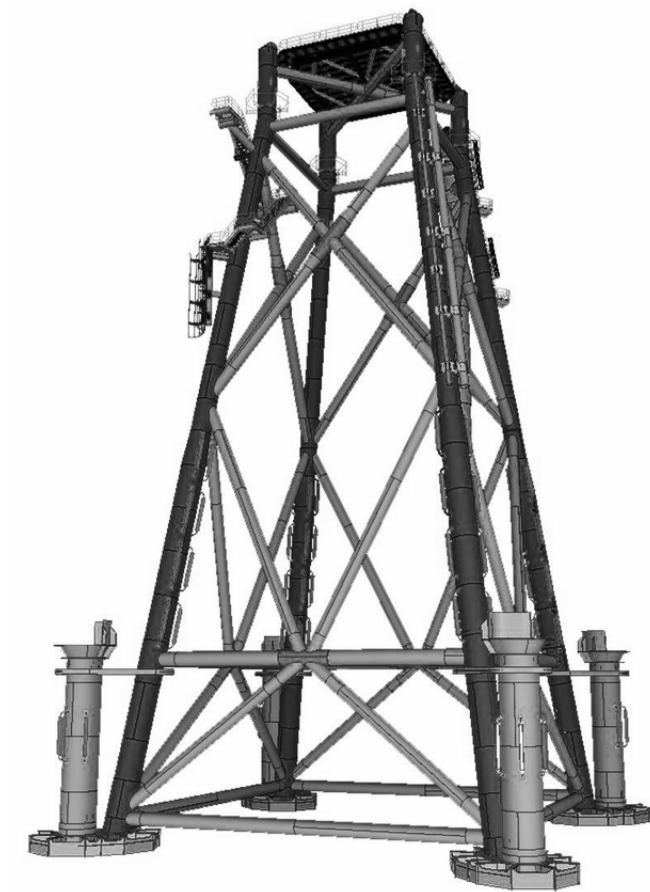


Rammeanalyse med matrisemetoden

10092, 10100 og 10006

6. november 2016



Forord

I faget TMR4167: Marin Teknikk 2 er vi blitt gitt i oppgave å lage et MATLAB-program som skal kunne beregne og analysere en konstruksjon ved hjelp av matrisemetoden. MATLAB er et program for høy-nivå programmering som vi ble introduserte oss i første semester gjennom faget TDT4105 Informasjonsteknologi Grunnkurs. Vi har fått en innføring i matrisemetoden gjennom forelesning ved Professor Jørgen Amdahl, samt kompendiet og øvingsopplegget.

Hensikten med prosjektet er å vise hvordan et dataprogram kan effektivisere en regneprosess når det blir flere variable enn ved enkle problemer, samt hvordan man skal gå frem for å designe et slikt program. Dette er et noe forenklet program, men det vil likevel være en god erfaring å ta med seg videre i studieløpet og arbeidslivet. Prosjektet har vært tidkrevende så vel som faglig krevende, noe som gjorde at det var utfordrende til å begynne med. Med god arbeidsmoral og en bratt læringskurve ble det likevel moro å jobbe med, og vi kom i mål med et resultat vi ble fornøyd med.

10092

Dato

10100

Dato

10006

Dato

Sammendrag

Prosjektets oppgave er å lage et MATLAB-program som analyserer en rammekonstruksjon bestående av rør- og I-profiler ved hjelp av matrisemetoden. Systemet er todimensjonalt og skal kunne ta ytre krefter gitt som punkt- og fordelte laster. Matrisemetoden tar utgangspunkt i en diskretisering av konstruksjonen. Rammekonstruksjonen vi skal analysere er gitt, men programmet skal være generelt slik at det kan fungere på en vilkårlige konstruksjon. Målet med oppgaven er å velge tverrsnittsdimensjoner på rør- og I-profilene. Dimensjoneringen gjøres på bakgrunn av en elementvis beregning av endemoment, midtmoment, skjærkraft og bøyespennning som beregnes av MATLAB-programmet. Rammebetingelsene tillater en maksimal bøyespennning å ligge i intervallet 30-70% av flytspenningen, som er gitt som 320 MPa.

Hovedfunksjonen rammeanalyse.m kjører samtlige underfunksjoner og lagrer resultatet i filen resultat.txt. Denne filen gir en elementvis oversikt over de nevnte belastningene. Største verdi og posisjon av endemoment, bøyespennning og skjærkraft er også presentert i resultatfilen. Tverrsnittsdimensjonene har blitt valgt gjennom en itereringsprosess for å gi en maksimal bøyespennning i henhold til oppgaveteksten.

Innholdsfortegnelse

1 Innledning	1
1.1 Konstruksjonsdata	1
1.2 Oppgaven	2
1.3 Rapporten	2
2 Matrisemetoden	3
2.1 Systemrelasjonen	3
2.2 Diskretisering	3
2.3 Lokal stivhetsmatrise	4
2.4 Global stivhetsmatrise	4
2.5 Lastvektor	5
2.6 Løsning av systemrelasjonen	5
3 MATLAB-funksjoner	6
3.1 Liste over MATLAB-funksjoner	6
3.1.1 rammeanalyse.m	6
3.1.2 lesInput.m	6
3.1.3 lengder.m	6
3.1.4 moment.m	6
3.1.5 lastvektor.m	6
3.1.6 I.m	7
3.1.7 stivhet.m	7
3.1.8 randbetingelser.m	7
3.1.9 endeM.m	7
3.1.10 midtM.m	7
3.1.11 boyespenning.m	7
3.1.12 shear.m	7
3.2 Tolkning av input-fil	8
3.2.1 Knutepunkter	8
3.2.2 Elementer	8
3.2.3 Laster	9
3.2.4 Profiler	9
4 Analyse av resultater	10
4.1 Momenter	10
4.1.1 Beregning av momenter	10
4.1.2 Beregning av midtmomenter	10
4.1.3 Moment på rammen	11
4.2 Bøyespenning	12
4.3 Skjærkraft	13
4.4 Aksialkraft	16
4.5 Iterering av tverrsnittsdata	17
4.6 Flytespenningsanalyse	18
4.7 Test på ny skipsramme - Øving 6	19

4.8 Sammenligning i Nauticus	20
5 Konklusjon	21
6 Gruppeevaluering	22
A MATLAB-filer	1
A.1 rammeanalyse.m	1
A.2 lesinput.m	4
A.3 lengder.m	6
A.4 moment.m	7
A.5 lastvektor.m	8
A.6 I.m	9
A.7 stivhet.m	11
A.8 randbetingelse.m	12
A.9 endeM.m	13
A.10 midtM.m	14
A.11 boyespenning.m	15
A.12 shear.m	16
B Inputfiler for MATLAB	17
B.1 input.txt	17
B.2 result.txt	18
B.3 inputoving6.txt	20
C Diagrammer	21
C.1 Momentdiagram	21
C.2 Skjærkraftdiagram	22
C.3 Aksialkraftdiagram	23
C.4 Spenningsfordeling ved originale tverrsnitt	24
C.5 Spenningsfordeling ved itererte tverrsnitt	25
D Resultater	26
D.1 Resultater fra Nauticus: Moment og skjærkrefter	26
D.2 Resultater fra Nauticus: Rotasjoner	27

Tabelliste

1	Konstruksjonsdata for rammekonstruksjonen	1
2	Sammenhengen mellom lokale og globale frihetsgrader, MNPC	4
3	Beskrivelse av knutepunkter	8
4	Beskrivelse av elementer	8
5	Beskrivelse av påsatte laster	9
6	Beskrivelse av profiler	9
7	Momentfordeling i rammekonstruksjonen	11
8	Spenningsfordeling i rammekonstruksjonen	13
9	Forskjeller skjærkrefter mellom MATLAB og Nauticus [®] 3D Beam	15
10	Tverrsnittsdata for rør- og I-profil	17
11	Andel av flytespenning for hvert element	18
12	Elementvis momentfordeling fra Øving 6	19

Figurliste

1	Rammekonstruksjonen som oppgitt i oppgaven	2
2	Diskretisering av rammekonstruksjonen	3
3	Eksempel på rammekonstruksjon	5
4	Moment fra lineært fordelt last.	11
5	Momentdiagram av rammekonstruksjonen	12
6	Skjærkraftdiagram for rammekonstruksjonen	14
7	Aksialkraftdiagram for rammekonstruksjonen	16
8	Profilene brukt for prosjektet	17
9	Bøyespenningsfordeling i rammekonstruksjonen	18
10	Skispramme fra Øving 6	19

1 Innledning

1.1 Konstruksjonsdata

Konstruksjonen har gitte ytre krefter, med unntak av P_1 , P_3 og q_{topp} . I tillegg er α_1 avhengig av studentnummer. For at ikke alle prosjektoppgavene skulle være like, får vi disse verdiene ut ifra studentnummeret til ett medlem av gruppen. Vi benyttet oss av XXX172. Dette gir oss følgende verdier:

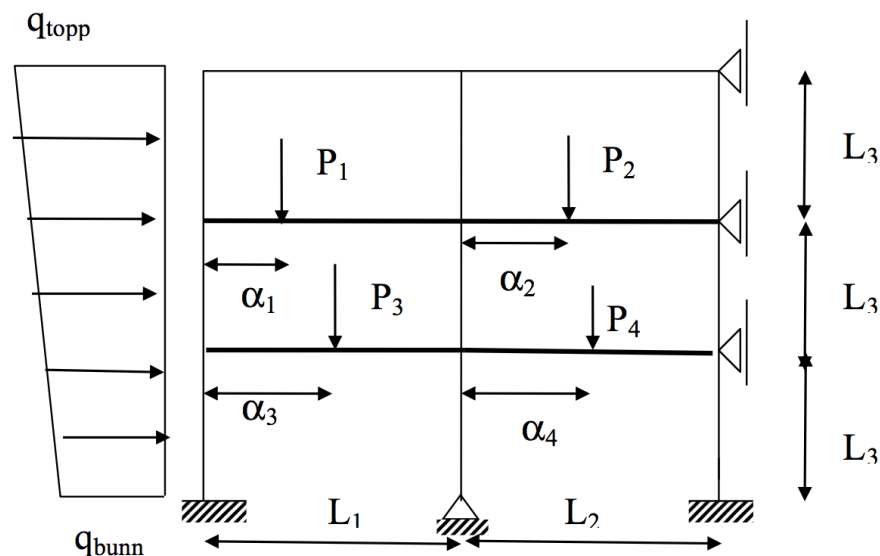
Punktlaster	
P_1	200 kN
P_2	300 kN
P_3	500 kN
P_4	400 kN
Jevnt fordelte laster	
Q_{topp}	160 kNm^{-1}
Q_{bunn}	80 kNm^{-1}
Lengder	
L_1	18 m
L_2	20 m
L_3	12 m
Forholdstall	
α_1	0,4
α_2	0,4
α_3	0,6
α_4	0,5
Spenningselementer	
Flytespenning, f_y	320 MPa
Elastisitetsmodul	210 GPa

Tabell 1: Konstruksjonsdata for rammekonstruksjonen

Dimensjoneringen av bjelke- og rørtverrsnittet blir en iterasjonsprosess hvor vi prøver oss fram med ulike tverrsnittsdata. Iterasjonsprosessen tar i utgangspunkt i at bøyespenningsnivået skal være i størrelsesorden 30-70% av flytespenningen ved de mest belastede elementene.

1.2 Oppgaven

Ved bruk av programmeringsverktøyet MATLAB, går oppgaven ut på å lage et enkelt program som gjør at vi kan analysere konstruksjonen med matrisemetoden. Programmet skal kunne brukes på en vilkårlig rammekonstruksjon. Vi tester programmet som lages ved å beregne på krefter til rammekonstruksjonen som vist i Figur 1.



Figur 1: Rammekonstruksjonen som oppgitt i oppgaven

Figur 1 representerer en idealisert modell av en stålramme i en modul på dekket til en offshoreplattform. Rammen antas å være uforskyvelig slik at deformasjon er fullstendig beskrevet av knutepunktsdeformasjonene. Flystespenningen er satt lik 320 MPa i alle elementene.

Rammekonstruksjonen er utsatt for sterke vindkrefter som antas å være lineært fordelt over høyden til rammekonstruksjonen. De har en henholdsvis intensitet q_{bunn} og q_{topp} ved topp og toppen av rammen. De vertikale søylene består av sirkulære rørtverrsnitt, mens de horisontale bjelkene består av I-profil. Knutepunktene kan antas å være stive.

1.3 Rapporten

Rapporten forklarer først matrisemetoden som skal brukes i MATLAB-funksjonene og hvordan vi ønsker å bruke dette. Vedlagt er alle funksjonene som brukes i beregningene av konstruksjonsrammen.

2 Matrisemetoden

Matrisemetoden er basert på deformasjonsmetoden, med noen få forskjeller. Den viktigste forskjellen er at den ser på hele systemet som en helhet, i stedet for ett og ett element [1]. Det er kun knutepunksdeformasjoner som vil være ukjente i oppgaven.

