk} , y_{Gk} og z_{Gk} . Før disse, skriv:

```
load lastdata.txt
```

«lastdata» vil nå være navn på en variabel for tabell-matrisen. I *Command Window*, skriv lastdata for å se dette:

```
lastdata =  
      8000      10      -1      4  
      5000       4       2      1  
     12000     18      0.5      5
```

Nå, fjern tidligere verdi-vektorer for W_k , x_{Gk} , y_{Gk} og z_{Gk} og erstatt definisjonen som første, andre tredje og fjerde kolonnen i matrisen lastdata ved å plukke ut disse kolonner:

```
Wk=lastdata(:,1);  
xGk=lastdata(:,2);  
yGk=lastdata(:,3);  
zGk=lastdata(:,4);
```

Tidligere hadde vi skrivit verdiene som rad-vektorer, og nå er de skrevne som kolonner. Dette må man holde kontroll på, ettersom vi «setter i hop» (concatenate) massevektor og koordinatvektorer radvis lengre ned i koden. Derfor må vi transponere W_k , x_{Gk} , y_{Gk} og z_{Gk} og legge til ' etter hver definisjon:

```
Wk=lastdata(:,1)'; % Transponert fra kolonn- til rad-vektor  
xGk=lastdata(:,2)';  
yGk=lastdata(:,3)';  
zGk=lastdata(:,4)';
```

Nå skal koden virke på samme vis som tidligere. Hvis vi ikke hadde transponert våre definisjoner hadde denne feilmeldingen kommet opp i *Command Window*:

```
Error using horzcat  
Dimensions of matrices being concatenated are not consistent.  
Error in Ov4d (line 37)  
mv=[delta0 Wk]; % [kg] masse vektor  
>>
```