

Bruprosjektering med bruk av parametrisk design

«En arbeidsflyt mellom Dynamo-Revit-
Sofistik»

Daniel Nilsen
Øystein Urdal

Master i Bygg- og miljøteknikk

Innlevert: juni 2018

Hovedveileder: Anders Rönquist, KT

Medveileder: Jan Arve Øverli, KT

Norges teknisk-naturvitenskapelige universitet
Institutt for konstruksjonsteknikk



MASTEROPPGAVE 2018

FAGOMRÅDE: Ingeniørarkitektur

DATO: 11.07.2018

ANTALL SIDER: 86

TITTEL:

Bruprosjektering med bruk av parametrisk design

«En arbeidsflyt mellom Dynamo-Revit-Sofistik»

Bridge design with the use of parametric design

«A workflow between Dynamo-Revit-Sofistik»

UTFØRT AV:

Daniel Nilsen
Øystein Urdal



SAMMENDRAG:

Denne masteroppgaven tar for seg bruken av parametrisk design til å effektivisere arbeidsflyten mellom modellproduksjon og beregning av enkle brukonstruksjoner. For å oppnå dette blir en arbeidsflyt mellom tegningsprogrammet Revit og elementprogrammet Sofistik etablert, med utgangspunkt i det visuelle programmeringsprogrammet Dynamo.

Gjennom å utforske mulighetene for en arbeidsflyt mellom Dynamo, Revit og Sofistik er fire alternative arbeidsflyter utformet. Alle fire arbeidsflyter er potensielle løsninger for samhandling mellom programmene, der modellen formes gjennom parametrisk design. Det alternativet man anser er best egnet for en god arbeidsflyt blir tatt i bruk i opprettelse av en parametrisk brumodell.

Den parametriske modellen blir sammenlignet med en tilsvarende modell, designet på tradisjonell måte gjennom Sofistik. Beregningsresultatene fra den parametriske modellen er gode og uten nevneverdige forskjeller fra den tradisjonelle. Ved å utføre parametriske studier hvor ulike utforminger testes, kommer det frem at modellen er fleksibel og funksjonell. Studiene gir en god bekreftelse på at modellen fungerer. Overgangen mellom Revit og Sofistik krever at bruken lages med analytiske element. Dette er den største utfordringen i målsetningen om god tegningsmodellen ettersom dette gir begrensninger i den geometrisk utforming. Følgelig er en helt nøyaktig representasjon av bruken vanskelig. Programvaren samarbeider dermed ikke godt nok til å realisere målsetningen om en parametrisk modell for effektivisering av bruprosjektering, hvor både tegnings- og beregningsmodellen blir komplett. Resultatmessig oppnår arbeidsflyten en god beregningsmodell. Bruken av parametrisk design i kombinasjon med Revit og Sofistik er derfor mer egnet for den konseptuelle fasen, hvor mange ulike utforminger effektivt kan beregnes.

FAGLÆRER: Nils Erik Anders Rønnquist

VEILEDER(E): Jan Arve Øverli og Marcin Luczkowski

UTFØRT VED: Institutt for konstruksjonsteknikk

Sammendrag

Denne masteroppgaven tar for seg bruken av parametrisk design til å effektivisere arbeidsflyten mellom modellproduksjon og beregning av enkle brukonstruksjoner. For å oppnå dette blir en arbeidsflyt mellom tegningsprogrammet Revit og elementprogrammet Sofistik etablert, med utgangspunkt i det visuelle programmeringsprogrammet Dynamo.

Gjennom å utforske mulighetene for en arbeidsflyt mellom Dynamo, Revit og Sofistik er fire alternative arbeidsflyter utformet. Alle fire arbeidsflyter er potensielle løsninger for samhandling mellom programmene, der modellen formes gjennom parametrisk design. Det alternativet man anser er best egnet for en god arbeidsflyt blir tatt i bruk i opprettelse av en parametrisk brumodell.

Den parametriske modellen blir sammenlignet med en tilsvarende modell, designet på tradisjonell måte gjennom Sofistik. Beregningsresultatene fra den parametriske modellen er gode og uten nevneverdige forskjeller fra den tradisjonelle. Ved å utføre parametriske studier hvor ulike utforminger testes, kommer det frem at modellen er fleksibel og funksjonell. Studiene gir en god bekreftelse på at modellen fungerer. Overgangen mellom Revit og Sofistik krever at bruken lages med analytiske element. Dette er den største utfordringen i målsetningen om god tegningsmodellen ettersom dette gir begrensninger i den geometriske utforming. Følgelig er en helt nøyaktig representasjon av bruken vanskelig. Programvaren samarbeider dermed ikke godt nok til å realisere målsetningen om en parametrisk modell for effektivisering av bruprojektering, hvor både tegnings- og beregningsmodellen blir komplett. Resultatmessig oppnår arbeidsflyten en god beregningsmodell. Bruken av parametrisk design i kombinasjon med Revit og Sofistik er derfor mer egnet for den konseptuelle fasen, hvor mange ulike utforminger effektivt kan beregnes.

Summary

This master thesis explores the possibilities to use parametric design to improve the efficiency of the workflow between model production and calculation of simple bridge structures. To achieve this a workflow between the drawing program Revit and the finite element program Sofistik is established, with the visual programming program Dynamo as a base.

By exploring the possibilities of a workflow between Dynamo, Revit and Sofistik, four alternative workflows are created. All four workflows introduce a potential solution for interaction between the programs, whereas the model is created through parametric design. The alternative that is considered the best will be used to create a parametric bridge model.

The parametric model is compared to an equivalent model which is designed with a traditional method by Sofistik. The results from the calculation by the parametric model are good and not significantly different from the traditional model. By carrying out parametric studies where different designs are tested, it is revealed that the model is flexible and functional. The studies provide a good confirmation that the model works. The transition between Revit and Sofistik requires that the bridge is modeled with analytical elements. This is the greatest challenge in the aim for a good drawing model, seeing that these elements give limitations to the geometrical design. Consequently, an exact representation of the bridge is difficult. Therefore, the software does not collaborate sufficiently enough to accomplish the goal of a parametric model to make bridge modeling more efficient, where both the drawing- and calculation model is complete. In terms of results, the workflow achieves a good calculation model. Parametric design combined with Revit and Sofistik is thus more suited to a conceptual phase, where many different designs can be effectively calculated.

Forord

Denne masteroppgaven er skrevet vårsemesteret 2018 som en del av vår mastergrad ved Norges teknisk-naturvitenskaplige universitet. Oppgaven er skrevet i samarbeid med Institutt for konstruksjonsteknikk.

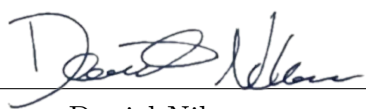
Oppgaven er utarbeidet i samarbeid med COWI med hensikt om å utforske muligheten til å effektivisere flyten mellom modellproduksjon/tegningsproduksjon og beregning av brukonstruksjoner.

Byggebransjen er i en stor utvikling innenfor digitalisering og gjennom de siste fem årene i vår studietid har vi fått være med på å se mye av denne utviklingen. Dette har gitt oss stor interesse for å ta del i utviklingen og skaffe oss verdifull og fersk kompetanse. Parametrisk design er i spissen for digitaliseringen og vi ser på det som et svært nyttig og interessant verktøy til å effektivisere prosjekter.

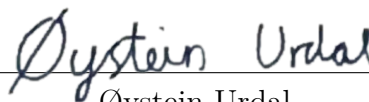
Bruer er en konstruksjon vi interesserer oss for og derfor var det et naturlig ønske om å ta i bruk parametrisk design for bruprojektering. Ettersom begge skal arbeide med bruprojektering etter studiene var dette en gylden mulighet til å få betydningsfull kunnskap innenfor parametrisk design, bruprojektering og relevante programvarer.

Vi ønsker å takke veiledere Anders Rönquist og Marcin Luczkowski som har inspirert til å skrive en masteroppgave innen parametrisk design. Vi har satt stor pris på deres veiledning og støtte gjennom hele oppgaven. Marcin har gitt verdifull innføring og veiledning i programvarene benyttet i oppgaven. En takk rettes også til Jan Arve Øverli for gode råd for betongkonstruksjoner. Videre vil vi også takke ansatte i COWI og Sweco for gode tips og råd gjennom oppgaven.