2.1 Systemrelasjonen

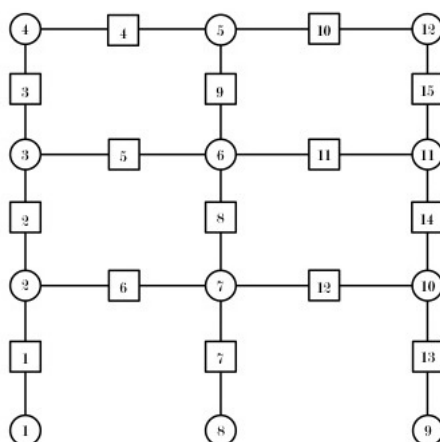
Matrisemetoden bruker en systemrelasjon som vi skal finne. Denne består av tre komponenter; stivhetsmatrisen, lastvektoren og rotasjonsvektoren. Disse er begge komponenter i Ligning (2.1.1).

$$\text{Systemrelasjonen: } \mathbb{K}r = \mathbb{R}. \quad (2.1.1)$$

Der \mathbb{K} er stivhetsmatrisen, r er rotasjonsvektoren og \mathbb{R} er lastvektoren. Ved bruk av funksjonene som lages i MATLAB ønsker vi å analysere informasjonen i stivhetsmatrisen og bruke dette til å løse systemrelasjonen som vist i Ligning (2.6.1). Denne vil også bli brukt til beregning av endemomenter, midtmomenter, bøyespenninger og skjærkrefter til bjelkene.

2.2 Diskretisering

Det første steget i matrisemetoden er å diskretisere rammekonstruksjonen i et gitt antall elementer. Forbindelsen mellom hvert element kalles knutepunkter. I Figur 2 er elementenumre markert i bokser, mens knutepunkter er omringet av sirkler. Knutepunktene forskyvninger og rotasjoner definerer vi gjennom systemets frihetsgrader. Vi har prøvd å dele opp elementer og knutepunkter ved at vi begynner på venstre side av konstruksjonen og jobber oss oppover, før tellingen går mot høyre.



Figur 2: Diskretisering av rammekonstruksjonen

Ved bruk av matrisemetoden ser vi bort i fra tverr- og aksialforskyvninger av rammen, slik at hovedfokuset er på bøyedeformasjoner. Etter diskretiseringen setter vi opp en sammenheng mellom de lokale og globale frihetsgradene som vist i Tabell 2. Et slikt matriseoppsett kalles for *Matrix of Nodal Point Correspondence* - *MNPC*, også kalt konntekvititetsmatrisen. DoF står for *Degree of Freedom*.

Element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DoF ₁	1	2	3	4	3	2	8	7	6	5	6	7	9	10	11
DoF ₂	2	3	4	5	6	7	7	6	5	12	11	10	10	11	12
Lengde	12	12	12	18	18	18	12	12	12	20	20	20	12	12	12

Tabell 2: Sammenhengen mellom lokale og globale frihetsgrader, MNPC

2.3 Lokal stivhetsmatrise

Etter at diskretiseringen av rammekonstruksjonen er utført, settes det opp en lokal stivhetsmatrise. Den lokale stivhetsmatrisen er en analyse av hvert element. Stivhetsmatrisen for element i kan generelt skrives som:

$$\mathbf{k}_i = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix} \frac{EI_i}{l_i}, \quad (2.3.1)$$

der k_{11} og k_{22} er henholdsvis stivhetsleddene for ende 1 og 2. k_{12} og k_{21} er koblingsledd mellom moment i den ene enden og rotasjon i den andre. I er annet arealmoment for det aktuelle elementets tverrsnitt. Den lokale stivhetsmatrisen endres bare på grunnlag av verdiene til I , E og L .

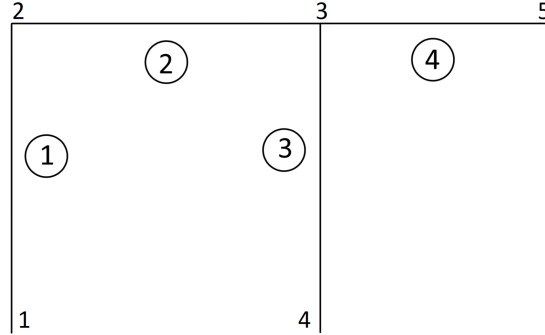
2.4 Global stivhetsmatrise

Den lokale stivhetsmatrisen \mathbf{k}_i skal settes sammen til et globalt system og det settes opp en global stivhetsmatrise, også kalt systemstivhetsmatrisen. Dette gjøres ved hjelp av MNPC-matrisen som vist i Tabell 2. Størrelsen i matrisen avgjøres avhengig av antall knutepunkter og frihetsgrader i rammen. For prosjektets rammekonstruksjon blir dette en matrise på 12 rader og 12 kolonner.

For å plassere komponentene fra den lokale stivhetsmatrisen gjør vi om elementstivhetsmatrisen om til en del av den globale stivhetsmatrisen. Ved å endre indeksen til k i matrisen i (2.3.1), endres den fra lokal- til global stivhet.

Eksempelvis har vi en rammekonstruksjon med 4 elementer og 5 knutepunkt som vist i Figur 3. La element 3 i konstruksjonen ha knutepunkt 3 og 4 i henholdsvis lokalt knutepunkt 1 og 2. For å gjøre denne en del av den globale stivhetsmatrisen gjøres følgende:

$$k_3 = \frac{E_3 I_3}{l_3} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} k_{33} & k_{34} \\ k_{43} & k_{44} \end{bmatrix} \Rightarrow \begin{bmatrix} (k_{11})_3 & (k_{12})_3 \\ (k_{21})_3 & (k_{22})_3 \end{bmatrix}. \quad (2.4.1)$$

**Figur 3:** Eksempel på rammekonstruksjon

$$K = \begin{bmatrix} (k_{11})_1 & (k_{12})_1 & 0 & 0 & 0 \\ (k_{21})_1 & (k_{22})_1 + (k_{11})_2 & (k_{12})_2 & 0 & 0 \\ 0 & (k_{21})_2 & (k_{22})_2 + (k_{11})_3 + (k_{11})_4 & (k_{12})_3 & (k_{12})_4 \\ 0 & 0 & (k_{21})_3 & (k_{22})_3 & 0 \\ 0 & 0 & (k_{21})_4 & 0 & (k_{22})_4 \end{bmatrix} \quad (2.4.2)$$

Matrisen som blir generert i Ligning (2.4.1) settes så inn i matrisen i (2.4.2) i punktene (3,3), (4,3), (3,4) og (4,4). Dette gjelder altså for en ramme med 5 knutepunkter og 2 frihetsgrader. Vi ser ut ifra denne globale matrisen at det lønner seg med minst mulig numerisk avstand mellom knutepunkt og bjelker. Dermed opprettholdes diagonaldominanten mest mulig.

2.5 Lastvektor

Lastvektoren \mathbb{R} er summen av påsatte moment og fastinnspenningsmoment, fortkoret til fim. Deformasjonsmetoden tar utgangspunkt i at indre moment i bjelken som følge av rotasjoner, $M_{\text{rotasjoner}}$ eller $M_{ij}(\theta_i, \theta_j)$, må utligne ytre momenter fra påsatte moment og ytre momenter fra laster, M_{ytre} .

$$M_{ij}(\theta_i, \theta_j) + M_{\text{ytre}} = 0. \quad (2.5.1)$$

2.6 Løsning av systemrelasjonen

Systemets stivhetsrelasjon er gitt ved Ligning (2.1.1). Ved å løse denne med hensyn på r får man den inverterte matrisen

$$r = \mathbb{K}^{-1}\mathbb{R}. \quad (2.6.1)$$

Dette ligingsystemet kan lett løses manuelt om systemene ikke blir for store. For større konstruksjoner med mange ukjente vil det være fordelaktig å bruke et

dataprogram som for eksempel MATLAB. Oppgaveteksten gir oss to muligheter til å løse dette ligningsystemet. En metode inkluderer en rotasjonsfjær som settes inn ved hvert knutepunkt. Stivhet velges ut ifra opplageret til de ulike knutepunktene. Vi har valgt den andre metoden, som tar utgangspunkt i at man nuller ut rad, kolonne og tilsvarende element i lastvektor dersom et knutepunkt er fast innspent. Diagonalene må også settes lik et vilkårlig tall. I funksjonen har vi satt diagonalene lik 1. Den siste metoden ble valgt da det virket lettere å implementere denne i MATLAB.

3 MATLAB-funksjoner

3.1 Liste over MATLAB-funksjoner

Dette er en liste over alle MATLAB-funksjonene som ble laget for dette prosjektet med hver sin beskrivelse av dens hensikt.

3.1.1 rammeanalyse.m

Denne funksjonen setter alle funksjonene sammen og kjører de, før resultatet blir skrevet ut i resultat.txt-filen i MATLAB, se vedlegg B.2. Denne definerer også rekkefølgen som den henter input fra og lagrer alle disse variablene i *workspace*. rammeanalyse.m kan analysere kreftene for en vilkårlig ramme, gitt at den følger oppsettet i input.txt-filen på samme måte som vi har definert det i lesinput.m-funksjonen. Se vedlegg A.1.

3.1.2 lesInput.m

Dette er filen som danner grunnlaget for de resterende funksjonene. Den leser input.txt-filen og legger input-data i forskjellige matriser avhengige av elementer, knutepunkter, laster og profiltyper. Se vedlegg A.2.

3.1.3 lengder.m

Tar inn knutepunktskoordinatene fra matrisen laget i lesInput.m og gir ut en ny $n \times 1$ -matrise med lengdene til elementene. Den tar også høyde for bjelker som ligger på skrå ved å bruke Pytagoras ligning. Se vedlegg A.3.

3.1.4 moment.m

Regner ut momentet på elementene avhengig av punktlast, jevnt fordelt last eller lineært fordelt last. Gir ut enmatrise med moment i lokalt knutepunkt 1 og 2 for hvert element. Se vedlegg A.4

3.1.5 lastvektor.m

Lager lastvektoren som danner grunnlaget for systemrelasjonen som beskrevet i (2.1.1). Denne blir også påvirket av randbetingelsene. Legger til summen av

innspenningsmomentene i lastvektoren i henhold til Ligning (2.5.1). Se vedlegg A.5.

3.1.6 I.m

Regner ut annet arealmoment for profilene brukt i rammekonstruksjonen. Hvert I-profil tar høyde for forskjeller på tykkelse av flens og steg. Se vedlegg A.6.

3.1.7 stivhet.m

Regner ut den globale stivhetsmatrisen og gir ut en $n \times n$ -matrise der $\frac{EI}{l}$ er multiplisert inn i matrisen. Det er denne som tar inn den lokale stivhetsmatrisen som beskrevet i Seksjon 2.3. Se vedlegg A.7.

3.1.8 randbetingelser.m

Sjekker om knutepunktet er fast innspent eller fritt opplagt. Ut i fra denne informasjonen, korrigerer den radene og kolonnene i den globale stivhetsmatrisen for de aktuelle knutepunktene. Se vedlegg A.8.

3.1.9 endeM.m

Avhengig av rotasjonen, stivhetsmatrisen og eksisterende moment, regner den ut endemomentet på hvert element. Vil også regne endemoment på elementer som ikke har påsatt last grunnnet momenetlikevekt i enkelte hjørner. Se vedlegg A.9.

3.1.10 midtM.m

Ved punktlast, regner den ut moment rett under den aktuelle resultanten. Ved jevnt fordelt last eller lineært fordelt last, regnes momentet ut på midten av elementet. Se vedlegg A.10.

3.1.11 boyespenning.m

Regner ut bøyespenningen på lokalt knutepunkt 1, 2 og i midten av hvert element. Tar inn lengden til arealsenteret og annet arealmoment, samt fastinnspenningsmomenter. Se vedlegg A.11.

3.1.12 shear.m

Regner ut skjærkraften i hvert element ved lokalt knutepunkt 1 og 2. Regner ikke ut skjærkraft på midten av elementer da dette ikke er nødvendig for analysen. Se vedlegg A.12.

3.2 Tolkning av input-fil

3.2.1 Knutepunkter

x-koordinat	y-koordinat	Opplager
0	0	1
0	12	0
0	24	0
0	36	0
18	36	0
18	24	0
16	12	0
18	0	0
38	0	1
38	12	0
38	24	0
38	36	0

Tabell 3: Beskrivelse av knutepunkter

Den første delen av input.txt-filen beskriver antall knutepunkter og lokaliseringen til knutepunktene. Positiv retning for koordinater av knutepunkt er høyre og oppover. Tabell 3 viser hvilken informasjon kolonnene gir. 3. kolonne tilsier om elementet er fritt opplagt eller fast innspent, henholdsvis 0 og 1.

3.2.2 Elementer

DoF ₁	DoF ₂	E-modul	Orientering
1	2	210e9	1
2	3	210e9	1
3	4	210e9	1
4	5	210e9	0
3	6	210e9	0
2	7	210e9	0
8	7	210e9	1
7	6	210e9	1
6	5	210e9	1
5	12	210e9	0
6	11	210e9	0
7	10	210e9	0
9	10	210e9	1
10	11	210e9	1
11	12	210e9	1

Tabell 4: Beskrivelse av elementer

Rammekonstruksjonen består av n elementer med lokalt knutepunkt 1 og 2 beskrevet i 1. og 2. kolonnei Tabell 4. Dette følger diskretiseringen som vist i Figur 2. I denne delen inkluderes også E-modulen til elementene. Orientering forteller om bjelken er horisontal eller vertikal med henholdsvis 0 og 1.

