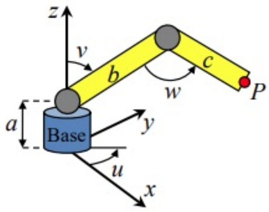
OPGAVE 1

Figur 1 illustrerer en toleddet robotarm placeret i et *xyz*-koordinatsystem. Robotarmen er monteret på en base, der kan dreje om den lodrette *z*-akse og dermed svinge robotarmen rundt. Svingvinklen i forhold til *x*-aksen benævnes *u*. Robotarmen er monteret i en højde *a* [m] over *xy*-planen, og dets inderste og yderste led har henholdsvis længderne *b* og *c* [m]. Det inderste leds drejningsvinkel i forhold til *z*-aksen benævnes *v*, mens vinklen mellem det inderste og yderste led benævnes *w*. Det yderste punkt på yderste led betegnes *P* med koordinaterne (*x*, *y*, *z*) [m].



Figur 1

(a)

Der defineres nu tre såkaldte rotationsmatricer, **U**, **V** og **W**, samt tre vektorer, **A**, **B** og **C**:

$$\mathbf{U}=\begin{bmatrix}\cos u & -\sin u & 0\\ \sin u & \cos u & 0\\ 0 & 0 & 1\end{bmatrix}\quad \mathbf{V}=\begin{bmatrix}\cos v & 0 & \sin v\\ 0 & 1 & 0\\ -\sin v & 0 & \cos v\end{bmatrix}\quad \mathbf{W}=\begin{bmatrix}\cos w & 0 & -\sin w\\ 0 & 1 & 0\\ \sin w & 0 & \cos w\end{bmatrix}$$

$$\mathbf{A}=\begin{bmatrix}0\\0\\a\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0\\0\\b\end{bmatrix}\quad \mathbf{C}=\begin{bmatrix}0\\0\\-c\end{bmatrix}$$

På grundlag heraf kan punktet *P*'s koordinater udregnes som:

$$\mathbf{P}=\mathbf{U}\cdot(\mathbf{V}\cdot(\mathbf{W}\cdot\mathbf{C}+\mathbf{B})+\mathbf{A})$$

hvor **P** er en søjlevektor, der indeholder *x*, *y* og *z*, og hvor prik (·) betegner matrixmultiplikation.

Lav et MATLAB-script, der kan udregne **P**, når $a = 1$ m, $b = 3$ m, $c = 2$ m, $u = 30°$, $v = 60°$ og $w = 90°$. Scriptets første linjer skal være:

```
a = 1; b = 3; c = 2; % m
u = 30*pi/180;  % rad
v = 60*pi/180;  % rad
w = 90*pi/180;  % rad
```

Besvarelsen skal indeholde det færdiggjorte script og den opnåede **P**-vektor efter kørsel af scriptet.

```
a = 1; b = 3; c = 2; % m
u = 30*pi/180; % rad
v = 60*pi/180; % rad
w = 90*pi/180; % rad

%Matricerne opskrives
U = [cos(u) -sin(u) 0
    sin(u) cos(u) 0
    0 0 1];
V = [cos(v) 0 sin(v)
    0 1 0
    -sin(v) 0 cos(v)];
W = [cos(w) 0 -sin(w)
    0 1 0
    sin(w) 0 cos(w)];

%Vektorende opskrives
A = [0; 0; a];
B = [0; 0; b];
C = [0; 0; -c];

%Der udregnes hvor resultatet bliver en vektor
P = U * (V * (W * C + B) + A) %Rækkefølgen af matricerne er ikke ligegyldig
```

```
P = 3×1
      3.116
      1.799
    0.76795
```

(b)

Det antages fortsat, at $a = 1$ m, $b = 3$ m og $c = 2$ m. Men nu ønskes det bestemt, hvad vinklerne *u*, *v* og *w* skal være, for at punktet *P*'s koordinater bliver $x = 1{,}1$ m, $y = 2{,}8$ m og $z = 0{,}6$ m. Med baggrund i matrixformlen i spørgsmål (a) kan det vises, at *u*, *v* og *w* kan bestemmes som løsning til følgende ulineære ligningssystem:

$$\cos(u)\cdot(b\sin(v)-c\sin(v-w))=x$$
$$\sin(u)\cdot(b\sin(v)-c\sin(v-w))=y$$
$$a+b\cos(v)-c\cos(v-w)=z$$