Trondheim, 10. juni 2018



Daniel Nilsen



Øystein Urdal

Introduksjon

Bakgrunn for oppgaven

Digitaliseringen i ingeniørbransjen har gitt opphav til parametrisk design, hvor det foreligger en ny måte å lage modeller på. I stedet for å tegne, lager man instruksjoner som forteller hva datamaskinen skal lage. Den videre overgangen fra tegnings- til beregningsmodell er i denne sammenhengen ikke spesielt utbredt.

Utgangspunktet for denne oppgaven er derfor å ta i bruk parametrisk design til å effektivisere arbeidsflyten mellom modellproduksjon og beregning av enkle brukonstruksjoner. Dette innebærer fokusering på overgangen mellom tegningsprogrammet Revit og beregningsprogrammet Sofistik i samhandling med Dynamo.

Målsetning

- Undersøke arbeidsflyten mellom Revit og Sofistik for bruprojektering.
- Benytte parametrisk verktøy for å effektiviser arbeidsflyten mellom modellproduksjon og beregning av enkle brukonstruksjoner.

Oppgavens oppbygning

- Kapittel 1 introduserer oppgaven og beskriver anvendte programvarer.
- Kapittel 2 inneholder beskrivelse av inspirasjonsbruen, spennarmering og laster.
- Kapittel 3 beskriver viktige momenter for hvordan tegnings- og beregningsmodellen modelleres. Hvilke element bruen skal bygges opp av og hvorfor.
- Kapittel 4 avslutter første «del» av oppgaven med tradisjonell modellering av en bru i Sofistik.
- Kapittel 5 presenterer fire alternative måter til å bruke parametrisk design i arbeidsflyt for bruprojektering.
- Kapittel 6 gir en detaljert beskrivelse av den arbeidsflyten med størst potensial, inkludert alle de viktige stegene i prosessen.
- Kapittel 7 inneholder studier for å vise fleksibiliteten til den parametriske modellen, samt viser hvordan den fungerer i praksis.
- Kapittel 8 diskuterer arbeidsflyten og parametrisk design, etterfulgt av en konklusjon.
- Kapittel 9 tar for seg forslag til videre arbeid for hvordan arbeidsflyten kan utvikles.

Innhold

Figurer	x
Tabeller	xiii
1 Innledning	1
1.1 Parametrisk design	2
1.2 BIM og FEM	3
1.3 Programmer	4
1.3.1 Revit	4
1.3.2 Dynamo	6
1.3.3 Sofistik	8
2 Brudesign	10
2.1 Spennarmering	10
2.2 Laster	15
2.2.1 Permanente laster	15
2.2.2 Variable laster	16
3 Element i analyse	20
3.1 Bjelkeelement	20
3.2 Skallelement	21
3.3 Valg av element	22
4 Tradisjonell modellering	24
4.1 Modelleringsgang i Sofistik	24
4.1.1 Designfasen	25
4.1.2 Resultatfasen	27
5 Arbeidsflyt i programvare	31
5.1 Flytalternativ I - Parametrisk modell med bruk av Sofiplus for tilleggsinformasjon	32
5.2 Flytalternativ II - Parametrisk modell gjennom Teddyskript	34
5.3 Flytalternativ III - Dynamo som database for separate modeller	36
5.4 Flytalternativ IV - Parametrisk modell med bruk av Teddy for tilleggsinformasjon	37
6 Parametrisk modell	38
6.1 Parametrisk modellering	38
6.1.1 Dynamo	38
6.1.2 Revit	54
6.1.3 Sofistik	57
6.2 Sammenligning av parametrisk- og tradisjonell modell	58
6.3 Spennarmering i den parametriske modellen	62
7 Parametrisk studie	66
7.1 Studie 1 – Spennendringer	66
7.2 Studie 2 – Endring av brutverrsnitt	67

7.3	Studie 3 – Endring av søyleverrsnitt	69
7.4	Studie 4 – Stivhetsendringer	72
7.5	Studie 5 – Endring i høyde	73
7.6	Studie 6 – Endring av veilinje	75
7.7	Studie 7 – Endring av søylevinkel	76
7.8	Studie 8 – Kompleks geometri	78
8	Diskusjon	80
8.1	Arbeidsflyt	80
8.2	Parametrisk design	84
8.3	Konklusjon	85
9	Videre arbeid	86
A	Inspirasjonsbru	
B	Ekvivalente krefter	
C	Egenlast laster	
D	Trafikklast	
E	Vindlast	
F	Temperaturlast	
G	Armeringskontroll	
H	Armeringsberegninger	
I	Skript fra Visual studio	
J	Pythonskript for linjelaster	
K	Tilleggsfiler	

Figurer

1.1	Ønsket arbeidsflyt	1
1.2	Dagens arbeidsflyt	1
1.3	Hendon bru [3]	3
1.4	3D-modell av en bru i Revit [8]	4
1.5	Elementhierarki i Revit [9]	5
1.6	Analytiske element i Revit [10]	5
1.7	Aktiv- og inaktiv node	6
1.8	Visuell programmering	7
1.9	Tekstbasert programmering	7

1.10	Fargekoder i Dynamo	8
2.1	Påvirkning av sentrisk spennkabel i bjelke [18]	10
2.2	Påvirkning av parabelformet spennkabel i bjelke [18]	11
2.3	Beregning av ekvivalente krefter utgangspunkt [18]	12
2.4	Ekvivalente krefter for parabelformet fritt opplagt bjelke [18]	12
2.5	Forankringskrefter [18]	13
2.6	Tvangskrefter	14
2.7	Fordeling av temperatur i et tverrsnitt [25]	17
2.8	Kombinasjonstabell for jevnt fordelt- og vertikalt lineært varierende temperatu- randel [20]	19
3.1	Frihetsgrader og krefter bjelkeelement [6]	20
3.2	Stivhetsmatrise 3D-bjelkeelement [6]	20
3.3	Bestemmelser av variabler gitt i bjelkestivhetsmatrisen [6]	21
3.4	Krefter i skallelement	21
3.5	Triangulær skallmatrise [6]	21
3.6	Bestemmelse av l_0 EK2 pkt [5.3.2.1]	23
3.7	Effektiv flensbredde EK2 pkt [5.3.2.1]	23
4.1	Tradisjonelt flytdiagram	24
4.2	Typisk tverrsnitt gitt av COWI, benyttet som inspirasjon under modellering	25
4.3	Tverrsnitt benyttet i analytisk modell, forenklet versjon	25
4.4	Statisk system for brumodell	26
4.5	Sofistikmodell etablert med Sofiplus	27
4.6	Maksimal og minimum momentdiagram beregnet fra Sofistik	29
4.7	Beregnet Armering fra Sofistik uten spennarmering	30
4.8	Maks/min moment om y-aksen etter utplassering av spennarmering	30
5.1	Ideell arbeidsflyt for effektivisering av enkel bruprosjektering	31
5.2	Flytdiagram I. Tilleggsinformasjon gjennom Sofiplus	32
5.3	Overføring av geometri til SAT-fil i Dynamo	33
5.4	Flytdiagram II. Modellering gjennom bruk av Teddy	34
5.5	Overførbare elementer Sofistik-Revit	35
5.6	Flytdiagram III. Dynamo som database for to separate modeller	36
5.7	Flytdiagram IV. Parametrisk modellering i Dynamo, hvor Teddy benyttes for tilleggsinformasjon	37
6.1	Parametere brutverrsnitt	38
6.2	Parametere i Revit	39
6.3	Structural framing-node med tilhørende input	39
6.4	Bru i Revit laget med adaptive family	40
6.5	Brulinje i Dynamo	41
6.6	Oppdeling av brulinje	41
6.7	Overføring til Revit	42
6.8	Bruparametere i Dynamo	43
6.9	Fastholding i Dynamo	44
6.10	Materiale i Dynamo	45
6.11	Akse til bruk i Sofistik	46
6.12	Søyler i Dynamo	47

