		NTNU Norges teknisk-naturvitenskapelige universitet Institutt for marin teknikk Faggruppe for marine konstruksjoner		Karaktergivende Prosjektoppgave	
TMR4167 MARIN TEKNIKK - KONSTUKSJONER					
Dato: 06.10.21		Sign: JA		Utleveringsdato: 06.10.21	
				Innleveringsfrist: 09.11.21	

ANALYSE AV PLAN FORSKYVELIG RAMME MED MATRISEMETODEN

MÅL:

- 1) Trening i å ta i bruk programmeringsverktøy (Python) til å utvikle et enkelt, men realistisk dataprogram for lineær analyse av plane, forskyvelige konstruksjoner.
- 2) Økt forståelse av hvordan dataverktøy kan brukes i daglige ingeniørberegninger
- 3) Bruk av kunnskap ervervet i tidligere fag: Informasjonsteknologi grunnkurs, Matematikk og Mekanikk
- 4) Trening i gruppe - og prosjektarbeid
- 5) Økt innsikt i konstruksjoners virkemåte/respons på for ytre laster .

Prosjektarbeidet er karaktergivende og utgjør 25 % av totalkarakteren i faget.

Selv om dataprogrammet er enkelt, skal et vellykket utviklet program være nesten like korrekt som eksisterende dataprogram for rammeanalyse av marine konstruksjoner som brukes av ingeniørselskap og de som er tilgjengelige her (f.eks. Sesam, 3DBeam, Abaqus og Usfos (begge brukt lineært) etc.), men det utviklede Python-programmet vil bare kunne regne på plane konstruksjoner og ikke romlige. (utvidelse til 3D er relativt trivielt, men arbeidskrevende).

Programmet vil kunne anvendes på «grønne»* konstruksjoner (f.eks. vindturbiner til havs, eksponerte oppdrettsanlegg) like så vel som «sorte»* konstruksjoner (f.eks. plattformer som utvinner gass til kontinentet).

*Som dere sikkert kjenner til er det ulike meninger i opinionen om fargen på disse innretningene.

Det er godt mulig at programmet ofte vil anvendes til løsning av såkalte «tame problems», men det forsikres at gjennom prosjektarbeidet vil dere få et sterkt forbedret grunnlag til også å bidra til løsning av såkalte «wicked problems» (umedgjørilige problem).

Det skal lages en kortfattet rapport over arbeidet som bl.a. viser skisse av konstruksjon, valgte tverrsnittsdimensjoner, elementinndeling, knutepunksnummerering, benyttede lastverdier, kontroll av programmet i pkt c og resultatene i pkt d) og e).

Prosjektet skal utføres i grupper på 3 personer. Man står fritt til å organisere dette. Påmelding gjøres på BB hvor dere da også får et gruppenummer som skal benyttes til å bestemme inngangsdata. Hver gruppe leverer inn 1 prosjektrapport, med kandidatnummer til alle gruppe-medlemmer oppført på forsiden