Ligningssystemet ønskes løst vha. Newton-Raphsons metode, idet der som startværdier af u, v og w anvendes henholdsvis 45°, 30° og 55°, og idet der ønskes en nøjagtighed på 0,0001%, dog højest 20 iterationer. Her følger et delvist færdigt MATLAB-script, der kan løse ligningssystemet på baggrund af disse betingelser:

```
a = 1; b = 3; c = 2;  % m
x = 1.1; y = 2.8; z = 0.6;  % m
u0 = 45*pi/180;  % rad
v0 = 30*pi/180;  % rad
w0 = 55*pi/180;  % rad
es = 0.0001; maxit = 20; % Ønsket nøjagtighed (%) og maks iterationer
X = [u0; v0; w0];
iter = 0;
while 1
  u = X(1); v = X(2); w = X(3);
  f = [cos(u)*(b*sin(v)-c*sin(v-w)) - x
       sin(u)*(b*sin(v)-c*sin(v-w)) - <UDFYLD>
       a + b*cos(v)-c*cos(v-w) - z];
  J = [-sin(u)*(b*sin(v)-c*sin(v-w)) cos(u)*(b*cos(v)-c*cos(v-w)) c*cos(v-w)*cos(u)
        cos(u)*(b*sin(v)-c*sin(v-w)) sin(u)*(b*cos(v)-c*cos(v-w))      <UDFYLD>
                     0                          <UDFYLD>                -c*sin(v-w)    ];
  dX = J\f;
  X = X - <UDFYLD>;
  iter = iter + <UDFYLD>;
  ea = 100*max(abs(dX./X));
  if iter>=<UDFYLD> || ea<=<UDFYLD>, break, end
end
u = X(1)*180/pi % u i grader
v = X(2)*180/pi % v i grader
w = X(3)*180/pi % w i grader
ea
iter
```

Scriptet kan downloades fra Blackboard under navnet `NewtRaphRobot.m`.

> Færdiggør scriptet ved at erstatte tekstangivelserne **<UDFYLD>** med korrekt MATLAB-kode. Besvarelsen skal indeholde både det færdiggjorte script og det output i kommandovinduet, som scriptet giver anledning til, når det køres.

```
clear
syms u v w x y z a b c
f = [cos(u)*(b*sin(v)-c*sin(v-w)) - x
        sin(u)*(b*sin(v)-c*sin(v-w)) - y
        a + b*cos(v)-c*cos(v-w) - z];
jacobian(f, [u,v,w])
```

ans =

$$\begin{pmatrix} -\sin(u)\,\sigma_2 & \cos(u)\,\sigma_1 & c\cos(v-w)\cos(u) \\ \cos(u)\,\sigma_2 & \sin(u)\,\sigma_1 & c\cos(v-w)\sin(u) \\ 0 & \sigma_3 - b\sin(v) & -\sigma_3 \end{pmatrix}$$

where

$$\sigma_1 = b\cos(v) - c\cos(v-w)$$

$$\sigma_2 = b\sin(v) - \sigma_3$$

$$\sigma_3 = c\sin(v-w)$$

```
a = 1; b = 3; c = 2; % m
x = 1.1; y = 2.8; z = 0.6; % m
u0 = 45*pi/180; % rad
v0 = 30*pi/180; % rad
w0 = 55*pi/180; % rad
es = 0.0001; maxit = 20; % Ønsket nøjagtighed (%) og maks iterationer
X = [u0; v0; w0];
iter = 0;
while 1
    u = X(1); v = X(2); w = X(3);

    f = [cos(u)*(b*sin(v)-c*sin(v-w)) - x

        sin(u)*(b*sin(v)-c*sin(v-w)) - y

        a + b*cos(v)-c*cos(v-w) - z];

    J = [-sin(u)*(b*sin(v)-c*sin(v-w)) cos(u)*(b*cos(v)-c*cos(v-w)) c*cos(v-w)*cos(u)
        cos(u)*(b*sin(v)-c*sin(v-w)) sin(u)*(b*cos(v)-c*cos(v-w)) c*cos(v-w)*sin(u)
        0 c*sin(v-w)-b*sin(v) -c*sin(v-w)];

    dX = J\f;

    X = X - dX;
```

```
    iter = iter + 1;

    ea = 100*max(abs(dX./X));


    if iter>=20 || ea<=es
        break
    end
end
u = X(1)*180/pi % u i grader
```

```
u = 
      68.552
```

```
v = X(2)*180/pi % v i grader
```

```
v = 
       58.87
```

```
w = X(3)*180/pi % w i grader
```

```
w = 
      71.589
```