6.13	Søyler i Dynamo 2	48
6.14	Rotasjon av søyler i Dynamo	48
6.15	Hosted line load	49
6.16	Hosted last i Dynamo	50
6.17	Eksentrisk last i Dynamo	51
6.18	Sletting av eksisterende laster før nye lages	52
6.19	Lasttilfelle angis last	52
6.20	Egenskaper/parametere for laster i Revit	53
6.21	Temperaturlast for Sofistik i Dynamo	53
6.22	Systemdialog i Revit	54
6.23	Lasttilfeller i Revit	55
6.24	Materiale i Revit	55
6.25	Opplagerbetingelser i Revit	56
6.26	Analytisk link i Revit	56
6.27	Generering av Sofistikk-fil i Revit	57
6.28	Informasjon og oppgaver i Sofistik, samt akse	57
6.29	Maks/min momentdiagram parametrisk bru	59
6.30	Uthenting av punkter fra XML-fil	60
6.31	Punkter (blått) med innplassert bruprofil	60
6.32	Unøyaktighet ved linjebasert element	60
6.33	Maks- og min moment for bruprofil som følger veilinje	61
6.34	Spennarmering- skript 1	62
6.35	Spennarmering- skript 2	62
6.36	Spennarmeringskript 3	63
6.37	Viser utplassert spennarmering ved bruk av nodealternativ 3 på bru som følger veilinje	64
6.38	Viser maks- og minimumsmoment ved utplassering av mulig spennarmering illustrert i figur 6.37	64
7.1	Maks/min moment og nedbøyning for spennendring	67
7.2	Maks/min moment og nedbøyning for tverrsnittsendringer	68
7.3	Midtre søyler 2000mm x 800mm	69
7.4	Søyler med diameter 900 mm	69
7.5	Dynamoskript modifisert for to sirkulære søyler	69
7.6	Momentinnvirkning for brudekket ved ulike søyleprofil, volumforskjell illustrert i grønt	70
7.7	Tykkspenning og moment med ulike søyleprofil	71
7.8	Effekt av endring i søyletykkelsen	71
7.9	Maks moment i brudekket ved endring av lager, X = 0 m t.v, X = 70,8m t.h	72
7.10	Min moment i brudekket ved endring av lager, X = 0 m t.v, X = 70,8m t.h	72
7.11	Kurvet bru i Sofistik, 3m høyde	73
7.12	Maks/min moment for høydeendring, absoluttverdi	74
7.13	Kurvet bru i Revit	75
7.14	Kurvet bru i Sofistik	75
7.15	Kurvet bru i Dynamo	75
7.16	Endring av søylevinkel	76

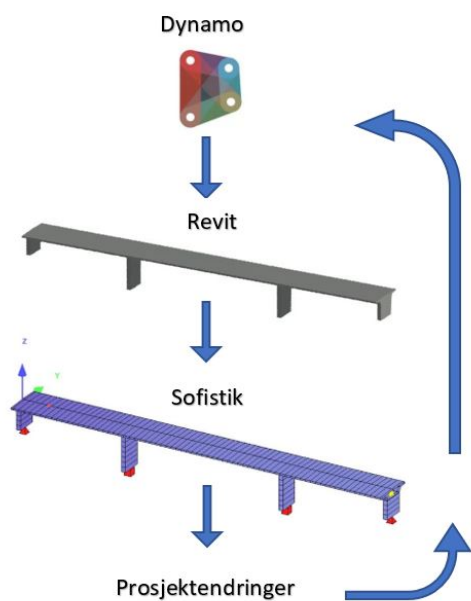
7.17	Vinkelendring i Dynamo	76
7.18	Hvordan vinkelendring i midtre søyler påvirker momentet, gitt som absoluttverdier	77
7.19	Maks opplagerbetingelser ved vinkelendring av søyle	77
7.20	Kompleks bruutforming med laster	78
7.21	Svingende gangbru i Revit	78
7.22	Svingene gangbru i Sofistik	78
7.23	Kompleks gangbru i Revit	79
7.24	Kompleks gangbru i Sofistik	79
7.25	Kompleks gangbru i Revit 2	79
7.26	Kompleks gangbru i Revit 2	79
7.27	Eksempel på momentdiagram om y-aksen for sirkulær gangbru, kun egenvekt. .	79

Tabeller

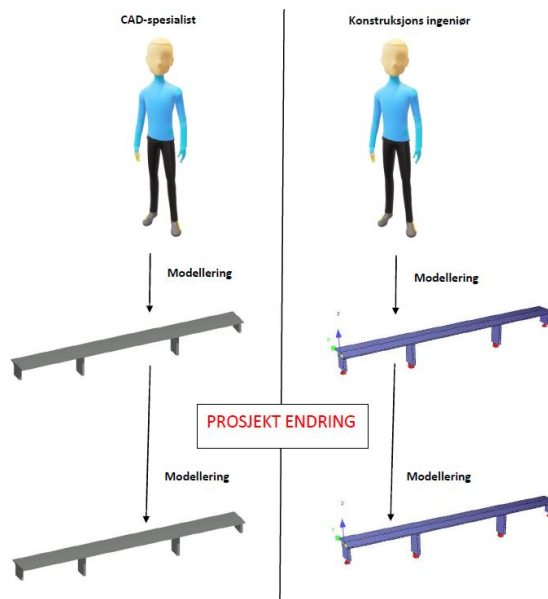
2.1	Verdier basert på ekvivalente krefter og bjelkemodell i Sofistik, parabelformet spennkabelføring	13
2.2	Verdier basert på ekvivalente krefter og bjelkemodell i Sofistik. rett spennkabelføring	13
2.3	Dimensjonerende belegningsvekt i kjørebane [22]	15
2.4	Lastmodell 1	16
2.5	Oppsummert karakteristiske laster	19
4.1	Karakteristiske laster	26
4.2	Effekten av mesh	27
4.3	Avleste verdier	28
4.4	Slakkarmering uten spennarmering	29
6.1	Sammenligning av My-verdier	61
6.2	Spennarmering i modellen	65
7.1	Resultater spennendringer	66
7.2	Resultater for tverrsnittsendringer	68
7.3	Endring av søyletverrsnitt	70
7.4	Endring i søyle/dekke -forbindelsen	73
7.5	Resultat for høydeendring av bru	74

1 Innledning

Infrastruktur er en stor satsing for regjeringen og en av grunnpilarene i moderne samfunn. «Det er viktig med god infrastruktur for å skape gode bo- og arbeidsregioner, og for å knytte produksjon av varer og tjenester til markedene», skriver regjeringen på sin politiske plattform i 2018 [1]. Da Norge er et langstrakt land med varierende topologi vil en utbedring eller utbygning av veinettet ofte kreve atskillige brukonstruksjoner. Ikke bare av hensyn til topologien, men blant annet kryssing og påkjøring av store veiårer, deriblant E18 og E39, krever en mengde bruer. Utbygningen gir hovedvekt av brukonstruksjoner i den mindre skalaen. Bruene kan ha flere likhetstrekk, men likevel kreves det at mye arbeid må gjøres på nytt. Endringer i tilnærmet like modeller kan være like tidkrevende som å lage en ny. Tilsvarende, kan fortløpende endringer i prosjektet skape utfordringer. Etter beregningen er det omtrent blitt et krav om 3D-modellering av bruene for å bistå entreprenørene i den utførende fasen. Prosessen med å etablere beregnings- og tegningsmodeller er i dag to uavhengige operasjoner, gjerne utført av to ulike aktører. Fremgangsmåten kan dermed enklere medføre feil i beregnings- eller tegningsmodellen og etterlater seg stort forbedringspotensial.



Figur 1.1: Ønsket arbeidsflyt



Figur 1.2: Dagens arbeidsflyt

Fremskritt i digitaliseringen åpner opp for å ta i bruk «nye» verktøy, og med det skape et konkurransefortrinn innenfor prosjektering. I spissen av denne digitaliseringen ligger parametrisk design. Parametrisering har ofte en effektiviserende effekt ved at endringer krever minimalt med tid og arbeid. Slike verktøy har regelmessig mindre begrensninger og kan bindes til ulike programmer. Parametriske verktøy, deriblant Dynamo, anvendes ofte til manipulasjon av eksisterende modeller. Likevel har verktøyet et større potensial til å være startpunkt for all input. Med støtte fra parametrisk design muliggjøres en arbeidsflyt skreddersydd for oppføring og endring av tegnings- og analysemodell under en og samme prosess. Vendes blikket mot da-

gens rådgivningsfirma er det blitt mer vanlig å ta i bruk parametrisk design. En fremtid hvor tilnærmet all modellering foregår ved bruk av et slikt hjelpemiddel er dermed ikke utenkelig.