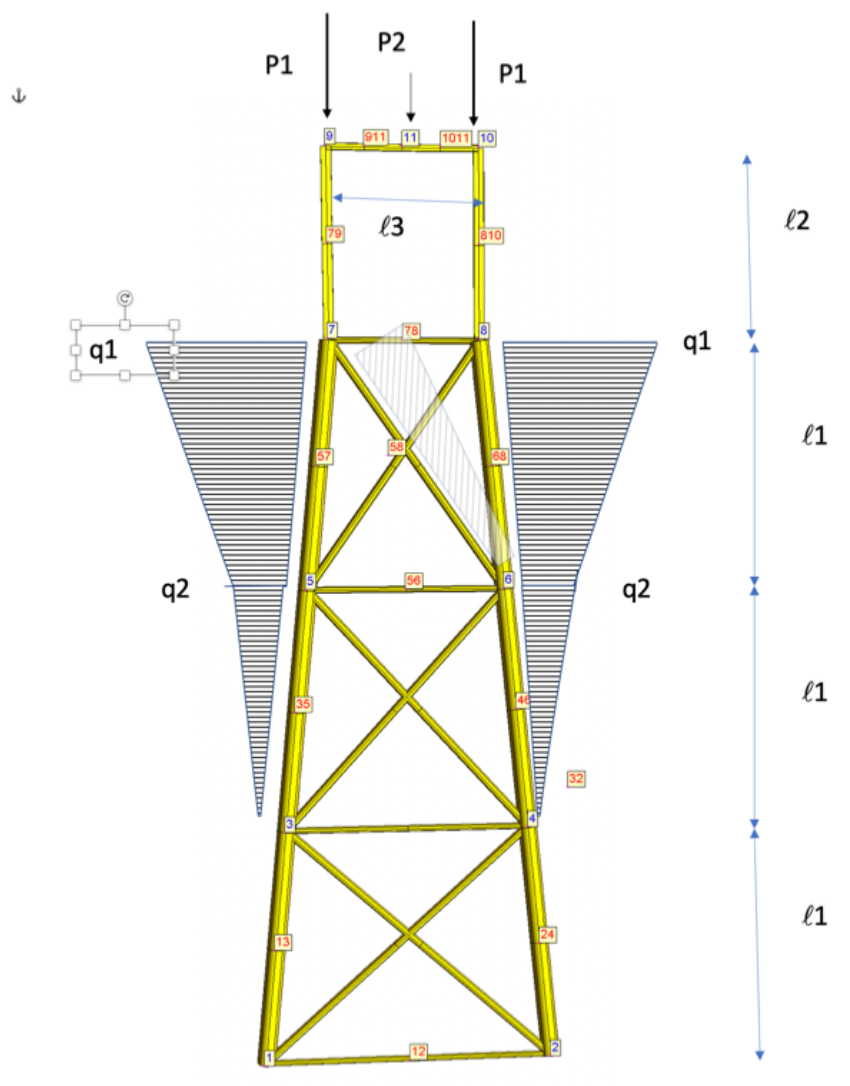
Oppgaven skal leveres både i papirform og på Blackboard. Legg ved en minnepenn med PYTHON-program med papiroppgaven, inklusive inngangsdata. På Blackboard legges PYTHON-programmet ved i en .zip-fil. Legg også til PYTHON-koden som vedlegg i prosjektrapporten.

Evalueringskriterier:

- *Strukturert rapport med gode figurer (diagram kan gjerne være håndtegnet).*
- *Gode vurderinger av resultat.*
- *Systematisk og godt strukturert PYTHON-program. Koden skal være oversiktlig og ha fylldige kommentarer. Det er ikke avgjørende at koden er mest mulig "avansert", beregningen kan med fordel brytes ned i små enheter/sub-rutiner. Dette vil også lette feilsøking*
- *. NB: Programmet vil bli testet for kjørbarehet.*
- *Vurderingsmatrisen blir publisert i et eget dokument på Blackboard sammen med prosjektet.*

Et råd:

Dere vil garantert gjøre feil under programmeringen. Foreta grundig kontroll av hver enkel sub-rutine. Lag et små testprogram som gir enkle inngangsdata over det hele mulige intervallet, hvor resultatene av beregningen kan sjekkes for hånd. Tenk gjennom hvor deling med 0 kan forekomme, og legg gjerne inn kode for å unngå dette eller en feilutgang når dette skjer. I hvilke intervall kan vinkler forekomme? Man bør være 100% sikker på at hver rutine gjør akkurat det den skal, før den implementeres i hovedprogrammet. Utsettelse av feilsøking til hele programmet kjøres gjør dette arbeidet mye vanskeligere, og man kommer sannsynligvis ikke i «mål»



Figur 1: Plattformmodul m/kranlast

Et eksempel på den plane rammen som skal analyseres er vist i Figur 1. Den består av en dekkskonstruksjon som står på en jacket. Utstyr på dekk gir en kraft $P1$ i dekkets hjørner og $P2$ på midten av dekkbjelken. Vannflaten regnes opp til topp av jacket'en. Det kan antas X-diagonaler ikke er forbundet i krysningspunktet.