3.2.3 Laster

Type	Verdi	Element	α /lokalt maksimum
1	200 000	5	0,4
1	300 000	11	0,4
1	500 000	6	0,6
1	400 000	12	0,5
2	80 000	1	1
3	26 666	1	2
2	106 666	2	1
3	26 666	2	2
2	133 333	3	1
3	26 666	3	2

Tabell 5: Beskrivelse av påsatte laster

Lastmatrisen består av n rader og tar inn punkt-, jevnt fordelte- og linært fordelte laster. I 2. kolonne beskrives lastintensiteten. Det blir også beskrevet hvilket element lasten virker på, samt α -verdien for punktlastene. 4. kolonne tilsier hvilket lokale knutepunkt som er maksimum for fordelte laster. Positiv retning er her satt ned og til høyre.

3.2.4 Profiler

Type	h_{steg} Diameter	t_{steg} Rørtykkelse	t_{topp}	b_{topp}	t_{bunn}	b_{bunn}
0	400	8,6	13,5	180	13,5	180
1	500	10	-	-	-	-

Tabell 6: Beskrivelse av profiler

Siste delen av input-filen beskriver tverrsnitsdataen brukt på rør- og I-profilet. 1. kolonne tilsier om det er rør eller I-profil, med henholdsvis verdiene 1 og 0. Kolonne 4 og 5 beskriver toppflensen, og kolonne 5 og 6 beskriver bunnflensen.

4 Analyse av resultater

4.1 Momenter

4.1.1 Beregning av momenter

Alle elementene er behandlet som fritt opplagt bjelker med tanke på rotasjon, men som fast innspenne bjelker når vi ser på momentet i endene. Ved punktlast på et vilkårlig sted på et element gjelder følgende moment:

$$m_{ab} = \frac{-P \cdot ab^2}{l^2}, \quad m_{ba} = \frac{P \cdot a^2b}{l^2}, \quad (4.1.1)$$

Momentene fra 4.1.1 er grunnlaget for fastinnspenningsmomentene for punktlast, mens følgende gjelder for jevnt fordelt last og lineært fordelt last [1]:

$$\text{Jevnt fordelt last: } m_{ab} = -\frac{1}{12}ql^2, \quad m_{ba} = \frac{1}{12}ql^2, \quad (4.1.2)$$

$$\text{Lineært fordelt last: } m_{ab} = -\frac{1}{30}ql^2, \quad m_{ba} = \frac{1}{20}ql^2. \quad (4.1.3)$$

4.1.2 Beregning av midtmomenter

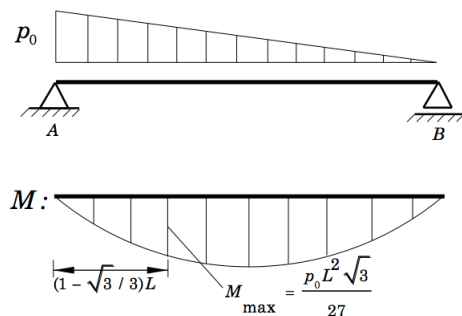
For bjelker med fordelte laster ønsker vi å finne momentet på midten, selv om maksimalmomentet egentlig vil ligge under resultanten. Momentet er skapt av endemomentene og kan regnes ut på følgende vis:

$$\text{Midtmoment fra endemoment: } = \frac{-M_1 + M_2}{L}, \quad (4.1.4)$$

der M_1 er endemomentet i lokal ende 1, M_2 er endemomentet i lokal ende 2. For bjelker med punktlaster, skal momentet regnes ut for plasseringen av punktlasten og ikke midten:

$$\text{Midtmoment: } M_{\text{midt}} = -M_1(1 - \alpha) + M_2\alpha. \quad (4.1.5)$$

Midtmomentet for lineært fordelte laster, dvs. trekanten, vil ha et midtmoment som vist i Figur 4 og beskrevet i Ligning (4.1.6).



Figur 4: Moment fra lineært fordelt last.

$$M(x) = \frac{p_0 x}{6L}(2L^2 - 3Lx + x^2) \Rightarrow M\left(x = \frac{L}{2}\right) = \frac{p_0 L^2}{16}. \quad (4.1.6)$$

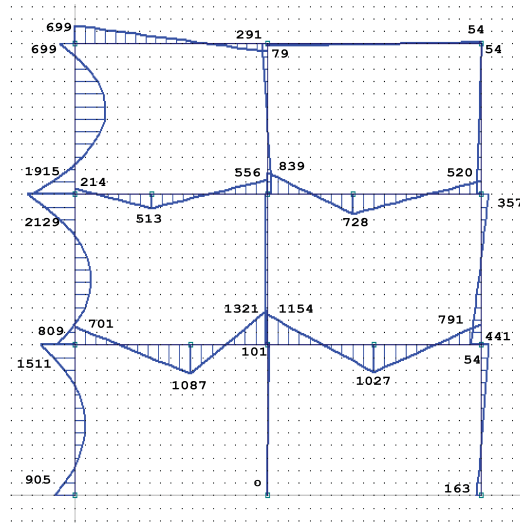
Ettersom vi har jevnt fordelt last på samme område som vi har lineært fordelt last, vil fastinnspegningsmomeneten fra 4.1.2 og 4.1.3 legges til dette.

4.1.3 Moment på rammen

Element	Lokal ende 1 [kNm]	Midten [kNm]	Lokal ende 2 [kNm]
1	908,28	-470,14	-1 511,45
2	809,54	-470,14	-2 129,75
3	1 915,59	-1 332,32	-699,78
4	699,78	0,00	291,09
5	214,16	-513,08	-556,05
6	701,91	-1 086,70	-1 320,89
7	-0,00	0,00	101,17
8	64,84	0,00	-72,66
9	-210,71	0,00	-211,27
10	-79,82	0,00	-53,81
11	839,42	-728,28	-520,17
12	1 154,87	-1 026,89	-791,34
13	1 75,11	0,00	350,22
14	441,12	0,00	356,91
15	163,26	0,00	53,81

Tabell 7: Momentfordeling i rammekonstruksjonen

Ut ifra Ligning (4.1.6) beregnes midtmomentet til elementer som har fordelte laster, på element 1, 2 og 3. Ved midtmomentene i element 5, 6, 11 og 12, beskrevet i Tabell 7, beskriver disse da momentet til punktlasterens resultant. Det virker logisk at maksimal moment utøves på element 2, da det er her det er mest jevnt fordelt last på dette elementet.



Figur 5: Momentdiagram av rammekonstruksjonen

Momentdiagrammet for rammen er tatt ut fra Nauticus[®] 3D Beam. Dette stemmer godt overens ut ifra type last som rammekonstruksjonen er utsatt for. Fast innsente knutpunkter (1 og 9) har et tydelig endemoment, mens det ikke er noe endemoment i knutepunkt 7.

4.2 Bøyespenning

Bøyespenningen med et gitt moment, M og arealmoment, I , regnes ut med følgende formel, avhengig av hvilken ende vi ser på:

$$\text{På ender: } \sigma_{ende} = \frac{M_{ende_i}}{I} y_c, \quad \text{på midten: } \sigma_{midt} = \frac{M_{midt}}{I} y_c. \quad (4.2.1)$$

Element	Lokal ende 1 [MPa]	Midten [MPa]	Lokal ende 2 [MPa]
1	89,539	-46,347	-149,000
2	79,805	-68,056	-209,953
3	188,841	-131,341	-68,985
4	94,685	0,000	39,387
5	28,977	-69,424	-75,237
6	94,973	-147,038	-178,725
7	-0,000	0,000	9,973
8	6,392	0,000	-7,163
9	-20,772	0,000	-20,828
10	-10,800	0,000	-7,281
11	113,579	-98,541	-70,382
12	156,262	-138,945	-107,073
13	17,262	0,000	34,525
14	43,486	0,000	35,185
15	16,094	0,000	5,304

Tabell 8: Spenningsfordeling i rammekonstruksjonen

Vi får dermed følgende resultater som vist i Tabell 8. Det er bare bergenet midtmoment på elementer som har ytre laster, følgelig er det ikke beregnet bøyesspenning på elementer uten last.

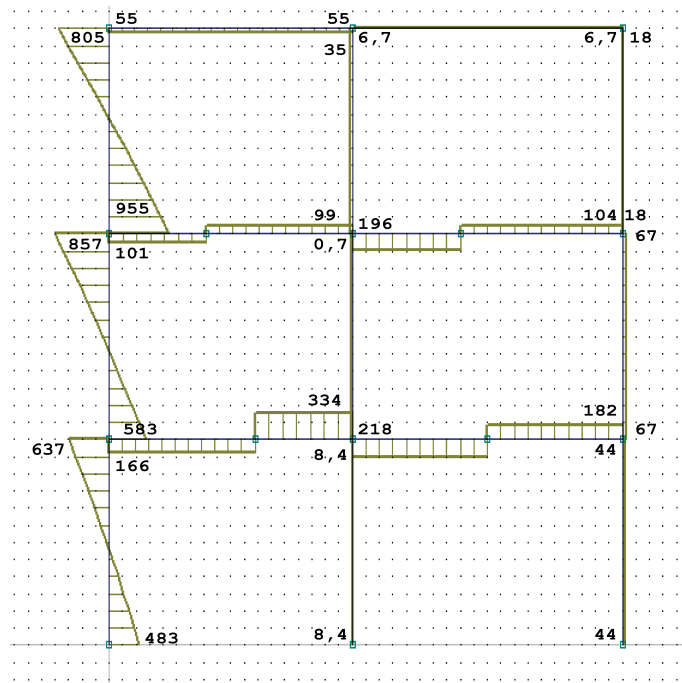
4.3 Skjærkraft

For å tegne skjærkraftdiagram tar man utgangspunkt i skjærkraften i hver ende av hvert element. Hvordan diagrammet vil se ut avhenger av type kraft, endemomenter og fordelte laster. Der hvor det opptrer mer enn én kraft, brukes gjerne superposisjonsprinsippet. Skjærkraften er gitt ved:

$$Q = Q_0 - \frac{M_{ij} + M_{ji}}{l_{ij}}. \quad (4.3.1)$$

Q_0 er skjærkraften i hver bjelkeende som følge av last. Skjærkreftene til elementer er uavhengig av opplager. Denne verdien kan beregnes ved kraft- og momentlikevekt eller eventuelt hentes ut fra tabeller. Det siste leddet i Ligning (4.3.1) er skjærkraft som følge av endemomenter. Verdien her vil være den samme som stigningen mellom bjelkeendemomentene. Dette er fordi skjærkraft er den deriverte av momentet, gitt ved Ligning (4.3.2):

$$\frac{dM}{dx} = V. \quad (4.3.2)$$



Figur 6: Skjærkraftdiagram for rammekonstruksjonen

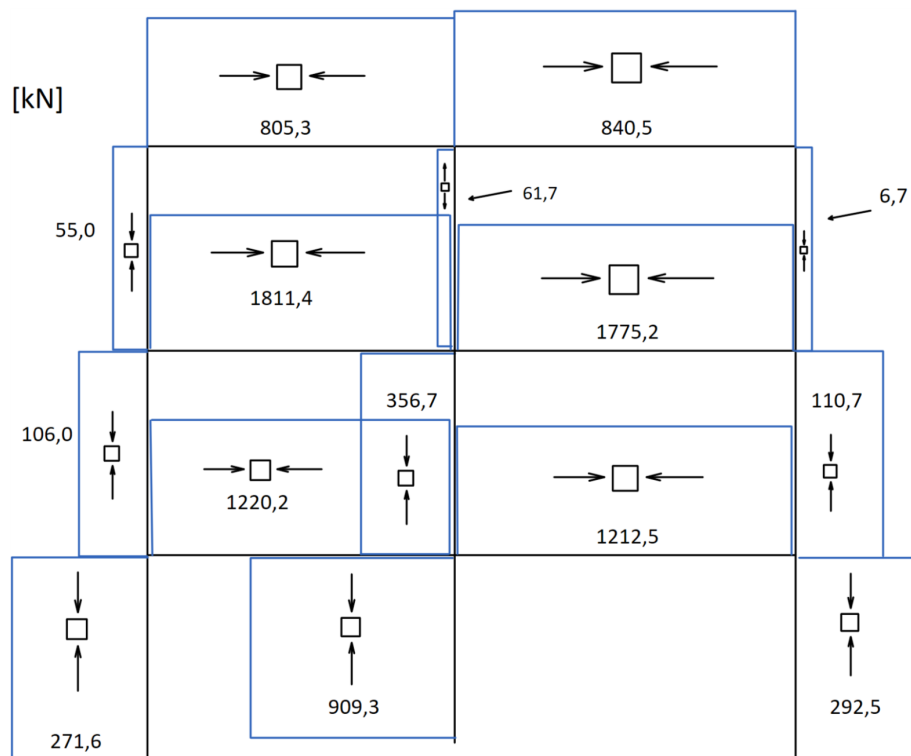
Skjærkraftdiagrammet er hentet fra Nauticus[®] 3D Beam. Som man kan se finner man den største verdien ved lokal ende 1 på bjelke 3, med verdien 955 kN. Med oppgaven gitte laster er dette å forvente.