Målsetningen er å benytte parametrisk design til å effektivisere arbeidsflyten mellom modellproduksjon og beregning av enkle brukonstruksjoner. Her ser oppgaven på bruken av det parametriske verktøyet Dynamo i samarbeid med beregningsprogrammet Sofistik, for å effektivisere brumodellering. I tillegg til anvendelse av parametrisk design undersøker oppgaven hvordan Sofistik samarbeide med tegningsprogrammet Revit. Introduisering av en arbeidsflyt mellom en tegning- og analytiskmodell for brumodellering er ikke uten mangler. Utilstrekkelig funksjonalitet vil dermed være like viktig for oppgaven.

Innledningsvis redegjøres det for ulike programvarer anvendt i oppgaven. Deretter beskrives dimensjoneringshensyn relevant for analyse før første del avsluttes med tradisjonell brudimensjonering i Sofistik. Denne dimensjoneringen er begrenset etter oppgavens behov og er med på å undersøke kvaliteten til den parametriske modellen. Andre del tar i større grad for seg flyten mellom analyse- og tegningsmodell. Her blir ulike metoder for arbeidsflyt vurdert basert på «prøve- og feile metoden». De ulike alternativene beskrives gjennom egne erfaringer for utarbeidelse av en parametrisk modell der arbeidsflyten fremmes. Arbeidsflyten er primært bygd rundt en fungerende analytisk modell. Konseptet med størst flyt- og parametriseringspotensial videreføres for mer detaljert beskrivelse. Gjennom ulike studier testes deretter arbeidsflyten for å underbygge effekten av parametrisk design, samtidig med at kvaliteten til modellen tydeliggjøres.

1.1 Parametrisk design

Tradisjonelt representerer dagens programvarer og hjelpemidler design på en statisk måte. Programvaren gir et digitalt 3D miljø og modelleringsverktøy hvor det lages singulære designløsninger statisk representert av fysisk geometri. Designet er derfor ofte låst og begrenset til en bestemt geometri. Et av målene med oppgaven er å utvikle et system der det kan lages mange ulike design basert på et sett med input parametere. Modellen vil dermed være en dynamisk modell som kan endres mer effektivt og raskere enn den statiske modellen. For å gjøre dette trengs andre typer designverktøy. I stedet for for å fokusere på fysisk geometri og bestemme en bestemt statisk form, bestemmes prosessen for hvordan designet lages. Gjennom å benytte koder for å lage et sett med instruksjoner kan man bestemme hvordan designet skal genereres. I dataterminologi beskrives et sett med instruksjoner en algoritme og man kan derfor kalle design gjennom instruksjoner for *algoritmisk design*. I praksis blir disse design prosessene ofte referert til som *parametrisk design* eller *computational design* [2].

Gjennom parametrisk design lages instruksjoner. Disse instruksene forteller datamaskinen og programvaren i en serie med trinn hva den skal lage. Et verktøy for parametrisk design er Dynamo. I stedet for å skrive kode fra bunn, vil man i Dynamo skrive koder i et annet programmeringsmiljø basert på visuell programmering. Dynamo fremstår derfor mer brukervennlig og attraktivt for ingeniører uten spesiell kompetanse innenfor ren programmering.



Figur 1.3: Hendon bru [3]

Et eksempel på en bru hvor parametrisk design er tatt i bruk er Te Whittinga, også kalt Hendon Bridge i New Zealand [4]. Bruen er en gangbru og ble oppført i 2016. Parametrisk design ble brukt for å tillate iterative testing og optimering av brudesignet. Gjennom parametriske instruksjoner er det opprettet geometriske forhold til design av alle aspekter av bruene. Alt fra brudekket til buen til rekkverket, illustrert i figur 1.3, kan styres parametrisk [3].

1.2 BIM og FEM

BIM – Bygningsinformasjonsmodellering

BIM er en prosess hvor all bygningsinformasjon legges inn i én og samme digital modell. Gjennom BIM styres bygningsinformasjonen gjennom alle faser i et byggeprosjekt, fra start til slutt [5]. BIM gjør det mulig å effektivt planlegge, prosjektere, bygge og styre byggeprosjekt for alle de involverte fagfeltene. Dette kan redusere feil og misforståelser da alle fagfeltene jobber i samme tverrfaglige modell. Fra den digitale modellen kan det blant annet hentes ut 2D-plantegninger, armeringstegninger i 3D og mengdelister.

I dag er forholdet mellom tegning- og analysemodell en av utfordringene med BIM. På bakgrunn av dette er det mest vanlig å lage en av hver. Ved to uavhengige modeller vil endringer gjøres i to operasjoner, illustrert i figur 1.1 og 1.2. Dette kan føre til mindre rom for undersøkelser av mer effektiv lasthåndtering der det kan forekomme økonomisk gevinst.

FEM – Finite element method

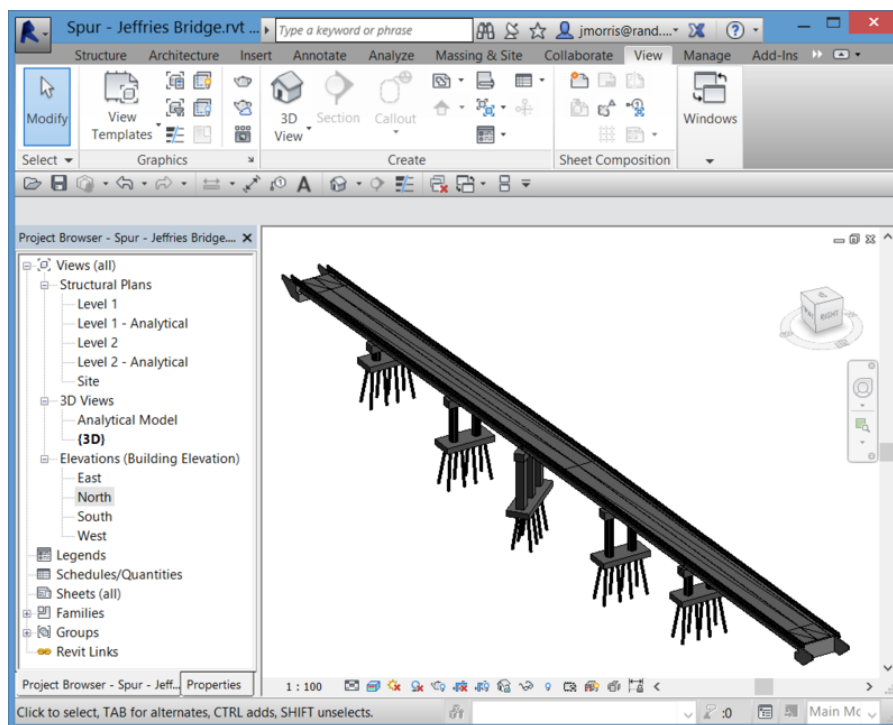
FEM er en numerisk metode for beregning av tilnærmet nøyaktige krefter. Gjennom oppgaven blir programvare basert på FEM tatt i bruk for å skape en beregningsmodell. Elementprogrammer er mye brukt for ingeniører i beregning av krefter for aktuelle konstruksjoner [6]. Gjennom oppgaven vil beregningsmodellen ofte være omtalt som den analytiske modellen.

1.3 Programmer

Flere programmer er benyttet i oppgaven. Disse presenteres dypere i dette delkapittelet. Hensikten med beskrivelsen er å gi grunnleggende forståelse av programvarens funksjon, da disse har en essensiell rolle i oppgaven. Hovedsakelig er kun vesentlige komponenter tilknyttet oppgaven forklart. Programmene har derfor ytterligere funksjoner og bruksområder utenom det som er beskrevet. Forkunnskap innenfor aktuelle programmer er en fordel under oppgavelesning, men ikke en nødvendighet.

1.3.1 Revit

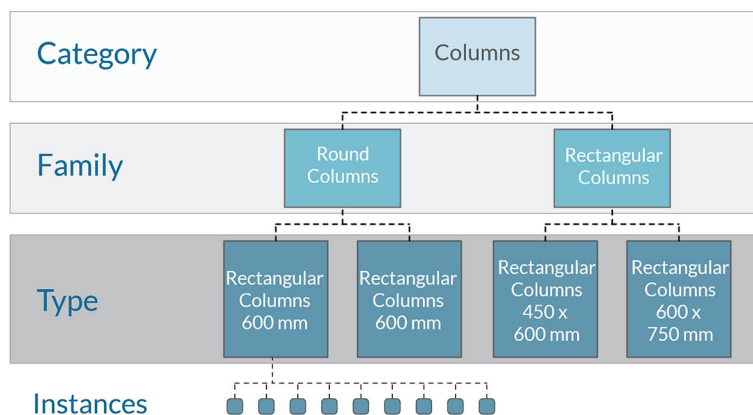
Revit er et program for bygningsinformasjonsmodellering (BIM) [7]. Revit benyttes til å tegne 3D-modeller og 2D-tegninger. I Revit er det mulig å designe og tegne bygninger med informasjon fra alle disipliner i prosjektet. Alle disipliner i et prosjekt har anledning til å bruke én og samme modell som deles mellom alle. Under oppgaven er det anvendt Revit versjon 2018.