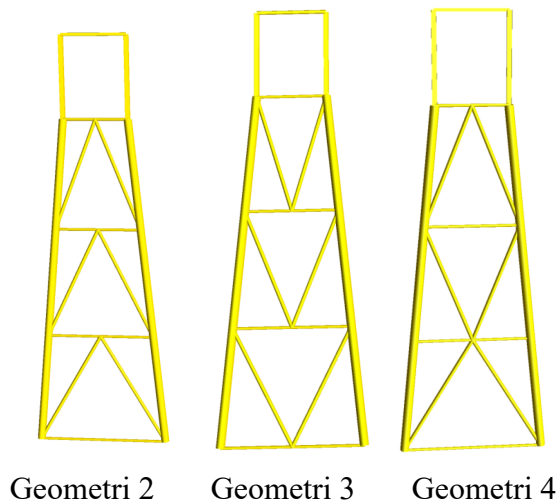
Jacket'en består av rør med ulike diametere. Om ønskelig kan samme diameter benyttes for henholdsvis bein, diagonalstag og horisontalstag som vist på Figur 1. Dekkskonstruksjonene er bygget i to I-profiler for henholdsvis søyle og tverrbjelke. Jacket'en er bygget i stål, mens dekkkonstruksjonen er bygget i Aluminium.

Jacket'en er utsatt for bølgelaster som regnes lineært fordelt over høyden, over to av etasjene på beina, mens det holder å ta i betraktning kun øverste etasje for diagonalstagene. Begge diagonalstagene er belastet (kun en er vist). Intensiteten virker normalt på elementene og gir alle en komponent i horisontalretning til «høyre». De kan regnes som $q1 = 5.4e+05$ [N/m] og

$q_2 = 1.8e+05$ [N/m] for en diameter på 1.5 m. Intensiteten vil være proporsjonal med diameteren.

Flytespenningen er satt lik 300 MPa for både aluminium og stål. Kapasitetskrav for knekking antas å være oppfylt uavhengig av de valgte tverrsnittsdimensjoner.

Hver gruppe skal analysere én av de 4 geometriene som er skissert i Figur 1 (Geometri 1) og Figur 2.

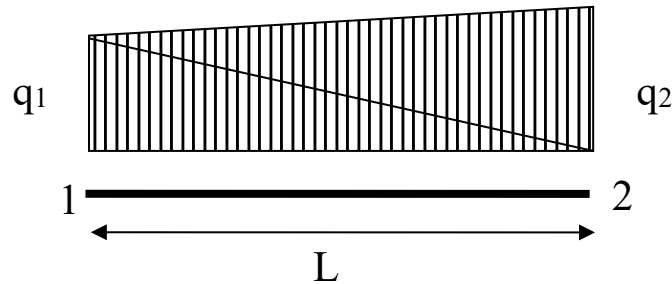


Figur 2: Ulike geometrier

Oppgaven går ut på å lage et program i PYTHON som gjør at man kan analysere konstruksjonen med matrisemetoden.