Element	MATLAB Ende 1 [kN]	Nauticus Ende 1 [kN]	MATLAB Ende 2 [kN]	Nauticus Ende 2 [kN]
1	483,1	483,1	-636,9	-636,9
2	583,3	583,3	-856,7	-855,6
3	954,7	953,9	-805,3	-804,8
4	55,0	55,0	55,0	55,0
5	101,0	101,0	-99,0	-99,0
6	165,6	165,5	-334,4	-334,3
7	8,4	8,4	8,4	8,4
8	-0,7	-0,9	-0,7	-0,9
9	35,2	34,5	35,2	34,5
10	-6,7	-6,6	-6,7	-6,6
11	196,0	196,0	-104,0	-104,0
12	218,2	218,2	-181,8	-181,8
13	43,8	43,8	43,8	43,8
14	66,5	66,3	66,5	66,3
15	18,1	18,2	18,1	18,2

Tabell 9: Forskjeller skjærkrefter mellom MATLAB og Nauticus[®] 3D Beam

Etter å ha sammenlignet resultatene fra vårt eget MATLAB-program og Nauticus[®] 3D Beam ser vi at det er svært små avvik. Det største avviket finner man i bjelke nummer 3. Her får vi en verdi på 954,7 kN i MATLAB mens Nauticus[®] 3D Beam gir oss 953,9 kN. Dette er et avvik på 0,083%, som er tilnærmet like. I element nummer 8 har vi målt en verdi på -0,7 kN og en på -0,9 kN i henholdsvis MATLAB og Nauticus[®] 3D Beam. Med en forskjell på 28,6% er dette en betydelig større prosentvis forskjell, men dersom man ser på disse verdiene sammen med de andre, vil ikke krefter på under 1 kN ha noen reell innvirkning. Vi vil derfor si at resultatene i realiteten er identiske.

4.4 Aksialkraft



Figur 7: Aksialkraftdiagram for rammekonstruksjonen

For å lage aksialdiagrammet brukes skjærkraftdiagrammet. Om en last går parallelt med en bjelke vil den gi aksialkrefter i systemet. Vår oppgave har ikke noen oppgitte ytre punktlaster som virker i et knutepunkt, så dette er ikke tatt hensyn til. I knutepunkter vil skjærkrefter fordele seg som aksialkrefter og skjærkrefter ut ifra orienteringen til bjelkene som er tilknyttet knutepunktet. Skjærkreftene fra et element brer seg over til aksialkrefter til andre elementer som står vinkelrett på dette.

Aksialkrefter kan bestemmes fra skjærkrefter om det ikke virker flere ukjente krefter på knutepunktet enn vi har likevektslikninger. Ved å starte i et punkt der kun aksialkraften er ukjent kan en bestemme aksialkraften fra skjærkraften. Videre kan dette gjøres for hele systemet. I vår oppgave var det kun aksialkreftene som var ukjente i element 4, så det falt seg naturlig å starte beregningen derfra. Aksialkraftdiagrammet er gitt i Figur 7.

4.5 Iterering av tverrsnittsdata

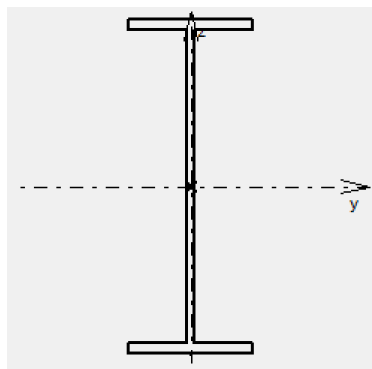
Ved oppgavestart får vi ikke oppgitt tverrsnittsdata, og har derfor gjennom programmeringsprosessen brukt en vilkårlig rør og I-profil. Vi ønsker å bruke dimensjonere tverrsnittene slik at bøyespenningsnivået er i størrelsesorden 30-70% av flytespenningen ved de mest belastede bjelkene. Dette gjelder da spesielt ved bjelkeender, ved punktlaster og midt på bjelker med fordelt last. Ved vertikale bjelker skal det bli brukt rørtverrsnitt, mens horisontale bjelker benytter I-profil.

Flytespenningen, f_y er gitt til å være 320 MPa. Maksimal bøyespenning i henhold til den ønskede størrelsesorden er derfor 220 MPa, mens minimal er 96 MPa. Vi ønsker dermed å iterere verdiene på tverrsnittene slik at vi ligger i underkant av 70% av bøyespenningen for å bruke minimalt med materialer.

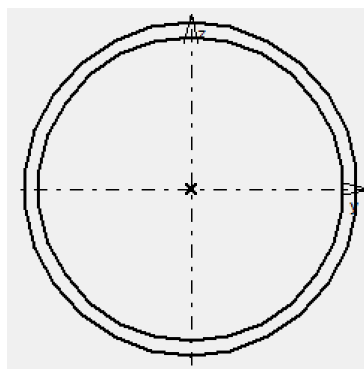
Dimensjoner	Originale [mm]	Itererte [mm]
Rørdiameter	500,0	700,0
Rørtykkelse	10,0	30,0
h_{steg}	400,0	750,0
t_{steg}	8,6	20,0
t_{topp}	13,5	25,0
b_{topp}	180,0	300,0
t_{bunn}	13,5	25,0
b_{bunn}	180,0	300,0

Tabell 10: Tverrsnittsdata for rør- og I-profil

Ved beregning av bøyespenning med de originale tverrsnittsprofilene, fikk vi en maksimalt bøyespenning på 1 081,01 MPa, altså langt over flytespenningen. Fra Nauticus® 3D Beam ser vi at dette gjelder spesielt ved knutepunktene 3 for rør-profilen og 7 for I-profilen.



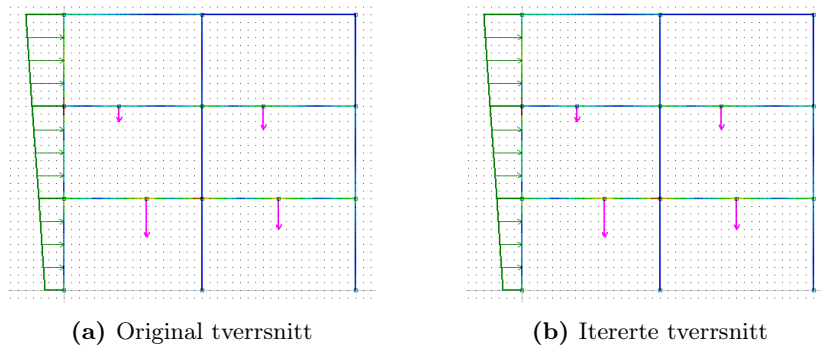
(a) I-profilen, for horisontale bjelker



(b) Rør-profilen, for vertikale bjelker

Figur 8: Profilene brukt for prosjektet

4.6 Flytespenningsanalyse



Figur 9: Bøyespenningsfordeling i rammekonstruksjonen

De røde områdene i Figur 9 viser hvor spenningen er størst. Begge tilfellene viser at spenningen er størst ved knutepunkt 3 og 7, henholdsvis på element 2 og 6. Med itererte tverrsnittsverdier ser vi fra Figur 9b at de røde område er redusert, riktignok rundt samme område. For forstørrede bilder, se vedlegg C.4 og C.5.

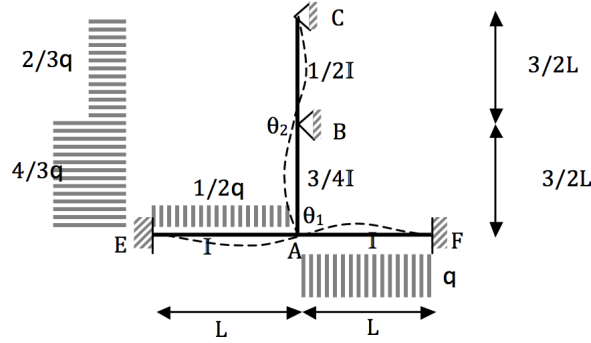
Element	Lokal ende 1 [%]	Midten [%]	Lokal ende 2 [%]
1	27,9	14,5	46,6
2	24,9	21,3	65,6
3	59,1	41,0	21,5
4	29,6	0,0	12,3
5	9,1	21,7	23,5
6	26,7	45,9	55,8
7	0,0	0,0	3,1
8	2,0	0,0	2,2
9	6,4	0,0	6,5
10	3,4	0,0	2,3
11	35,5	30,8	22,0
12	48,8	43,4	33,5
13	5,4	0,0	10,8
14	13,6	0,0	11,0
15	5,0	0,0	1,7

Tabell 11: Andel av flytespenning for hvert element

Vi ser fram Tabell 8 at høyeste bøyespenning er beregnet til 209,9 MPa som tilsvarer 65,5 % av flytespenningen, $f_y = 320$ MPa. Dette er innenfor kriteriene til flytespenningsområdet på 30-70%. Vi ønsker å ha så høy andel som mulig for å unngå overdimensjonering av profilene.

4.7 Test på ny skipsramme - Øving 6

I prosjektoppgaven ønsker vi å regne ut bøyemomentet ved alle bjelkeender for skipsrammen fra Øving 6, som vist i Figur 10. Vi utfører samme rammeanalyse som for systemet i MATLAB-programmet og sjekker at de får verdier som samsvarer med tidligere resultater.



Figur 10: Skipsramme fra Øving 6

Det blir riktignok ikke oppgitt profildata for bjelkeelementene i konstruksjonen, men heller forskjellige koeffisienter for hver I-profil. Det ble derfor satt en fast EI -verdi som et forholdstall mellom alle arealmomentene.

Likevekten i knutepunktene tilsier at $\Sigma M_A = \Sigma M_B$. Dette gir oss videre at $M_{AE} + M_{AF} + M_{AB} = 0$, mens $M_{BA} + M_{BC} = 0$. Dette gir oss tilsammen følgende likningssystem:

$$\frac{EI}{L} \begin{bmatrix} 10 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \frac{qL^2}{16} \begin{bmatrix} 2 \\ -1 \end{bmatrix}. \quad (4.7.1)$$

Løsningen til Ligningen (4.7.1) for skipsrammen kan dermed uttrykkes som:

$$\theta_1 = 0,18 \cdot \frac{qL^3}{12EI} \quad \text{og} \quad \theta_2 = -0,31 \frac{qL^3}{12EI}. \quad (4.7.2)$$

Med uttrykket for θ_1 og θ_2 fra Ligningene (4.7.2) kan det dermed bøyemomentene regnes ut ved enden av hvert element i konstruksjonen som vist i Tabell 12.

Element	Ende 1 [qL^2]	Midten [qL^2]	Ende 2 [qL^2]	Rotasjoner [rad]
EA	-0,0115	-0,0058	0,1020	0
AF	0,1436	0,0266	-0,0532	0,0151
AB	-0,2456	-0,1454	0,2134	- 0,0258
BC	-0,2134	-0,0809	0,00	-0,0809

Tabell 12: Elementvis momentfordeling fra Øving 6

I rammeanalysen av skipsrammen i Øving 6, er annet arealmoment, I , og elastisiteten, E satt lik 1. Dermed blir benevningen til momentverdiene uttrykt som qL^2 . Alle endemomenter, midtmomenter og rotasjoner stemmer med utregninger for hånd.

Det må likevel nevnes at skipsrammen ikke har noen punktlaster, og programmet fra MATLAB får dermed ikke testet om denne delen av programmet er riktig. Med midtmoment og rotasjoner riktig uttrykt i Tabell 12, vet vi at punktlaster ville vært riktig – også for skipsrammen.

4.8 Sammenligning i Nauticus

Med mange variabler i spill kan det være vanskelig å kontrollere MATLAB sine utregninger for hånd, og Nauticus[®] 3D Beam ble brukt hyppig til å kryss-sjekke resultatene. Ved sammenligning av håndregnede tall, MATLAB-resultater og Nauticus[®] 3D Beam ser vi at det er svært små avvik. På bøyepening og skjærkrefter er MATLAB-programmet mer nøyaktig ved å ikke runde opp desimaler. Ved bøyemomenter ser vi likevel at det er enkelte avvik på noen fåtall elementer i størrelsesorden ± 2 kNm. Midtmomentene er vanskelig å lese av i Nauticus[®] 3D Beam direkte. Det er mulig å få en viss antydning til svarets troverdighet ved å bruke plot-funksjonen i Nauticus[®] 3D Beam.

Ved lineært fordelte laster i kombinasjon med jevnt fordelte laster ser vi at momentavviket er noe redusert, i størrelsesorden $\pm 0,05$ kNm. Det må nevnes at vi ikke vet hvilken metode Nauticus[®] 3D Beam bruker for beregning av momentet, men ettersom det finnes enkelte avvik, kan vi anta at denne ikke er likt som oss.