Figur 1.4: 3D-modell av en bru i Revit [8]

Elementoppbygging i Revit

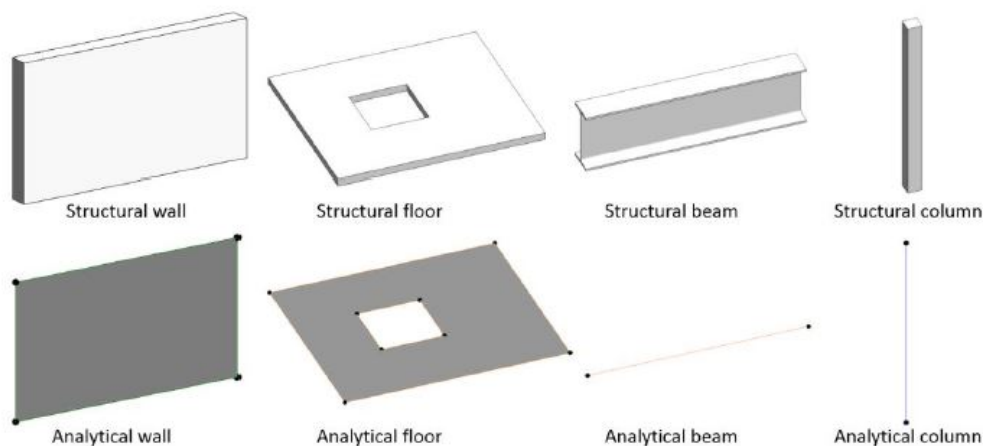
Revitelement er kategorisert i et elementhierarki som kan brytes ned til *kategorier*, *familier*, *typer* og *tilfelle* (*instance*). Forklaringen på dette hierarkiet er best illustrert med et søyleeksempel, se figur 1.5. Denne inndelingen er vesentlig for hvordan Dynamo samarbeider med Revit.



Figur 1.5: Elementhierarki i Revit [9]

Analytisk modell

Ved design i Revit genereres en analytisk modell. Modellen visualiseres med linjer eller skall dersom det ved tegningen er benyttet henholdsvis bjelke eller dekke, se figur 1.6. I den analytiske modellen kan opplagerbetingelser, laster og forbindelser legges til. Forbindelser, omtalt analytiske linker, er uendelig stive deler hvor fastholdningsgrad i start- og endepunkt bestemmes. Grunnlaget i denne analytiske modellen kan overføres til beregningsprogram.



Figur 1.6: Analytiske element i Revit [10]

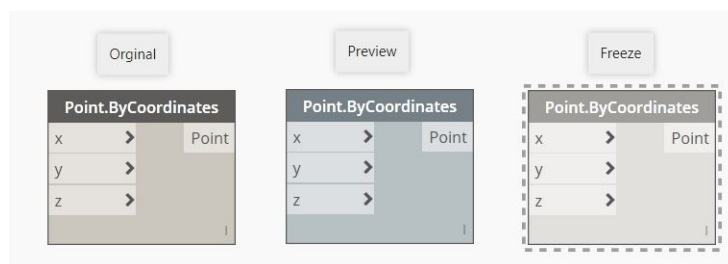
1.3.2 Dynamo

Dynamo er et visuelt programmeringsverktøy som gir brukeren mulighet til å visualisere det man programmerer [11]. Gjennom visuell programmering brukes eksisterende *noder* eller egen-definerte *Code Blocks*. Disse linkes sammen til å bestemme hvordan data bearbeides. I denne sammenhengen er data ikke bare nummer og tekst, men også geometri, parametere og element. Med Dynamo kan modeller enkelt parametriseres, samt skape mer kompleks geometri enn man ellers har mulighet til i Revit. Fordelen med å bruke Dynamo og Revit sammen er at programmene har en toveis datautveksling. Dette betyr at Dynamo kan både eksportere og lese data fra Revit. Denne interaksjonen med BIM/FEM-program er avgjørende for at modellering gjennom parametrisk design er mulig.

Forkunnskaper om programmering er heller ikke nødvendig når man jobber med Dynamo. Dynamo fungerer både som et eget stand-alone program og som et tilleggsprogram til Revit [11]. I oppgaven er det tatt i bruk Dynamo 2017 versjon 1.3.1.

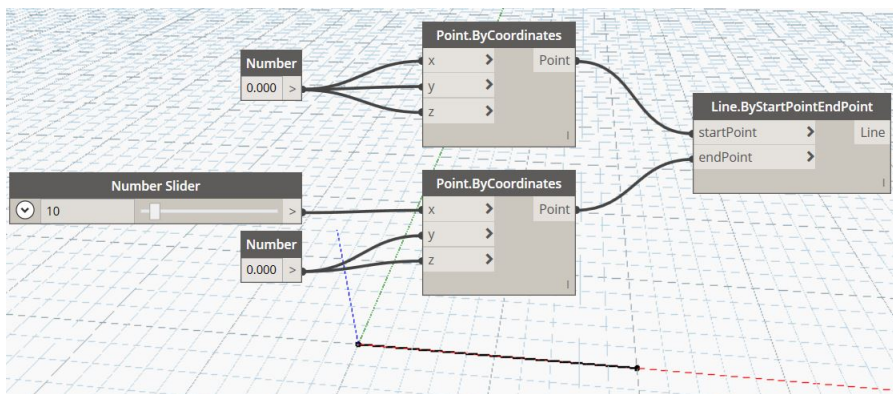
Oppbygging av Dynamo:

- *Noder* er de objektene Dynamo bruker for å lage et visuelt program. Hver node utfører en operasjon som kan være alt fra å lage en linje i Dynamo til å lage kompleks geometri i Revit. Nodene kan velges å være aktive (original), ikke visuelt synlig (preview deaktivert), eller ikke-aktive (freeze), Illustrert av figur 1.7.

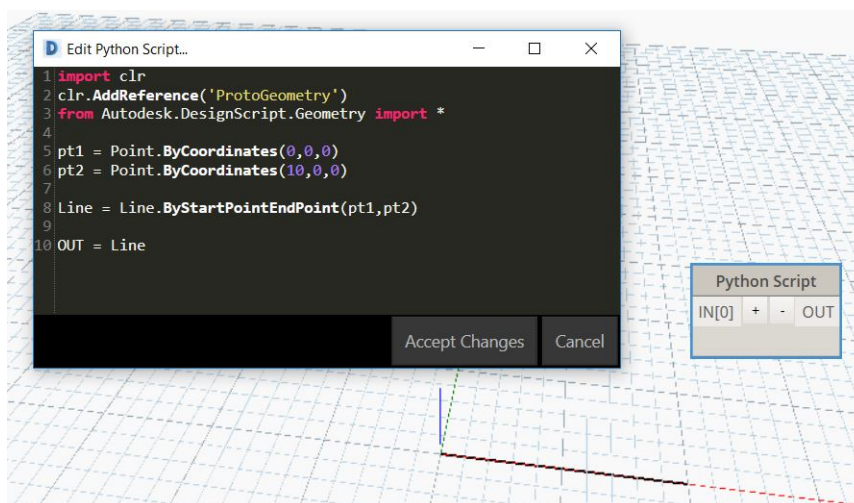


Figur 1.7: Aktiv- og inaktiv node

- *Wires* brukes til å sammenkoble nodene og etablerer flyten i det visuelle programmet. Wiren kobles fra output-porten i en node til input-porten i en annen for å overføre data fra en node til en annen. Arbeidsflyten fungerer alltid fra venstre mot høyre, nettopp på grunn av det er slik henholdsvis input- og output-portene er organisert.
- *Porter* er mottakerne for wirene og tar som oftest en spesifikk type data. Noden *Point.By.Coordinates* vil for eksempel forvente en double, kort forklart: et tall. Porter er også output for data og er alltid plassert på høyre siden av noden.
- *Script* er ofte sett på som en samling av koder og i Dynamo sitt tilfelle består et script av et sett med noder som er satt sammen i samme fil.