- a) Etabler en pseudo-kode av det programmet skal utføre.
Lag en funksjon som leser inn konstruksjonsdata som knutepunktskoordinater, elementenes tilknytning til knutepunktene, fordelte og konsentrerte laster til elementene, tverrsnittsdata for rør- og I-profil, grensebetingelser, materialdata osv. Lag også en funksjon som beregner bøyestivheten for elementene på bakgrunn av tverrsnittsdata. Det skal være mulig å benytte ulik tykkelse på flensene og stegene til I-profil. Elementene skal kobles til riktig geometri og material via geometri- og materialnummer i inputfilen. La programmet kunne lese et vilkårlig antall knutepunkt, elementer, laster, tverrsnittsdata og materialer.
- b) Lag de nødvendige funksjoner for å beregne elementstivhetsmatriser, transformasjonsmatrise fra lokalt til globalt system, fastinnspenningsmoment og tilhørende fastinnspenningskraft, lastvektor, oppbygging av systemstivhetsmatrise, beregning av bøyemoment osv. Disse funksjonene implementeres i hovedprogrammet, som kan bygges på koden som ligger på IMT Softwiki. Noen av funksjonskallene er gitt i hovedprogrammet slik at det blir lettere å komme i gang med programmeringen.

Fastinnspenningsmoment og fastinnspenningskraft skal kunne beregnes for last som er lineært fordelt over elementet. Det kan for eksempel splittes i to trekantbidrag.



Grensebetingelsene for knutepunktsforskyvning og - rotasjon skal innføres ved å inkludere deformasjons og rotasjonsfjær til jord i knutepunktet. Dersom en frihetsgrad er fri skal fjæren ha null stivhet, dersom en frihetsgrad er fastholdt skal tilhørende fjær ha stor stivhet, for eksempel 10^6 ganger stivheten til et bjelke-element knyttet til knutepunktet. Disse stivhetene må adderes inn på korrekt plass på diagonalen i systemstivhetsmatrisen.

Programmet skal kunne beregne bøyemoment og skjærkraft på midten av elementer utsatt for fordelte laster.

- c) Kontroller først at programmet regner rett ved å sammenligne med håndberegninger for
 - 1) Fast innspant bjelke med punktlast på midten og en jevnt fordelt last
 - 2) Fast innspant portalramme med horisontal last i et hjørneknutepunkt («side sway».
- d) Analyser konstruksjonen i Figur 3 med Python-programmet. Bestem tverrsnittsdimensjoner slik at bøyespenningsnivået for den mest belastede bjelken i en gruppe av elementer (f.eks. ben, diagonalstag, tverrstag og dekkbjelke) er i størrelsesorden 30-70 % av flytespenningen ved en bjelkeende eller midt på bjelken ved fordelt last. En gruppe av elementer er her definert som i) bein ii) diagonalstag iii) horisontalstag iv) dekkbjelker

Dimensjoneringen blir en iterasjonsprosess hvor man må prøve seg fram med ulike tverrsnittsdata. Her kan man evt. bruke Nauticus 3D Beam for å lette bestemmelsen av dimensjonene, men man skal bruke Python-programmet til den endelige analysen

- e) På grunnlag av valgte tverrsnittsdimensjoner og outputverdier fra PYTHON-programmet skal aksialkraft-, skjærkraft- og momentdiagram tegnes for konstruksjonene.

Merk at det er akseptabelt å kun angi moment – og skjærkraftverdi på midten av bjelker med fordelt last selv om maksimal verdi ikke befinner seg her. Tegning av diagrammer kan godt utføres for hånd, men skulle noen lage en algoritme som kan foreta tegning, teller det positivt.

Data for konstruksjonen:

E-modul for stål = 210 GPa

E-modul for aluminium = 70 GPa

Flytespenning = 300 MPa

		Geometri	L ₁ [m]	L ₂ [m]	L ₃ [m]	P ₁ [MN]	P ₂ [MN]
Gruppe#	1, 5, 9,...	1	20	15	12		
	2, 6, 10...	2	18	14	12		
	3, 7, 11...	3	25	18	18		
	4, 8, 12..	4	22	16	15		
Gruppe#	1, 2, 3....8					3.5	0.5
	9, 10.....16					4.0	1.0
	17,18....24					3.0	1.0
	25, 26..100					5.0	0.6

Tabell 1: Inngangsdata avhengig av gruppenummer

Ved bruk av Nauticus 3D Beam:

Dersom man kontrollerer resultater fra PYTHON-programmet med Nauticus 3D Beam vil man kunne observere små avvik: Det skyldes bl.a. at 3D Beam er basert på en bjelkeformulering som inkluderer bidrag fra skjærdeformasjoner.