5 Konklusjon

Gjennom prosjektets periode har vi laget et program i MATLAB som beregner endemomenter, midtmomenter, bøyespenning og skjærkrefter på bjelker i en vilkårlig rammekonstruksjon. Programmet er brukt til å analysere konstruksjonen og for å kunne komme frem til et sett med passende tverrsnittsdata. Maksimal bøyespenningen i et hvert element skal ligge rundt 30-70% av flytspenningen, f_y på 320 MPa. For vertikale bjelker ble det brukt rør-profil med ytre diameter på 500 mm og 10 mm tykkelse. Horisontale bjelker er I-profiler, med data tilnærmet IPE 800 med steghøyde på 750 mm. Kombinert ga dette en maksimal spenning på 209,9 MPa, 65,6% av flytespenningen. Dette ble utøvet på element 2 og 6. Dette stemmer i henhold til maksimal bøyespenning fra Nauticus® 3D Beam.

Etter flere runder med iterering av både rør- og I-profil, har vi fått tilstrekkelig små profiler som mulig innenfor prosjektets rammer. Underveis i prosessen så vi en overraskende stor uavhengighet mellom bøyespenning og profiltykkelse på de forskjellige profilene. Dette kan skyldes at maksimal bøyespenning var på et rør-profil, mens nest mest bøyespenning befant seg et annet sted, på et I-profil. Dimensjonene på enkelte av elementene kan sikkert bli skalert ned, men oppgaven tillot kun én rør-profil og én I-profil. Derfor fungerer MATLAB-funksjonen vår for bare en av hver av de nevnte profiltypene.

Vi har også brukt rammeanalysen (se vedlegg A.1) til å kontrollere resultatene på skipsrammen i Øving 6, beskrevet i Seksjon 4.7 for å se at programmet kunne brukes på en vilkårlig, todimensjonal ramme. Resultatene fra tidligere innlevert øving og programmets resultater stemmer overens. Det må riktignok nevnes at programmet, samt matrisemetoden har enkelte begrensninger. Begge neglisjerer både tverr- og aksialdeformasjoner. Oppgaven lyder dessuten som følgende:

Det kreves ikke at programmet skal beregne bøyemomenter for tilfeller hvor en bjelke utsatt både for punktlast og fordelt last [4].

Dermed har vi ikke tatt høyde for dette i kodingen av programmet og lastene må derfor være enten fordelte eller punktvis på et hvert element for å få riktige resultater. I vårt tilfelle har vi en jevnt fordelt last kombinert med lineært fordelt last over 3 elementer. Med så små avvik i resultatene mellom Nauticus® 3D Beam og MATLAB-programmet vårt er vi meget fornøyde med det endelige resultatet fra rammeanalysen.

6 Gruppeevaluering

Utleveringsdatoen til prosjektet var 05.10.2016 og innleveringsfristen 06.11.2016. Alt arbeid har foregått i fellesskap. For å få en oversikt over hele oppgaven brukte vi flytdiagram for å visualisere hvordan vi ønsket at funksjonene skulle kalle på hverandre. Brorparten av tiden har blitt brukt til å feilsøke MATLAB-kodene vi har skrevet. Under hele prosessen har verdiene våre blitt sammenlignet med det vi har modellert i Nauticus® 3D Beam. Itereringsprosessen gikk relativt fort. Et godt utgangspunkt lot oss komme frem til ønskede verdier med få iterasjoner. Da vi omsider var sikre på at resultatene stemte overens gikk vi i gang på rapportskrivingen. Arbeidet ble fordelt og vi tok for oss delene vi selv har jobbet mest med. Denne fremgangsmåten har fungert veldig bra for oss som en gruppe. Fordelen med å sitte sammen og jobbe er at vi raskt og enkelt kan bli enige om rapportens utforming, og dermed rette opp eventuelle uklarheter underveis. Mye tid har blitt brukt til diskusjon for å komme frem til et resultat som vi er godt fornøyd med.

De største utfordringene med dette prosjektet har for oss vært å holde orden på all informasjonen som er gitt. Med mange funksjoner og variabler har det vært helt kritisk å holde tungen rett i munnen. For å holde orden har vi aktivt brukt håndskrevne illustrasjoner, tabeller og tegninger helt fra starten av. Dette har vært til stor nytte og gjort det lettere for oss å lage gode variabel- og funksjonsnavn. Vi har hele tiden prøvd å holde koden og innholdet i rapporten kortfattet slik at vi unngår unødvendig informasjon. Slutten av prosjektarbeidet har blitt brukt til å fremstille resultatene våre i denne rapporten ved bruk av L^AT_EX.

Referanser

- [1] Institutt for Marin Teknikk, Fakultet for ingeniørvitenskap og teknologi, NTNU, *Kompendium for bruk i faget TMR 4167 Marin Teknikk 2*, 2010.
- [2] Institutt for Konstruksjonsteknikk, Fakultet for ingeniørvitenskap og teknologi, NTNU, *Formelsamling for konstruksjonsmekanikk - Fasthetslære*
- [3] Amdahl, Jørgen. Professor, Institutt for Marin Teknikk, Fakultet for ingeniørvitenskap og teknologi, NTNU.
- [4] Institutt for Marin Teknikk, Fakultet for ingeniørvitenskap og teknologi, NTNU, TMR 4167, Faggruppe for marine konstruksjoner *Karaktergivende prosjektoppgave: Rammearbeid med matrisemetoden*

A MATLAB-filer

A.1 rammeanalyse.m

```

1  % ----- RAMMEANALYSE ----- %
2
3  % Sletter alle variabler
4  clear all
5  clc
6
7  % ----- Leser input-data ----- %
8  [nKnutepunkt, Knutepunkt, nElem, Elem, nLast, Last, nProfil, Profil] =
    lesinput();
9
10 % ----- Regner lengder til elementene ----- %
11 elementlengder = lengder(Knutepunkt, nElem, Elem);
12
13 % ----- Fastinnspenningsmomentene ----- %
14 fim = moment(nElem, nLast, Last, elementlengder);
15
16 % ----- Setter opp lastvektor ----- %
17 b = lastvektor(nKnutepunkt, nElem, Elem, fim);
18
19 % ----- Annet arealmoment ----- %
20 [Im, yc] = I(Profil, nElem, Elem);
21
22 % ----- Setter opp systemstivhetsmatrisen ----- %
23 K = stivhet(nKnutepunkt, nElem, Elem, elementlengder, Im);
24
25 % ----- Innfoerer randbetingelser -----
26 [Kn, R] = randbetingelse(nKnutepunkt, Knutepunkt, b, K);
27
28 % ----- Loeser ligningssystemet -----
29 rot = Kn\R;
30
31 % ----- Finner endemoment for hvert element -----
32 endemoment = endeM(nElem, Elem, elementlengder, fim, Im, rot);
33
34 % ----- Finner midtmoment for hvert element -----
35 midtmoment = midtM(nElem, nLast, Last, elementlengder, endemoment);
36
37 % ----- Finner boeyespenning for hvert element -----
38 spenning = boeyespenning(nElem, Im, yc, endemoment, midtmoment);
39
40 % ----- Finner skjaerspenning for hvert element -----
41 skjaer = shear(nElem, nLast, Last, elementlengder, endemoment);
42
43 % ----- Total momentmatrise (svarmatrise) ----- %
44 totmom=[];
45 totmom(:,1) = endemoment(:,1).*10^(-3); % Henter inn endemomenter ende
    1
46 totmom(:,2) = endemoment(:,2).*10^(-3); % Henter inn endemomenter ende
    2
47 totmom(:,3) = midtmoment.*10^(-3);      % Henter inn endemomenter
48 totmom(:,4) = spenning(:,1).*10^(-6);    % Henter inn boeyespenninger i

```

```

    ende 1
49 totmom(:,5) = spenning(:,2).*10^(-6); % Hennter inn boeyespenninger i
    ende 2
50 totmom(:,6) = spenning(:,3).*10^(-6); % Hennter inn boeyespenninger i
    midten
51 totmom(:,7) = skjaer(:,1).*10^(-3); % Henter inn skjaerkraft i ende
    1
52 totmom(:,8) = skjaer(:,2).*10^(-3); % Henter inn skjaerkraft i ende
    2

53
54 % ----- Lager svarfil ----- %
55
56 % Lager en tekstfil med de viktigste resultatene
57 filidl = fopen('result.txt','w');
58
59 fprintf(filidl,'----- RESULTATER
    -----\n');
60 fprintf(filidl,'----- FOR RAMMEKONSTRUKSJONEN
    -----\n');
61
62 % Printer ut momenter som tabell
63 fprintf(filidl,'\nMOMENT\n');
64 % Finner verdi og posisjon til stoerste endemoment
65 [max_num1 idx1] = max(abs(endemoment(:)));
66 [row1 column1] = ind2sub(size(endemoment), idx1);
67 fprintf(filidl,'Absoluttverdien til maksimalt endemoment er:\n%4.2f [
    kNm] paa element %i.\n', max_num1.*10^(-3), row1);
68 fprintf(filidl,'
    -----\n');
69 fprintf(filidl,'Element      Ende 1 [kNm]      Midten [kNm]      Ende 2 [
    kNm]\n');
70 fprintf(filidl,'
    -----\n');
71 for i = 1:nElem
72 fprintf(filidl,'%4i          %10.2f          %10.2f          %8.2f\n',i,
    totmom(i,1), totmom(i,3), totmom(i,2));
73 end % for
74 fprintf(filidl,'
    -----\n');
75
76 % Printer ut spenninger som tabell
77 fprintf(filidl,'\n\nSPENNINGER\n');
78 % Finner verdi og posisjon til stoerste boeyespenning
79 [max_num2, idx2] = max(abs(spenning(:)));
80 [row2, column2] = ind2sub(size(spenning), idx2);
81 fprintf(filidl,'Absoluttverdien til maksimal boeyespenning er:\n%4.3f [
    MPa] paa element %i.\n', max_num2.*10^(-6), row2);
82 fprintf(filidl,'
    -----\n');
83 fprintf(filidl,'Element      Ende 1 [Mpa]      Midten [Mpa]      Ende 2 [
    MPa]\n');
84 fprintf(filidl,'
    -----\n');
85 for i = 1:nElem
86 fprintf(filidl,'%4i          %11.3f          %11.3f          %11.3f\n',i, totmom
    (i,4), totmom(i,6), totmom(i,5));
87 end % for

```

```
88 fprintf(filidl, '
    -----\n');
89
90 % Printer ut skjaerkrefter som tabell
91 fprintf(filidl, '\n\nSKJAERKREFTER\n');
92 % Finner verdi og posisjon til stoerste skjaerkraft
93 [max_num3, idx3] = max(abs(skjaer(:)));
94 [row3, column3] = ind2sub(size(skjaer), idx3);
95 %[max_num3, max_idx3] = max(skjaer(:));
96 fprintf(filidl, 'Absoluttverdien til maksimal skjaerkraft er:\n%4.1f [kN
    ] paa element %i, kolonne %i.\n', max_num3.*10^(-3), row3, column3)
    ;
97 fprintf(filidl, '-----\n');
98 fprintf(filidl, 'Element      Ende 1 [kN]      Ende 2 [kN]\n');
99 fprintf(filidl, '-----\n');
100 for i = 1:nElem
101 fprintf(filidl, '%4i          %8.1f          %7.1f \n', i, totmom(i,7),
    totmom(i,8));
102 end % for
103 fprintf(filidl, '-----\n');
104
105 % Lukker svarfil
106 fclose(filidl);
107 disp('Analyse gjennomfoert. Svarene finner du i result.txt')
108 disp('Denne filen maa aapnes i MATLAB')
```