Figur 1.8: Visuell programmering



Figur 1.9: Tekstbasert programmering

Et eksempel hvor forskjellen for hvordan en linje med startpunkt og endepunkt lages ved hjelp av visuell programmering og vanlig tekstbasert programmering er illustrert henholdsvis i figur 1.8 og figur 1.9. Linjen for det tekstbaserte eksempelet er laget i et Python script, men vil være relativt lik uavhengig av hvilket programmeringsprogram man bruker.

Dynamo er et åpent kildekodeverktøy, bygget av et samfunn som bidrar til å gjøre programmet bedre. Dette betyr at brukere kan publisere sine *pakker* med egendefinerte noder og dele dem med andre brukere. I denne oppgaven er det benyttet flere pakker på grunn av manglende noder i Dynamo sin egen database. Disse pakkene er:

- archi-lab.net
- Clockwork for Dynamo 1.x
- LunchBox for Dynamo

Det er en kobling mellom Visual studio, et programmeringsprogram, og Dynamo til å designe egne noder. Et annet alternativ er å bruke *Python*-noden i Dynamo. I denne noden kan koder skrives direkte i Dynamo. Dersom man vil kode noder hvor innholdet arbeider med Revit, kan

Revit sitt programmeringsgrensesnitt benyttes. Dette er kalt Revit API (Application programming interface). Bruken av Revit API gjør det mulig å utvikle skript, der innholdet bestemmer hva programvaren skal gjøre. Revit API støttes av blant annet programmeringsspråkene C-sharp og IronPython, begge benyttet i denne oppgaven [12]. Egendesignede noder bistår der Dynamo har mangler.

Dynamoscriptene bærer ofte preg av å være uoversiktlig og store. Det er av den grunn hensiktsmessig å gruppere forskjellige deler av scriptet. Grupperingen kan gjøres ved hjelp av fargegrupper som lager et mer ryddig script med god oversikt. Gjennomgående i oppgaven er det anvendt fargekodene gitt i figur 1.10. Inspirasjonen til fargekodene er hentet fra White Arkitekter AB med sin Dynamo standard for hvordan de setter opp sine dynamoscript [13]. Noder og elementer benyttet i Dynamo og Revit skrives i *kursiv* gjennom oppgaven.



Figur 1.10: Fargekoder i Dynamo

1.3.3 Sofistik

Sofistik er et FEM-program med støtte for design av bygninger, bruer, stål og lettvektkonstruksjoner [14]. Sofistik støtter også geotekniske design deriblant fundamenter og tunneler. Sofistik håndterer spesielt avansert analyse og design av alle områder innen bruprosjektering.

Sofistik består av flere programmer, hvor Structural Desktop (SSD) kontrollerer kommunikasjonen mellom de individuelle programmene. Her styres pre-processing, processing og post-processing for prosjektene. All data fra de forskjellige programmene lagres i en felles database, *central data base (cdb*)* [15].

Sentrale program i Sofistik

- Sofiplus brukes til grafisk pre-processing. Sofiplus er basert på AutoCad, dermed muliggjør design av nesten hvilken som helst konstruksjon. I Sofiplus defineres akser og langs akser opprettes relevante punkter, omtalt *placements*. Konstruksjonen modelleres ved hjelp av AutoCad-linjer og punkt. Geometri, laster, opplagerbetingelser og spennarmering er eksempler på element Sofiplus kan modellere. Ved å eksportere til Sofistik vil det automatisk genereres et *mesh* av modellen. Det foreligger samtidig alternativ om manuell oppdeling av *mesh*.
- Teddy er et tekstbasert alternativ til Sofiplus for pre-processing. I Teddy brukes programmeringsspråket CADINP. Gjennom Teddy kan man programmere skript for design eller manipulasjon av eksisterende modell. En slik manipulasjon kan være å bestemme bæretningen til laster i en skallmodell. Teddyskripting er svært nyttig dersom eksempelvis senere prosjekter er av lignende type. Endring av parametere i skriptet utgjør ingen vanskelighet, men krever en viss forståelse. Ved eventuelle endringer av parametere kjøres ny analyse slik at generering av ny modell med gitte endringer i Sofistik gjennomføres. [16].
- Wingraf er et verktøy for post-processing. Wingraf viser tilgjengelige data grafisk. Her kan man for eksempel hente ut diagrammer for krefter og momenter i hele modellen [17].
- Ursula (Report Browser) er et annet verktøy hvor rapporter for post-processing lages. I Ursula er rapporter for alt fra materialer og spennarmering til resultat fra beregningsanalysene tilgjengelig.

Gjennom oppgaven er det tatt i bruk studentversjon av Sofistik 2018.

2 Brudesign

I samarbeid med COWI er en etteroppspent 70 meter lang enkel brukonstruksjon inspirasjon under modelleringen, se vedlegg A. Bruen er en trespenns bjelkebru, og skal plasseres over hovedveien på E18 i Rugtvedt-Dørdal. En etteroppspent brukonstruksjon er en brutype som blir spent opp etter støping. Her må armeringen ligge fritt i, eller tres gjennom utsparingskanaler i den støpte konstruksjonen. Etter at betongen er herdet tilstrekkelig vil armeringen spennes opp i den «aktive» delen og låses mot en innstøpt forankringsplate. Oppspenningen foregår med en oppjekking i den «aktive» delen og resulterer i direkte overføring av trykkraft i betongen og strekkraft i armeringen [18].

Da bruer ofte tar i bruk spennarmering, inkludert gitt bru, er det relevant med grunnleggende forståelse av spennarmeringen sin funksjon. Kapitlet vil dermed beskrive spennarmering, samt dens materialbetingelser tilknyttet analyseverktøy. Samtidig vil laster aktuelle for oppgaven beregnes, for bruk i tradisjonell- og parametrisk modellering.

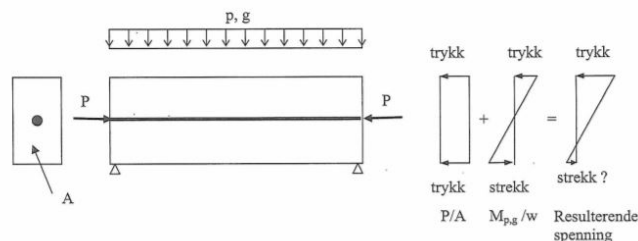
Under oppgaven er det benyttet følgende standarder.

- Prosjektering av betongkonstruksjoner etter NS-EN-1992-1-1:2004+NA:2008
- Allmenne laster etter NS-EN 1991-1-1:2002 + NA:2008
- Trafikklast etter NS-EN 1991-2:2003+NA:2010
- Vindlast etter NS-EN 1991-1-4:2005+NA:2009
- Temperaturlast etter NS-EN 1991-1-5:2003 + NA:2008
- Lastkombinasjoner etter NS-EN 1990:2002+A1:2005+NA:2016

Lastfaktorer i lastkombinasjonene finnes automatisk i Sofistik og blir dermed ikke beskrevet videre. Kombinasjonsreglene for veibruer finnes i kapittel A2.2.2 i standarden [19].

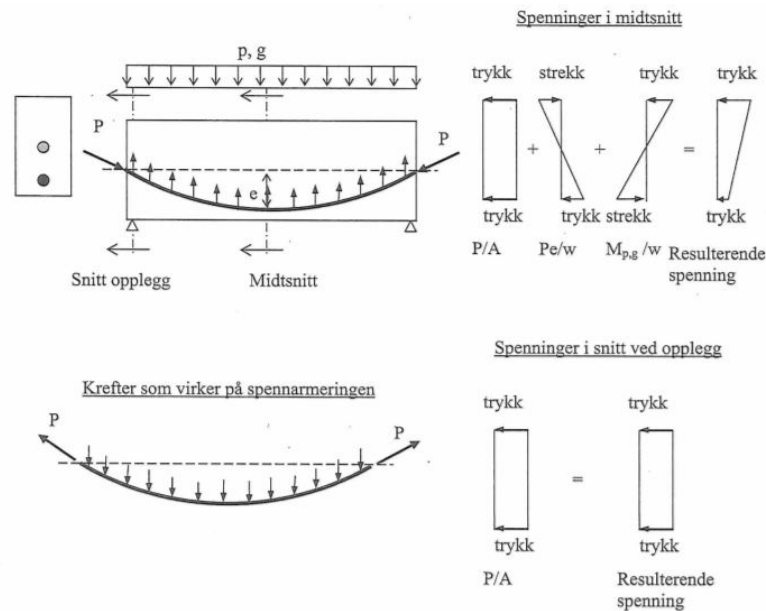
2.1 Spennarmering

Spennarmeringen påfører betongen trykkspenninger med størrelse og fordeling slik at ytre laster blir motvirket. For å oppnå trykkspenningene gis spennarmeringen en initialtøyning med strekkraft uavhengig av betongens tøyning. Den da oppnådde strekkraften i spennarmeringen overføres som trykk i betongen med endeforankringer av spennarmeringen, eller med heft mellom betong og spennarmering [18].