En input-fil til programmet USFOS kan se ut som nedenfor. Dette gir en indikasjon på strukturen til input-filen, men dere må lage den på den måten som er praktisk. Alle input-data skal leses inn, ingen skal hard-kodes.

ModelName
---- Rammekonstruksjon ----

Node ID	X	Y	Z	Boundary code
NODE 1	-12.000	0.000	0.000	1 1 1 1 0 1
NODE 2	+12.000	0.000	0.000	1 1 1 1 0 1
NODE 3	-10.000	0.000	20.000	0 1 0 0 0 1
NODE 4	+10.000	0.000	20.000	0 1 0 0 0 1
NODE 5	-8.000	0.000	40.000	0 1 0 0 0 1
NODE 6	+8.000	0.000	40.000	0 1 0 0 0 1
NODE 7	-6.000	0.000	60.000	0 1 0 0 0 1
NODE 8	+6.000	0.000	60.000	0 1 0 0 0 1
NODE 9	-6.000	0.000	75.000	0 1 0 0 0 1
NODE 10	+6.000	0.000	75.000	0 1 0 0 0 1
NODE 11	0.000	0.000	75.000	0 1 0 0 0 1

ElemID	Nod1	Nod2	Mat	Geom	L.Corr.
BEAM 13	1	3	1	1	
BEAM 35	3	5	1	1	
BEAM 57	5	7	1	1	
BEAM 24	2	4	1	1	
BEAM 46	4	6	1	1	
BEAM 68	6	8	1	1	
dekkonstruksjon					
BEAM 79	7	9	2	4	1
BEAM 810	8	10	2	4	1
BEAM 911	9	11	2	5	
BEAM 1011	10	11	2	5	
Tverr diagonaler					
BEAM 12	1	2	1	2	
BEAM 34	3	4	1	2	
BEAM 56	5	6	1	2	
BEAM 78	7	8	1	2	
Skrå diagonaler					
BEAM 14	1	4	1	3	
BEAM 32	3	2	1	3	
BEAM 36	3	6	1	3	
BEAM 54	5	4	1	3	
BEAM 58	5	8	1	3	
BEAM 76	7	6	1	3	

GeomID	Do	Thick
PIPE 1		
PIPE 2		
PIPE 3		

GeomID	H	T-web	W-top	T-top	W-bot	T-bot
IHPROFIL 4						
IHPROFIL 5						

L.Case	Node	Fx	Fy	Fz	Mx	My	Mz
NODELOAD 1	9	0.000e+00	0.000e+00	-1.000e+07/10	0.000e+00	0.000e+00	0.000e+00
NODELOAD 1	10	0.000e+00	0.000e+00	-1.000e+07/10	0.000e+00	0.000e+00	0.000e+00
NODELOAD 1	11	0.000e+00	0.000e+00	-0.500e+07/40	0.000e+00	0.000e+00	0.000e+00

Load Case	Elem ID	L O A D I N T E N S I T Y					
		qx1	qx1	qx1	qx2	qx2	qx2
BEAMLOAD 1	57	0.5e+05	0	0	1.5e+05	0	0
BEAMLOAD 1	68	0.5e+05	0	0	1.5e+05	0	0
BEAMLOAD 1	35	0.0e+05	0	0	0.5e+05	0	0
BEAMLOAD 1	46	0.0e+05	0	0	0.5e+05	0	0

Mat no	E-mod	Poisson ratio	Yield	Density	Therm. expansion
MISOIEP 1	2.1e+11	0.3	3.55e+08		
MISOIEP 2	2.1e+11/3	0.3	3.55e+08		