A.2 lesinput.m

```
1 function [nKnutepunkt, Knutepunkt, nElem, Elem, nLast, Last, nProfil,  
    Profil] = lesinput()  
2 % Leser inputverdiene  
3  
4 % i = heltall (integer)  
5 % f : desimaltall (flyt-tall)  
6  
7 % Aapner input-filen  
8 fid = fopen('input.txt', 'r');  
9  
10 % Leser inn hvor mange knutepunkt systemet har og lager en 1x1 matrise  
    med  
11 % ett hel-tall  
12 nKnutepunkt = fscanf(fid, '%i', [1,1]);  
13  
14 % Lager en matrise med 3 kolonner og n rader med antall knutepunkter  
15 Knutepunkt = fscanf(fid, '%f', [3 nKnutepunkt]);  
16 % Leser inn koordinatene til hvert knutepunkt  
17 % Kolonne 1: x-koordinater  
18 % Kolonne 2: y-koordinater  
19 % Kolonne 3: Informasjon om opplager;  
20 %     0 = fritt opplagt  
21 %     1 = fast innspent  
22  
23 % Leser inn hvor mange elementer systemet har og lager en 1x1 matrise  
    med  
24 % ett hel-tall  
25 nElem = fscanf(fid, '%i', [1,1]);  
26  
27 % Lager en matrise med 4 kolonner og n rader med antall elementer  
28 Elem = fscanf(fid, '%i %i %f %i', [4 nElem]);  
29 % Kolonne 1: lokalt knutepunkt 1  
30 % Kolonne 2: lokalt knutepunkt 2  
31 % Kolonne 3: E-modul [GPa]  
32 % Kolonne 4: Informasjon om elementets orientering;  
33 %     0 = vannrett, I-profil  
34 %     1 = loddrett, roerprofil  
35  
36 % Leser inn hvor mange laster som virker paa systemet og lager en 1x1  
    matrise med  
37 % ett hel-tall  
38 nLast = fscanf(fid, '%i', [1,1]);  
39  
40 % Lager en matrise med 4 kolonner og n rader med antall laster  
41 Last = fscanf(fid, '%i %f %i %f', [4 nLast]);  
42 % Kolonne 1: Informasjon om hva type last;  
43 %     1 = paasatt punktlast  
44 %     2 = jevnt fordelt last  
45 %     3 = lineært fordelt trekanlast  
46 % Kolonne 2: Verdien til kraften [N]  
47 % Kolonne 3: Elementet kraften virker paa  
48 % Kolonne 4: Ulike verdier avhengig av type last;  
49 %     Punktlast: Alfa-verdi for gitt last [-]  
50 %     Jevnt fordelt last: Samme verdi i begge knutepunkter. Satt lik
```

```
1
51 % Lineært fordelt last: Oppgir om maksverdi opprter i lokalt ...
52 % knutepunkt 1 eller 2
53
54 % Leser inn hvor mange profiler systemet har og lager en 1x1 matrise
    med
55 % ett hel-tall
56 % Positive retninger: Ned og hoeyre
57 nProfil = fscanf(fid, '%i', [1,1]);
58
59 % Lager en matrise med 7 kolonner og n rader med antall profiler
60 Profil = fscanf(fid, '%i %f %f %f %f %f %f', [7 nProfil]);
61
62 % Kolonne 1: Type profil;
63 % 0 = I-profil
64 % 1 = roerprofil
65 % Kolonne 2: Informasjon om profilets dimensjoner;
66 % I-profil: Hoejde steg [m]
67 % Roerprofil: Diameter [m]
68 % Kolonne 3: Informasjon om profilets dimensjoner;
69 % I-profil: Tykkelse steg [m]
70 % Roerprofil: Tykkelse [m]
71 % Kolonne 4: Informasjon om profilets dimensjoner;
72 % I-profil: Tykkelse toppflens [m]
73 % Roerprofil: Ingen informasjon. Satt lik 0
74 % Kolonne 5: Informasjon om profilets dimensjoner;
75 % I-profil: Bredde toppflens [m]
76 % Roerprofil: Ingen informasjon. Satt lik 0
77 % Kolonne 6: Informasjon om profilets dimensjoner;
78 % I-profil: Tykkelse bunnflens [m]
79 % Roerprofil: Ingen informasjon. Satt lik 0
80 % Kolonne 7: Informasjon om profilets dimensjoner;
81 % I-profil: Bredde bunnflens [m]
82 % Roerprofil: Ingen informasjon. Satt lik 0
83
84 % Lukker input-filen
85 fclose(fid);
86 end % function
```

A.3 lengder.m

```
1 function svar = lengder(Knutepunkt, nElem, Elem)
2 % For beregning av elementenes lengder
3
4 svar = zeros(nElem,1); % Lager tom svarmatrise
5
6 for i = 1:nElem
7     x = Knutepunkt(Elem(i,1), 1) - Knutepunkt(Elem(i,2), 1);
8     y = Knutepunkt(Elem(i,1), 2) - Knutepunkt(Elem(i,2), 2);
9 % For aa ta hoeyde for eventuelle skraa bjelker brukes pytagoras
10    svar(i) =sqrt((x^2) + (y^2));
11 end % for
12 end % function
```

A.4 moment.m

```

1 function svar = moment(nElem, nLast, Last, elementlengder)
2 % For beregning av fastinnspenningsmoment
3
4 svar = zeros(2, nElem); % Lager tom svarmatrise
5
6 % Gaar gjennom lastene og finner fastinnspenningsmomentene grunnet ...
7 % punktlast, jevnt fordelt last og lineaar last
8 for i = 1:nLast
9     if Last(i,1) == 1 % Gjelder punktLaster
10         L = elementlengder(Last(i,3));
11         a = Last(i,4) * L; % Avstand fra lokalt knutepunkt 1 til lasten
12         b = L - a; % Avstand fra lokalt knutepunkt 2 til lasten
13         % Benytter formel fra side 281 i kompendiet del 1
14         moment_ab = (-(Last(i,2)) * a * (b)^2) / L^2;
15         moment_ba = (Last(i,2) * (a)^2 * b) / L^2;
16
17         % Plasserer svaret paa riktig plass
18         svar(1,Last(i,3)) = svar(1,Last(i,3)) + moment_ab;
19         svar(2,Last(i,3)) = svar(2,Last(i,3)) + moment_ba;
20
21     elseif Last(i,1) == 2 % Gjelder jevnt fordelte laster
22         L = elementlengder(Last(i,3));
23         % Benytter formel fra side 281 i kompendiet del 1
24         moment_ab = (-Last(i,2) * L^2) / 12;
25         moment_ba = -moment_ab;
26
27         % Plasserer svaret paa riktig plass
28         svar(1,Last(i,3)) = svar(1,Last(i,3)) + moment_ab;
29         svar(2,Last(i,3)) = svar(2,Last(i,3)) + moment_ba;
30
31     elseif Last(i,1) == 3 % Gjelder lineært fordelte laster
32         L = elementlengder(Last(i,3));
33         % Benytter formel fra side 281 i kompendiet del 1
34         moment_1 = (-Last(i,2) * (L)^2) / 30;
35         moment_2 = (Last(i,2) * (L)^2) / 20;
36
37         if Last(i,4) == 1 % Sjekker om maksimum er i lokal ende 1
38             moment_ab = moment_2;
39             moment_ba = moment_1;
40         else % Beregner for maksimum i lokal ende 2
41             moment_ab = moment_1;
42             moment_ba = moment_2;
43         end %if
44
45         % Plasserer svaret paa riktig plass
46         svar(1,Last(i,3)) = svar(1,Last(i,3)) + moment_ab;
47         svar(2,Last(i,3)) = svar(2,Last(i,3)) + moment_ba;
48     end % if
49 end %for
50 end %function

```

A.5 lastvektor.m

```
1 function svar = lastvektor(nKnutepunkt, nElem, Elem, fim)
2 % For beregning av lastvektoren
3 % R = Rc - Ro
4 % Tar ikke hensyn til paasatte momenter; Rc = 0
5
6 svar = zeros(nKnutepunkt,1); % Lager tom svarmatrise
7
8 for i=1:nElem
9
10     % Legger til -summen av inspenningmomenter i lastvektoren i
11     % henhold til
12     % formel
13     svar(Elem(i,1)) = svar(Elem(i,1)) - fim(1,i);
14     svar(Elem(i,2)) = svar(Elem(i,2)) - fim(2,i);
15
16 end %for
17 end % function
```

A.6 I.m

```

1 function [Im, yc] = I(Profil, nElem, Elem)
2
3 Im = zeros(nElem,1); % Lager tom svarmatrise
4 yc = zeros(nElem,1); % Lager tom svarmatrise
5
6 for i = 1:nElem
7     if Elem(i,4) == 0 % Om elementet er vannrett er det I-profil
8         % Regner ut profilens fysiske verdier
9         hoyde_tot = (Profil(1,6) + Profil(1,2) + Profil(1,4));
10        a_toppfLens = Profil(1,4) * Profil(1,5);
11        a_steg = Profil(1,2) * Profil(1,3);
12        a_bunnflens = Profil(1,6) * Profil(1,7);
13        a_total = a_steg + a_toppfLens + a_bunnflens;
14
15        % Avstand fra bunn til de forskjellige arealsentrene
16        y_toppfLens = Profil(1,6) + Profil(1,2) + Profil(1,4) / 2;
17        y_steg = Profil(1,6) + Profil(1,2) / 2;
18        y_bunnflens = Profil(1,6) / 2;
19
20        % Regner ut om arealsenter om tverrsnittet er symmetrisk eller
21        ikke
22        if (Profil(1,6) ~= Profil(1,4)) || (Profil(1,7) ~= Profil
23            (1,5))
24            yc(i) = (a_bunnflens * y_bunnflens + y_steg * a_steg
25                ...
26                + a_toppfLens * y_toppfLens) / a_total;
27        else
28            yc(i) = hoyde_tot / 2;
29        end % if
30
31        % Regner ut annet arealmoment for hver del og legger til ...
32        % Steiners tillegg. Summerer saa for aa finne totalt annet ...
33        % arealmoment
34        aam_topp = ((Profil(1,5) * Profil(1,4)^3) / 12) +...
35            ((hoyde_tot - yc(i) - Profil(1,4)/2)^2 * a_toppfLens);
36
37        aam_steg = (Profil(1,3) * Profil(1,2)^3) / 12 + (abs(yc(i) -
38            y_steg)^2 *...
39            a_steg);
40
41        aam_bunn = ((Profil(1,7) * Profil(1,6)^3) / 12) +...
42            ((yc(i) - Profil(1,6) / 2)^2 * a_bunnflens);
43
44        aamI = (aam_topp + aam_steg + aam_bunn);
45        Im(i) = aamI;
46    end % if
47
48    if Elem(i,4) == 1 % Om elementet er loddrett er det roerprofil
49        R = Profil(2,2) / 2; % ytterdiameter til roer
50        r = R - Profil(2,3); % innerdiameter til roer
51        % Regner ut annet arealmoment av roerprofilen
52        aamR = ((pi * (R^4 - r^4)) / 4);
53        Im(i) = aamR;
54        % Passer ap at diameteren er stoerre enn tykkelsen

```

```
51         if Profil(2,2) < Profil(2,3)
52             error('Diameter er mindre enn tykkelse.');
```

53 elseif Profil(2,2) == Profil(2,3);

54 error('Diameter kan ikke være samme som tykkelse.');

55 end % if

56 yc(i) = R;

57 end % if

58 end % for

59 end % function

A.7 stivhet.m

```
1 function svar = stivhet(nKnutepunkt, nElem, Elem, elementlengder, Im)
2 % Lager stivhetsmatrise
3
4 % Setter opp kvadratisk matrise med antall knutepunkt
5 C = zeros(nKnutepunkt, nKnutepunkt);
6 k = [4 2; 2 4]; % Standard stivhetsmatrise
7 L = elementlengder; % Elementets lengde
8 E = Elem(:,3); % Elementets E-modul
9
10 for i = 1:nElem
11
12     KK = ((E(i) * Im(i)) / L(i)) * k;
13     C(Elem(i,1), Elem(i,1)) = C(Elem(i,1), Elem(i,1)) + KK(1,1);
14     C(Elem(i,1), Elem(i,2)) = C(Elem(i,1), Elem(i,2)) + KK(1,2);
15     C(Elem(i,2), Elem(i,1)) = C(Elem(i,2), Elem(i,1)) + KK(2,1);
16     C(Elem(i,2), Elem(i,2)) = C(Elem(i,2), Elem(i,2)) + KK(2,2);
17 end % for
18 svar = C;
19 end % function
```


A.8 randbetingelse.m

```
1 function [Kn, R] = randbetingelse(nKnutepunkt, Knutepunkt, b, K)
2
3 betingelse = Knutepunkt(:,3);
4 Kn = K;
5 R = b;
6
7 for i = 1:nKnutepunkt
8     if betingelse(i) == 1 % Sjekker om punktet har fast innspenning
9         Kn(:,i) = 0;      % Nuller ut kolonner
10        Kn(i,:) = 0;      % Nuller ut rader
11        Kn(i,i) = 1;      % Setter vilkaarlig tall (1) ut ifra
                           oppgavetekst
12        R(i) = 0;        % Tilhoerende plass i lastvektoren settes lik
                           null ihht oppgavetekst
13    end %if
14 end %for
15 end %function
```

A.9 endeM.m

```
1 function svar = endeM(nElem, Elem, elementlengder, fim, Im, rot)
2 % For beregning av endemoment
3
4 svar = zeros(nElem,2); % Lager tom svarmatrise
5 k = [4 2 ; 2 4];      % Lager lokal stivhetsmatrise
6 L = elementlengder;   % Lengden til hvert element
7 E = Elem(:,3);        % Elementnummer
8
9 for i = 1:nElem
10
11     r = [rot(Elem(i,1)) ; rot(Elem(i,2))]; % Lager lokal
12         rotasjonsmatrise
13     m = [fim(1,i) ; fim(2,i)];             % Lager
14         fastinnspenningsmatrise
15     KK = ((E(i) * Im(i)) / L(i)) * k;      % 4EI / L
16     M = (KK * r) + m;
17
18     svar(i,1) = -M(1);
19     svar(i,2) = -M(2);
20
21 end % for
22 end % function
```