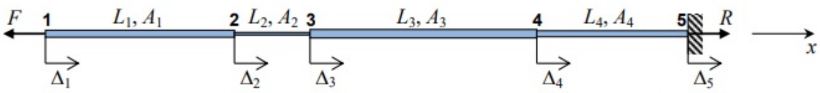
```
ea
```

```
ea = 
  4.5887e-05
```

```
iter
```

```
iter = 
     5
```

OPGAVE 4



Figur 2: Sammenføjede stangelementer påvirket af trækkraft

I figur 2 er vist en konstruktion bestående af fire sammenføjede stålstænger (stangelementer) med E-modul på $E = 210$ GPa og forskellige længder ($L_1, \ldots, L_4$) og tværsnitsarealer ($A_1, \ldots, A_4$). Konstruktionen er i knude **1** påvirket af en kraft F mod venstre og er i den fikserede knude **5** påvirket af en lige så stor og modsatrettet reaktionskraft, R. Kraftpåvirkningen bevirker at knude **1** til **4** udsættes for vandrette forskydninger ($\Delta_1, \ldots, \Delta_4$), mens knude **5** ikke forskydes ($\Delta_5 = 0$ mm). Der regnes positiv retning mod højre.

Der er givet følgende værdier af kraft og dimensioner:

$F = 1244$ N, $E = 210$ GPa
$L_1 = 250$ mm, $L_2 = 100$ mm, $L_3 = 300$ mm, $L_4 = 200$ mm
$A_1 = 65$ mm^2, $A_2 = 8$ mm^2, $A_3 = 102$ mm^2 og $A_4 = 47$ mm^2

Følgende ufærdige MATLAB-script er beregnet til at udregne forskydningerne $\Delta_1, \ldots, \Delta_4$ i form af en vektor D:

```
clear
L = [250 100 300 200]*1e-3; % m     Længder af stangelementerne
A = [65 8 102 47]*1e-6;     % m^2   Tværsnitsarealer af stangelementerne
E = 210e9;                  % Pa    E-modul for stangelementerne
k = A*E./L;                 % N/m   Fjederkonstanter for stangelementerne
K = [ k(1)     -k(1)        0          0         0    % N/m  Stivhedsmatrix
     -k(1)    <UDFYLD>    -k(2)        0         0
       0       -k(2)     <UDFYLD>    -k(3)       0
       0         0        -k(3)     <UDFYLD>  -k(4)
       0         0          0        -k(4)     k(4)];
Kred = K(<UDFYLD>,<UDFYLD>);
Fvek = [<UDFYLD>; 0; 0; <UDFYLD>];
D = Kred\Fvek
```

MATLAB-scriptet kan downloades fra Blackboard under navnet `FEMstangelementer.m`.

(a)

Færdiggør MATLAB-scriptet ved at erstatte tekstangivelserne **<UDFYLD>** med korrekt MATLAB-kode, og kør herefter scriptet. I besvarelsen ønskes såvel det færdiggjorte script som de beregnede forskydninger, Δ_1, Δ_2, Δ_3, og Δ_4 anført.

```
clear
L = [250 100 300 200]*1e-3; % m Længder af stangelementerne
A = [65 8 102 47]*1e-6; % m^2 Tværsnitsarealer af stangelementerne
E = 210e9; % Pa E-modul for stangelementerne
k = A*E./L; % N/m Fjederkonstanter for stangelementerne
K = [ k(1) -k(1) 0 0 0 % N/m Stivhedsmatrix
    -k(1) k(1)+k(2) -k(2) 0 0
    0 -k(2) k(2)+k(3) -k(3) 0
    0 0 -k(3) k(3)+k(4) -k(4)
    0 0 0 -k(4) k(4)];

Kred = K(1:4,1:4);
F = 1244; %N
Fvek = [F; 0; 0; F];
D = Kred\Fvek;

format("shortG")
tab = table(D, VariableNames={'Forskydning i punkt'})
```

tab = 4×1 table

	Forskydning i punkt
1	0.00016467
2	0.00014189
3	6.7838e-05
4	5.0415e-05

(b)