Figur 2.1: Påvirkning av sentrisk spennkabel i bjelke [18]

Plasseres spennarmeringen med eksentrisitet vil den «løfte» kreftene som blir påsatt en bjelke. Med andre ord kan man si at eksentrisiteten til spennkabelen i forhold til tyngdepunktet gir et moment. Plasseres spennkabelen der det er strekkmoment vil man kunne motvirke dette og danne et tverrsnitt i trykk. Effekten av plasseringen til spennarmeringen tydeliggjøres i figur 2.1 og figur 2.2.



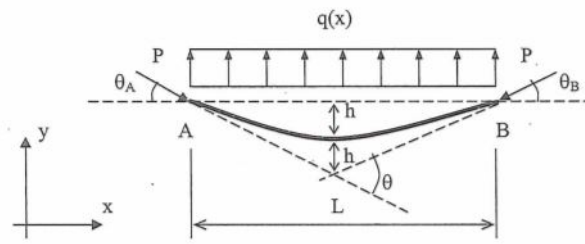
Figur 2.2: Påvirkning av parabelformet spennkabel i bjelke [18]

Fordelene av å bruke spennarmering kontra slakkarmering er at opprissing og rissvidder vil begrenses, et fordelaktig resultat når det kommer til bestandighet og tetthet. Bruer langs kysten som utsettes for høy mengde klorider er for eksempel avhengig av en tett konstruksjon for å fungere ut sin levetid. Med redusert oppriss øker bøyestivheten til konstruksjonen [18].

Spennarmering vil som nevnt bidra til å la betongen spenne lengre og øke stivheten dersom spennføringen er riktig. Føringens eksentrisiteter, kraft og vinkel bidrar til direkte lastvirkninger kalt *ekvivalente krefter*. Ekvivalente krefter beskriver spennarmeringens påvirkning på betongen ved hjelp av ytre krefter, se figur 2.4. Beregningen av ekvivalente krefter utføres ved å beskrive spennkabelføringen som en andregrads- eller lineær funksjon:

$$y(x) = ax^2 + bx + c \tag{1}$$

$$y(x) = ax + b \tag{2}$$



Figur 2.3: Beregning av ekvivalente krefter utgangspunkt [18]

Helningen av spennarmeringen kan beskrives av vinkelendringen, $\theta(x) = dy/dx$. Likevekt i y-retning gitt i figur 2.3 hjelper med beskrivelsen av ekvivalente krefter i y- retning, gitt av formel 3:

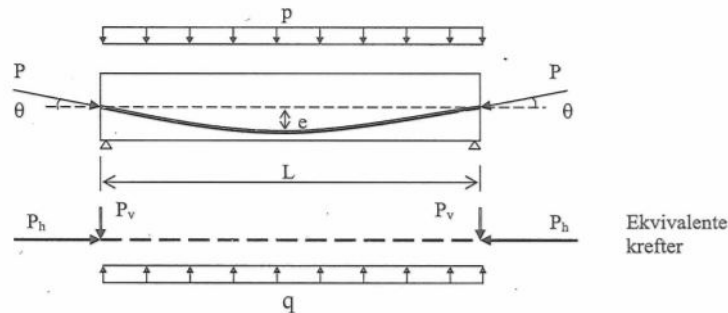
$$q(x) = P \frac{d\theta}{dx} = P \frac{d^2y}{dx^2} \quad (3)$$

For en parabelformet spennarmering vil $q(x) = P * 2a$, gitt av parabelfunksjon. Dette medfører at total last langs parabelen kan beskrives som:

$$\int_A^B q(x)dx = \int_A^B P \frac{d\theta}{dx} dx \quad (4)$$

Løses integralet resulteres det i ekvivalent kraft for parabelformet spennarmering gitt som:

$$q = \frac{8PH}{L^2} \quad (5)$$



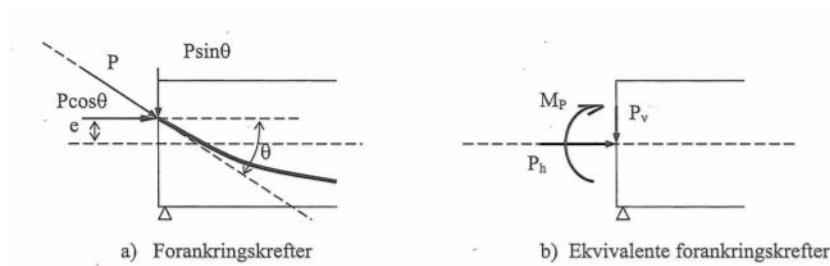
Figur 2.4: Ekvivalente krefter for parabelformet fritt opplagt bjelke [18]

I tillegg til ekvivalente kreftene langs spennarmeringen vil det oppstå ekvivalente krefter i forankringen, ofte omtalt *ekvivalente forankringskrefter*, se figur 2.5. Med enkel trigonometri vil disse bestemmes av:

$$P_h = P \cos \theta \approx P \quad (6)$$

$$P_v = P \sin \theta \approx P * \theta \quad (7)$$

$$M_p = P \cos \theta * e \approx P * e \quad (8)$$



Figur 2.5: Forankringskrefter [18]

Gitt en fritt opplagt bjelke med forutsetninger lik figur 2.4. Se vedlegg B for bestemmelser av e, L og P , samt Sofistik-resultat. Dette resultere i kreftene gitt i tabell 2.2 og 2.1.

Tabell 2.1: Verdier basert på ekvivalente krefter og bjelkemodell i Sofistik, parabelformet spennkabelføring

	<i>Håndberegninger</i>	<i>Sofistik</i>
M_{ed}	856.8 kNm	799.6 kNm
P_v	343 kN	223 kN, 214 kN
P_h	2865 kN	2847 kN, 2728 kN

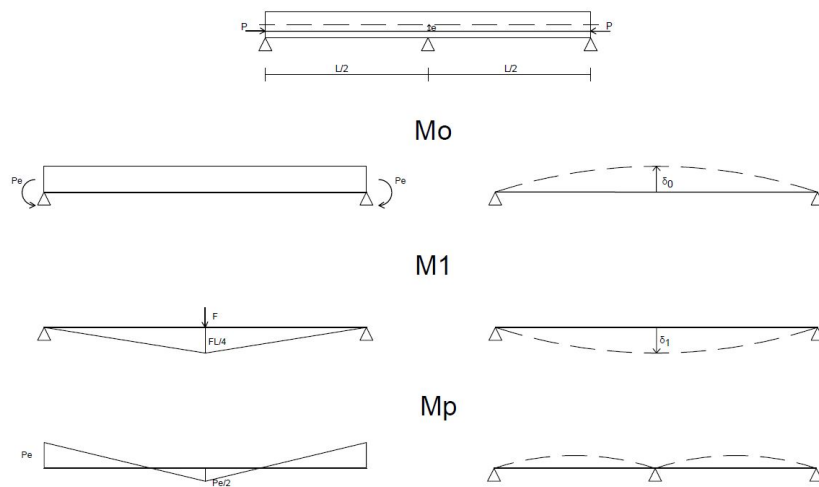
Tabell 2.2: Verdier basert på ekvivalente krefter og bjelkemodell i Sofistik. rett spennkabelføring

	<i>Håndberegninger</i>	<i>Sofistik</i>
M_{ed}	856.8 kNm	856.8 kNm, 847.9kNm
P_v	0 kN	0 kN
P_h	2856 kN	2856 kN, 2826kN

Ekvivalente krefter beregnes med konstant kraft, P , gjennom hele konstruksjonen. I tillegg tas det ikke hensyn til eksempelvis låsetap ved start av kabelen og dette resulterer i forskjellig krefter sammenlignet med beregningene i Sofistik. Tydeliggjort i beregnede ekvivalente krefter ser man at dette overestimerer spennkabelens effekt.