A.10 midtM.m

```

1 function svar = midtM(nElem, nLast, Last, elementlengder, endemoment)
2 % For beregning av endemoment
3
4 svar = zeros(nElem,1); % Lager tom svarmatrise
5
6 for i = 1:nLast
7     E = Last(i,3); % Elementnummer
8     L = elementlengder(E); % Lengden til hvert element
9     P = Last(i,2); % Verdien / maksverdien til lastens kraft
10
11     if Last(i,1) == 1 % Gjelder punktlaster
12         a = Last(i,4) * L; % Avstand fra lokalt knutepunkt 1 til
13                             lasten
14         b = L - a; % Avstand fra lokalt knutepunkt 2 til
15                             lasten
16
17         % Bidrag fra endemomentene
18         if abs(endemoment(E,2)) >= abs(endemoment(E,1))
19             M2 = endemoment(E,1) - (endemoment(E,1) + endemoment(E,2))
20                 * Last(i,4);
21         else
22             M2 = -endemoment(E,2) + (endemoment(E,1) + endemoment(E,2))
23                 * (1-Last(i,4));
24         end % if
25
26         % Bidrag fra punktlasten
27         M1 = (P * a * b) / L;
28         svar(E,1) = - M1 + M2 + svar(E,1);
29
30     elseif Last(i,1) == 2 % Jevnt fordelt last
31         % Bidrag fra jevnt fordelt last
32         M1 = (-P * (L^2)) / 8;
33         % Bidrag fra endemomentene
34         M2 = (endemoment(E,1) - endemoment(E,2)) / 2;
35         svar(E,1) = M1 + M2 + svar(E,1);
36
37     elseif Last(i,1) == 3 % Gjelder lineært fordelt last
38         % Samme midtmoment uansett hvilken ende maks moment er i
39         % Bidrag fra lineært fordelt last
40         M1 = (-P * (L^2)) / 16;
41         % Bidrag fra endemomentene er lagt til i jevnt fordelt last
42         svar(E,1) = M1 + svar(E,1);
43     end % if
44 end % for
45 end % function

```

A.11 boyespenning.m

```
1 function svar = boyespenning(nElem, Im, yc, endemoment, midtmoment)
2 % For beregning av boeyespenning
3
4 svar = zeros(nElem,3); % Lager tom svarmatrise
5
6 for i = 1:nElem
7
8     %Regner ut spenning og legger den inn i matrisen elementvis
9     svar(i,1) = endemoment(i,1) * yc(i) / Im(i);
10    svar(i,2) = endemoment(i,2) * yc(i) / Im(i);
11    svar(i,3) = midtmoment(i,1) * yc(i) / Im(i);
12
13 end % for
14 end % function
```

A.12 shear.m

```

1 function svar = shear(nElem, nLast, Last, elementlengder, endemoment)
2 % Beregner skjaerkraft
3
4 svar = zeros(nElem,2);      % Lager tom svarmatrise
5 L = elementlengder(:,1);    % Lengden til hvert element
6 P = Last(:,2);              % Verdien / maksverdien til lastens kraft
7
8 % Skjaerkraft grunnet endemomenter
9 for i = 1:nElem
10     svar(i,1) = (endemoment(i,1) + endemoment(i,2)) / L(i);
11     svar(i,2) = (endemoment(i,1) + endemoment(i,2)) / L(i);
12 end % for
13
14 % Skjaerkraft grunnet laster
15 for i = 1:nLast
16     E = Last(i,3);
17     if Last(i,1) == 1 % Punktlast
18         svar(E,1) = svar(E,1) + (P(i) * (1 - Last(i,4)));
19         svar(E,2) = svar(E,2) - (P(i) * Last(i,4));
20
21     elseif Last(i,1) == 2 % Jevnt fordelt last
22         svar(E,1) = svar(E,1) + ((P(i) * L(E)) / 2);
23         svar(E,2) = svar(E,2) - ((P(i) * L(E)) / 2);
24
25     elseif Last(i,1) == 3 % Lineaert fordelt last
26         % Ser hvilken ende den har stoerst verdi
27         if Last(i,4) == 1 % Maks i lokalt knutepunkt 1
28             svar(E,1) = svar(E,1) + ((P(i) * L(E)) / 3);
29             svar(E,2) = svar(E,2) - ((P(i) * L(E)) / 6);
30         elseif Last(i,4) == 2 % Maks i lokalt knutepunkt 2
31             svar(E,1) = svar(E,1) + ((P(i) * L(E)) / 6);
32             svar(E,2) = svar(E,2) - ((P(i) * L(E)) / 3);
33         end % if
34     end % if
35 end % for
36
37 end

```

B Inputfiler for MATLAB

B.1 input.txt

```
1 12
2 0 0 1
3 0 12 0
4 0 24 0
5 0 36 0
6 18 36 0
7 18 24 0
8 18 12 0
9 18 0 0
10 38 0 1
11 38 12 0
12 38 24 0
13 38 36 0
14 15
15 1 2 210e9 1
16 2 3 210e9 1
17 3 4 210e9 1
18 4 5 210e9 0
19 3 6 210e9 0
20 2 7 210e9 0
21 8 7 210e9 1
22 7 6 210e9 1
23 6 5 210e9 1
24 5 12 210e9 0
25 6 11 210e9 0
26 7 10 210e9 0
27 9 10 210e9 1
28 10 11 210e9 1
29 11 12 210e9 1
30 10
31 1 200e3 5 0.4
32 1 300e3 11 0.4
33 1 500e3 6 0.6
34 1 400e3 12 0.5
35 2 80e3 1 1
36 3 26.6667e3 1 2
37 2 106.6667e3 2 1
38 3 26.6667e3 2 2
39 2 133.3333e3 3 1
40 3 26.6667e3 3 2
41 2
42 0 0.750 0.02 0.025 0.3 0.025 0.3
43 1 0.700 0.03 0 0 0 0
```

B.2 result.txt

```

1 ----- RESULTATER -----
2 ----- FOR RAMMEKONSTRUKSJONEN -----
3
4 MOMENT
5 Absoluttverdien til maksimalt endemoment er:
6 2129.75 [kNm] paa element 2.
7
8 Element      Ende 1 [kNm]      Midten [kNm]      Ende 2 [kNm]
9 -----
10 1             908.28           -470.14           -1511.45
11 2             809.54           -690.36           -2129.75
12 3            1915.59          -1332.32           -699.78
13 4             699.78             0.00             291.09
14 5             214.16          -513.08           -556.05
15 6             701.91          -1086.70          -1320.89
16 7             -0.00             0.00             101.17
17 8             64.84             0.00             -72.66
18 9            -210.71             0.00            -211.27
19 10            -79.82             0.00             -53.81
20 11            839.42           -728.28           -520.17
21 12           1154.87          -1026.89           -791.34
22 13            175.11             0.00             350.22
23 14            441.12             0.00             356.91
24 15            163.26             0.00              53.81
25 -----
26
27
28 SPENNINGER
29 Absoluttverdien til maksimal boeyespenning er:
30 209.953 [MPa] paa element 2.
31
32 Element      Ende 1 [Mpa]      Midten [MPa]      Ende 2 [MPa]
33 -----
34 1             89.539           -46.347           -149.000
35 2             79.805           -68.056           -209.953
36 3            188.841          -131.341           -68.985
37 4             94.685             0.000             39.387
38 5             28.977           -69.424           -75.237
39 6             94.973          -147.038          -178.725
40 7             -0.000             0.000             9.973
41 8              6.392             0.000             -7.163
42 9            -20.772             0.000            -20.828
43 10            -10.800             0.000             -7.281
44 11            113.579          -98.541           -70.382
45 12            156.262          -138.945          -107.073
46 13             17.262             0.000             34.525
47 14             43.486             0.000             35.185
48 15             16.094             0.000              5.304
49 -----
50
51
52 SKJAERKREFTER
53 Absoluttverdien til maksimal skjaerkraft er:
54 954.7 [kN] paa element 3, kolonne 1.

```

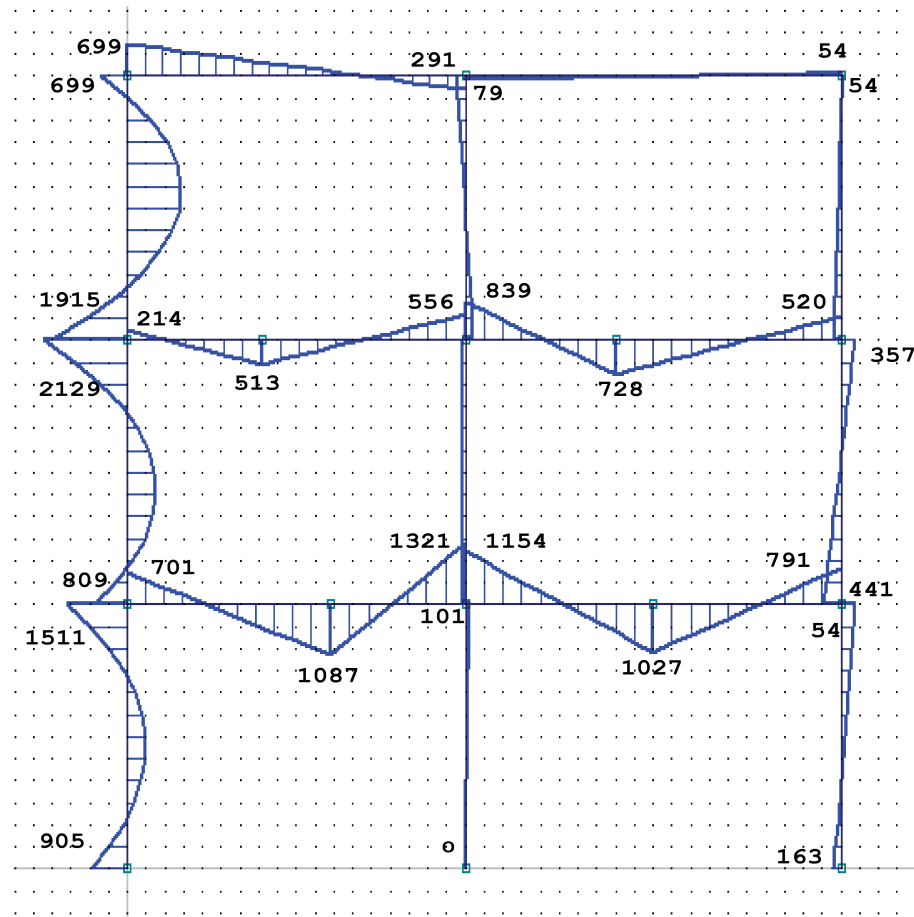
55	-----		
56	Element	Ende 1 [kN]	Ende 2 [kN]
57	-----		
58	1	483.1	-636.9
59	2	583.3	-856.7
60	3	954.7	-805.3
61	4	55.0	55.0
62	5	101.0	-99.0
63	6	165.6	-334.4
64	7	8.4	8.4
65	8	-0.7	-0.7
66	9	-35.2	-35.2
67	10	-6.7	-6.7
68	11	196.0	-104.0
69	12	218.2	-181.8
70	13	43.8	43.8
71	14	66.5	66.5
72	15	18.1	18.1
73	-----		

B.3 inputoving6.txt

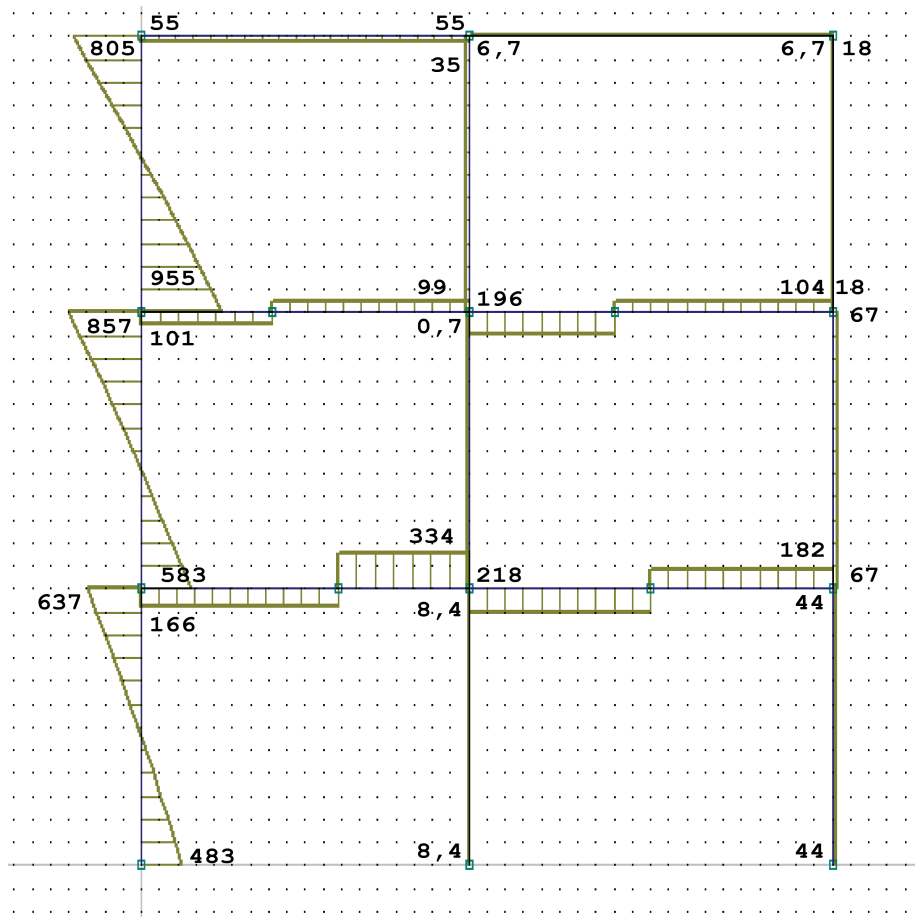
```
1 5
2 0 0 1
3 1 0 0
4 2 0 1
5 1 1.5 0
6 1 3 0
7 4
8 1 2 1 0
9 2 3 1 0
10 2 4 1 1
11 4 5 1 1
12 4
13 2 0.5 1 1
14 2 -1 2 1
15 2 1.333 3 1
16 2 0.667 4 1
17 1
18 1 400 8.6 13.5 180 13.5 180
```