Kraftpåvirkningen forårsager forlængelser af de fire stangelementer. Beregn forlængelsen af hvert stangelement i mm og i procent. Hvis spørgsmål (a) ikke er besvaret, kan man antage, at resultatet i spørgsmål 4a blev

D = [-0.1245e-3; -0.1008e-3; -0.0623e-3; -0.0240e-3].

```
%flipper vectorne
D_lang = flip(D);
L_rev = flip(L);

%start værdiger til forloop
d = 0;
forlaengelse = [];
d_proc = [];

%forloop udregner forlængelserne
for i = 1:length(L)
     d = D_lang(i) - d;
     forlaengelse(i) = d * 1000;
     d_proc(i) = d/L_rev(i) * 100;
end

%tabel for klarhed
clear table
tab_delta = table(flip(forlaengelse'), flip(d_proc'), VariableNames={'Forlængelse i mm', 'Forlængelse i %'})
```

tab_delta = 4×2 table

	Forlængelse i mm	Forlængelse i %
1	0.040207	0.016083
2	0.12446	0.12446
3	0.017423	0.0058077
4	0.050415	0.025208

OPGAVE 5

Lad der være givet følgende ligningssystem, hvor a, b, c, d og k er konstanter, mens u og v er ubekendte:

$$\frac{\ln a + bu}{\sin c} + \left(3v + u\sqrt{d} - k^2\right)\sin c - 1 = 0$$

$$6 + \frac{4}{b+d} + \frac{\left(\cos c - \frac{3a}{b}\right)(2u - v\sin c + k)}{b - 2a^2} = 0$$

(5a)

Vis at ligningssystemet er lineært ud fra en bestemmelse og vurdering af relevante partielt afledte af ligningernes venstresider. De partielt afledte og argumentationen for linearitet skal fremgå af besvarelsen.

```
%% Konstanter sættes til 1 og u og v sættes til symboler, der evalueres for at få en ligning som gerne skulle være linær.
syms u v
a = 1;
b = 1;
c = 1;
d = 1;
k = 1;
func1 = (log(a)+b*u)/sin(c)+(3*v + u * sqrt(d)-k^2)*sin(c)-1
```

func1 =

$$\frac{3464384470253401692409223818 7345\,u}{1706705918346190893412923945 7792} + \frac{11368945240871781\,v}{4503599627370496} - \frac{8293248040994423}{4503599627370496}$$

```
%%Vi kan se at formen af funktione er linær, for at evaluere det så differentieres den, vis den er linær burde vi gerne kun have 1 hældning.

vpa(diff(func1, u), 4)
```

ans = 2.03

```
vpa(diff(func1, v), 4)
```

ans = 2.524

```
%% vi har kun 1 hældning derfor må funktion 1 være linær, dette gøres igen for function 2.

func2=6+4/(b+d)+(cos(c)-3*a/b)*(2*u-v*sin(c)+k)/(b-2*a^2)
```

func2 =

$$\frac{5538746809368157\,u}{1125899906842624} - \frac{20989903059586623388313487092539\,v}{10141204801825835211973625643008} + \frac{23553145318850141}{2251799813685248}$$

```
vpa(diff(func2, u), 4)
```

ans = 4.919

```
vpa(diff(func2, v), 4)
```

ans = -2.07

(5b)

> Benyt matrixberegning i MATLAB til at løse ligningssystemet, når $a = 1$, $b = 3$, $c = \pi/2$, $d = 1$ og $k = 2$. Besvarelsen skal indeholde de anvendte MATLAB-kommandoer og de fundne løsningsværdier af u og v.

```
a = 1;
b = 3;
c = pi/2;
d = 1;
k = 2;


eqs = [func1 == 0; func2 == 0]; %definer en simpel 2x1 matrice
[A,B] = equationsToMatrix(eqs); %Bruger indbygget function til at omdanne 2x1 matrice til et matrice lignigns system.
X = vpa(linsolve(A,B),5) %solver for matricen
```

X =

$$\begin{pmatrix} -1.3594 \\ 1.8226 \end{pmatrix}$$