C Diagrammer

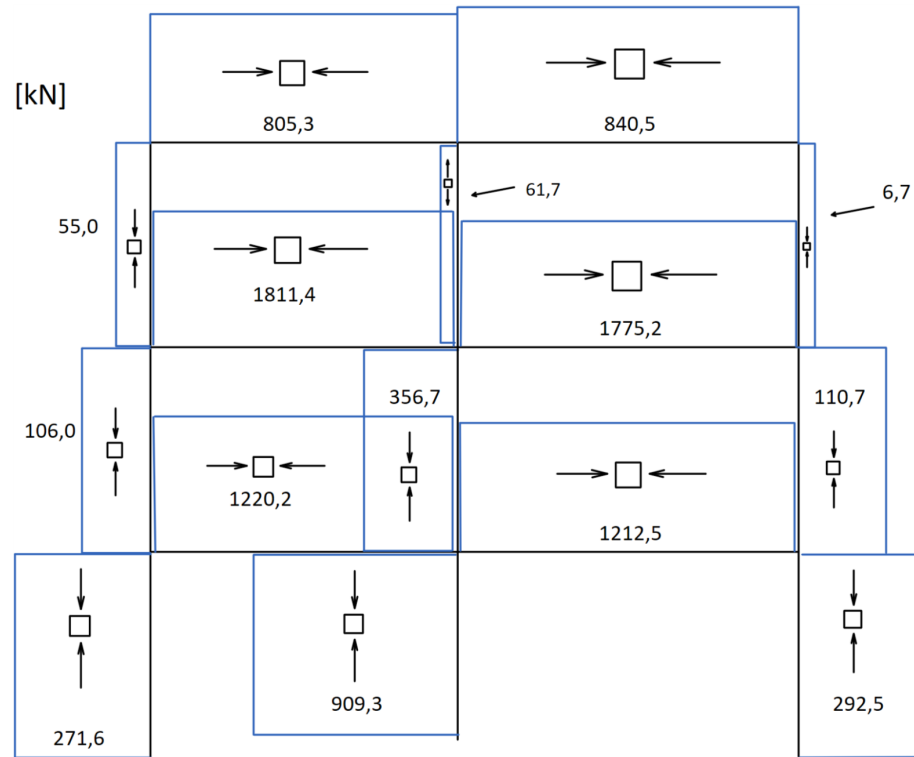
C.1 Momentdiagram



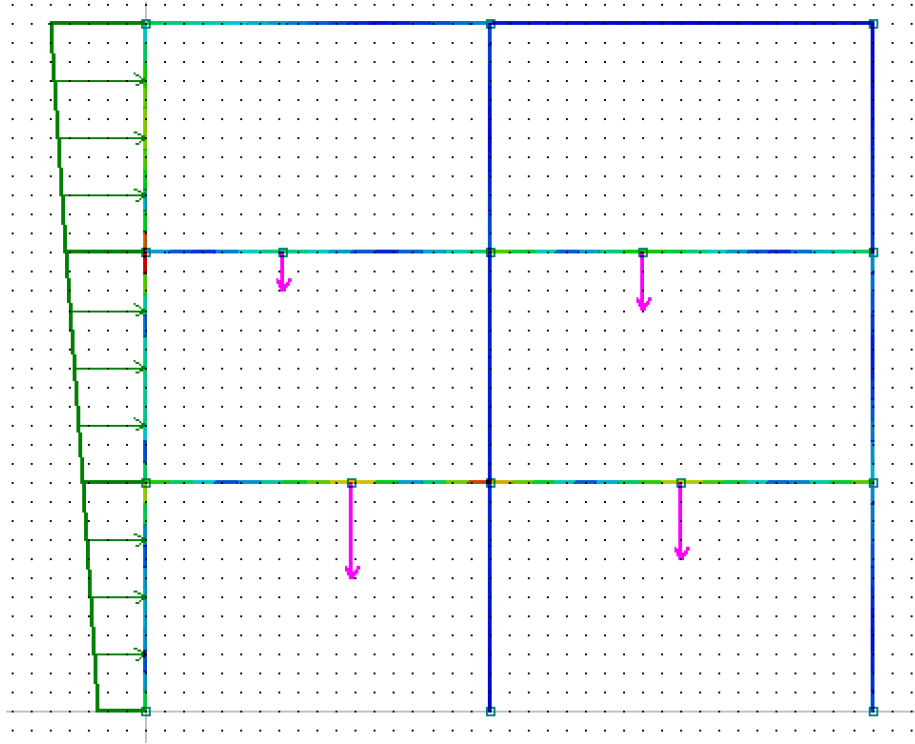
C.2 Skjærkraftdiagram



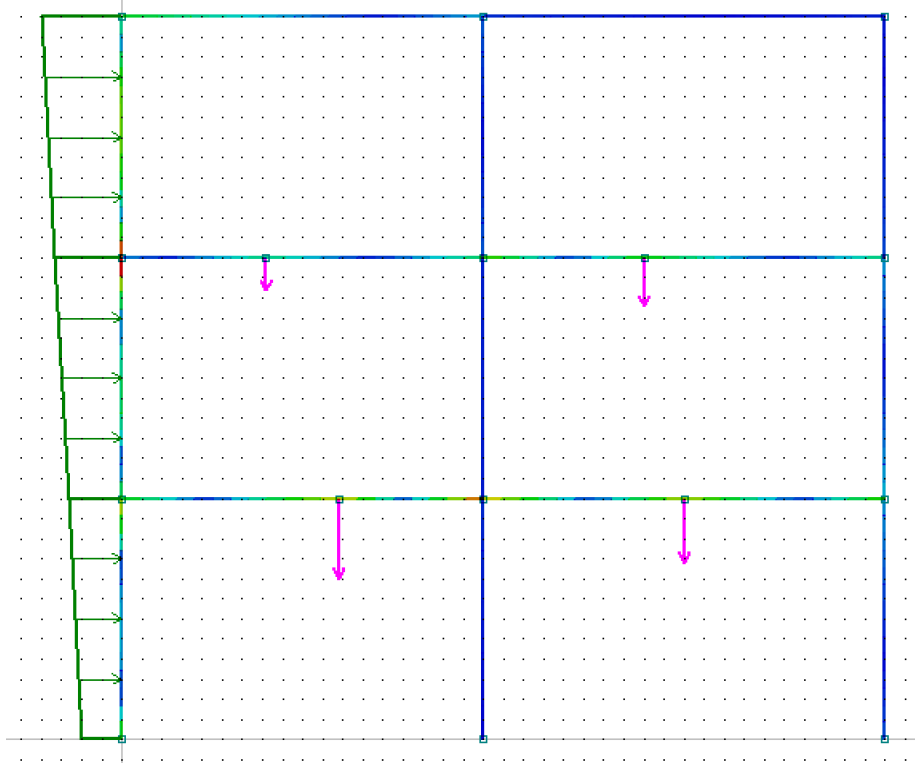
C.3 Aksialkraftdiagram



C.4 Spenningsfordeling ved originale tverrsnitt




C.5 Spenningsfordeling ved itererte tverrsnitt



D Resultater

D.1 Resultater fra Nauticus: Moment og skjærkrefter

	Report:	Model Id:	Sign:
	Beam Responses	NONAME	NN
		Description:	Date:
		Model file name:	Last saved:
		H:\Untitled2.clb	2016-11-02 15:55

Load case No 1 Load case #1 -


Forces, Moments and Deflections, Signed values, sorted by Beam in Ascending order

Beam No.	N_x [N]	Q_y [N]	Q_z [N]	M_x [Nmm]	M_y [Nmm]	M_z [Nmm]	δ [mm]	δ_x [mm]	δ_y [mm]	δ_z [mm]
1	0	0	54702	0	698928713	0	15,189	0	0	15,189
2	0	0	101173	0	-512854400	0	13,447	0	0	-13,447
3	0	0	165807	0	-1088564061	0	28,495	0	0	-28,495
4	0	0	8371	0	-100456819	0	1,2446	1,2446	0	0
5	0	0	-918	0	74768383	0	1,6721	-1,6721	0	0
6	0	0	-34555	0	-207678627	0	0,64609	0,64609	0	0
7	0	0	-6584	0	-78718566	0	1,465	0	0	-1,465
8	0	0	195804	0	836929603	0	22,22	0	0	-22,22
9	0	0	218116	0	1153420188	0	30,887	0	0	-30,887
10	0	0	43548	0	-350505272	0	2,5358	2,5358	0	0
11	0	0	18187	0	165280547	0	1,4768	-1,4768	0	0
12	0	0	-98827	0	554482594	0	13,451	0	0	-13,451
13	0	0	-334193	0	1317627723	0	27,603	0	0	-27,603
14	0	0	-104196	0	-729498729	0	22,827	0	0	-22,827
15	0	0	-181884	0	-1027740656	0	30,912	0	0	-30,912
16	0	0	-637221	0	1515338823	0	5,2578	5,2578	0	0
17	0	0	-855662	0	2121931202	0	9,1832	9,1832	0	0
18	0	0	953818	0	1906343597	0	22,889	22,889	0	0
19	0	0	66348	0	440593230	0	1,971	-1,971	0	0

Abbreviations

N_x : Axial force (Positive gives tension)
 Q_y : Shear force in local y-direction (Positive rotates an isolated piece clockwise)
 Q_z : Shear force in local z-direction (Positive rotates an isolated piece counter-clockwise)
 M_x : Torsional moment (Positive produces a right-handed screw)
 M_y : Bending moment about local y-axis (Positive gives tension at local positive Z-axis side of profile)
 M_z : Bending moment about local z-axis (Positive gives tension at local positive Y-axis side of profile)
 δ : Maximum total deflection of beam ($\sqrt{\delta_x^2 + \delta_y^2 + \delta_z^2}$)
 $\delta_x, \delta_y, \delta_z$: Maximum deflection of beam in global X-, Y-, and Z- directions

D.2 Resultater fra Nauticus: Rotasjoner

	Report:	Model Id:	Sign:
	Beam Responses	NONAME	NN
		Description:	Date:
		Model file name:	Last saved:
		H:\Untitled2.clb	2016-11-02 15:55

Load case No 1 Load case #1 -

Forces, Moments and Deflections, Signed values, sorted by Beam in Ascending order

Beam No.	N_x [N]	Q_y [N]	Q_z [N]	M_x [Nmm]	M_y [Nmm]	M_z [Nmm]	δ [mm]	δ_x [mm]	δ_y [mm]	δ_z [mm]
1	0	0	54702	0	698928713	0	15,189	0	0	15,189
2	0	0	101173	0	-512854400	0	13,447	0	0	-13,447
3	0	0	165807	0	-1088564061	0	28,495	0	0	-28,495
4	0	0	8371	0	-100456819	0	1,2446	1,2446	0	0
5	0	0	-918	0	74768383	0	1,6721	-1,6721	0	0
6	0	0	-34555	0	-207678627	0	0,64609	0,64609	0	0
7	0	0	-6584	0	-78718566	0	1,465	0	0	-1,465
8	0	0	195804	0	836929603	0	22,22	0	0	-22,22
9	0	0	218116	0	1153420188	0	30,887	0	0	-30,887
10	0	0	43548	0	-350505272	0	2,5358	2,5358	0	0
11	0	0	18187	0	165280547	0	1,4768	-1,4768	0	0
12	0	0	-98827	0	554482594	0	13,451	0	0	-13,451
13	0	0	-334193	0	1317627723	0	27,603	0	0	-27,603
14	0	0	-104196	0	-729498729	0	22,827	0	0	-22,827
15	0	0	-181884	0	-1027740656	0	30,912	0	0	-30,912
16	0	0	-637221	0	1515338823	0	5,2578	5,2578	0	0
17	0	0	-855662	0	2121931202	0	9,1832	9,1832	0	0
18	0	0	953818	0	1906343597	0	22,889	22,889	0	0
19	0	0	66348	0	440593230	0	1,971	-1,971	0	0

Abbreviations

N_x : Axial force (Positive gives tension)
 Q_y : Shear force in local y-direction (Positive rotates an isolated piece clockwise)
 Q_z : Shear force in local z-direction (Positive rotates an isolated piece counter-clockwise)
 M_x : Torsional moment (Positive produces a right-handed screw)
 M_y : Bending moment about local y-axis (Positive gives tension at local positive Z-axis side of profile)
 M_z : Bending moment about local z-axis (Positive gives tension at local positive Y-axis side of profile)
 δ : Maximum total deflection of beam ($\sqrt{\delta_x^2 + \delta_y^2 + \delta_z^2}$)
 $\delta_x, \delta_y, \delta_z$: Maximum deflection of beam in global X-, Y-, and Z- directions