I tillegg til de direkte lastvirkingskreftene vil det oppstå tvangskrefter hvor bruene er statisk ubestemt. Tvangskrefter oppstår når bruene ikke kan deformeres fritt, men tvinges på plass. Et godt eksempel er ved en tofeltsbru der kabelføringen er lagt rettlinjert med eksentrisitet e . I en statisk bestemt bru, hvor for eksempel midtre opplegg ikke er tilstede, vil bruene deformere seg fritt og resultere i et jevnt moment over hele bruene basert på kraften til spennkabelen multiplisert med eksentrisiteten. Opplagerbetingelsen i midten av bruspennet hindrer bruene i å deformere seg, og det oppstår en tvangskraft i opplegget som gir opphav til et tvangsmoment i konstruksjonen, illustrert i figur 2.6 [18]. Her er M_0 primærmomentet, mens M_1 tvangsmomentet og det resulterende momentet M_p beskrives av uttrykket nedenfor.

$$M_p = M_0 + M_1 \quad (9)$$



Figur 2.6: Tvangskrefter

Det er hensiktsmessig å utplassere spennarmering etter momentforløpet til permanente laster.

Materialbetingelser

Ved bruk av spennarmering egner det seg å ta i bruk betong med høy fasthet grunnet påsatte trykkrefter, som blant annet gir høye lokale trykkspenninger ved konsentrert forankring. Det er av den grunn vanlig med fasthetsklasse B35-B55 eller høyere [20]. Videre må spennarmeringen klassifisere etter følgende begreper:

- f_{pk} - Karakterisert fasthet.
- $f_{p0.1k}$ - Spenning som gir 0.1% permanent forlengelse.
- e_{uk} - Tøyning ved maksimal spenning.
- $f_{pd} = f_{pk} / \gamma_s$ - Dimensjonerende fasthet.

Statens Vegvesen sin rapport 668 – *beregningsveiledning av etteroppsepente betongbruer* tilsier at det mest vanlige spennstålet benyttet i Norge er «1860-stål», som indikerer stålqualität Y1860S7 med materialbetingelser gitt nedenfor [20]. Andre bestemmelser kan hentes fra Eurokode 2 punkt 3.3.6 – *dimensjoneringsforutsetninger* [21].

- $f_{pk} = 1.860 \text{ MPa}$
- $f_{p0.1k} = 1.640 \text{ MPa}$
- $e_{uk} = 0.035$
- $E_p = 195.000 \text{ MPa}$

Tap i Spennarmering

En oppspent kable vil ikke ha den oppspente kraften gjennom hele dens lengde, det vil forekomme et tap. Det er tre hovedtyper av spennkrafttap [18].

- Tap av tøyningdifferanse mellom spennarmering og betong:
Tap som skyldes tilstrekkelig heft mellom spennarmeringen og betongen. Det er hoved-

sakelig i etteroppspent betong man finner denne form for tap. Tap i tøyingsdifferense deles videre inn i låse-, friksjons- og temperaturtap.

- Spenningsendring av korttidslast:
Tap forårsaket av at flere spennkabler spennes opp etter hverandre i etteroppspente konstruksjoner.
- Tidsavhengig tap:
Tap som skyldes kryp og svinn i betongen og relaksasjon av spennstålet.

Denne oppgaven vil ikke gå dypere inn på tapsberegninger av spennarmering. Videre hentes $K1 - K8$ verdier fra EK2 [5.10.2]. Verdiene kontrolleres opp mot forhåndsbestemte verdier i Sofistik eller, om nødvendig, legges til.

2.2 Laster

Laster klassifiseres etter variasjon over tid:

- Permanente påvirkninger
- Variable påvirkninger
- Ulykkespåvirkninger

I dimensjoneringsgrunnlaget benyttes karakteristiske verdier. Den karakteristiske lastpåvirkningen bestemmes av forventet lasthistorie og konstruksjonens dimensjonerende brukstid.

Kun de mest aktuelle laster for denne oppgaven vil bli beskrevet.

2.2.1 Permanente laster

Betong

Vekten av bruprofilet og søylene. Verdier for egenvektsberegning av armert betong er gitt etter 25kN/m^3 . Dette stemmer overens med Sofistik sin automatisk egenvektsberegning.

Belegning

For bruer skal belegning alltid inkluderes i egenlasten. Minimumskravet for dimensjonerende belegningsvekt finnes i Håndbok N400 - Bruprosjektering og er avhengig av største spennvidde for bruene, vist i tabell 2.3 [22]. Største spennvidde på bruene er 30 meter og belegningsvekten er dermed $3,5\text{ kN/m}^2$. Med 7,5 m bredde på kjørebane blir vekten fra belegningsvekten $26,25\text{ kN/m}$. Beregningene er vist i vedlegg C.

Tabell 2.3: Dimensjonerende belegningsvekt i kjørebane [22]

Største spennvidde [m]		
$l \leq 50$	$50 < l \leq 200$	$l > 200$
$3,5\text{ kN/m}^2$	$2,5\text{ kN/m}^2$	$2,0\text{ kN/m}^2$

2.2.2 Variable laster

Trafikklaster

Bruen dimensjoneres for fire forskjellige lastmodeller [23]:

- Lastmodell 1: Konsentrert og jevnt fordelt last som dekker det meste av effekter fra biler og lastebiler. Modellen brukes for generell og lokal verifikasjon.
- Lastmodell 2: En enkel akslinglast som påføres bruene. Denne modellen skal dekke de dynamiske effektene for normal trafikk på korte bruer.
- Lastmodell 3: Flere akslinglaster som representerer spesielle kjøretøy som kan kjøre på veier ment for unormale laster. Lastmodellen brukes for generell og lokal verifikasjon. Relevant kjøretøy av spesiell variant defineres.
- Lastmodell 4: Last som representerer en samling av mennesker. Lastmodell brukes for generell verifikasjon. Hvis relevant legges en jevnt fordelt last på 5 kN/m^2 .

I denne oppgaven benyttes kun Lastmodell 1 til forenkling. Trafikklasten legges inn med Sofistik sin egen oppgave for trafikklast. Verdiene i tabell 2.4 stemmer med lastene Sofistik beregner. Beregningene er vist i vedlegg D.

Lastmodell 1 består to deler:

- Dobbel akslingslaster i tandemsystem hvor hver aksling har to konsentrerte punktlaster.
- Jevnt fordelt last som virker over hele kjørebanelen.

Tabell 2.4: Lastmodell 1

Lastfelt	Q_{ik}	B	q_{ik}
1	300 kN	3,0 m	$5,4 \text{ kN/m}^2$
2	200 kN	3,0 m	$2,5 \text{ kN/m}^2$
Rest	-	1,5 m	$2,5 \text{ kN/m}^2$

Snølast

For bruer regnes ikke snølaster å opptre samtidig med trafikklast.

Vindlast

Vindlast på bruer bestemmes ut i fra ulike vindlastklasser [24]. Disse klassene baserer seg på den dynamiske lastvirkningen vinden har på bruene. For bruene man beregner vil den dynamiske lastvirkningen fra vind være ubetydelig og vindklasse I vil være aktuell.

Vindlast beregnes etter SVV rapport 668 Beregningsveiledning for etteroppspente bruer hvor NS-EN 1991-1-4:2005+NA:2009 benyttes. Beregningene er vist i vedlegg E.

I denne oppgaven deles vindlast på bruene inn i tre:

1. Horisontal vindlast i tverretning:

Horisontal vindlast er regnet med lastflate på 8.0m

$$f_x = 7.20 \text{ kN/m} \tag{10}$$

2. Vertikal vindlast:

Den vertikale lasten vil kunne virke enten nedadrettet eller oppadrettet og legges inn med en eksentrisitet fra bruens senterlinje på $B/4 = 2.125\text{m}$. Beregnet verdi:

$$f_z = 4.88\text{kN/m} \quad (11)$$

Vertikal vindlast i denne oppgaven legges inn nedadrettet og posisjonert slik at den vil gi lik bøyning om lengdeaksen som horisontal vindlast.

3. Vindlast på søyler:

Vindlast på søyler skal påføres i den retningen som er mest ugunstig på hele konstruksjonen etter NS-EN 1991-1-4. Det vil si at vindlasten påføres i tverretningen på bruene det fokuseres på i denne oppgaven.

For yttersøyler er lasten beregnet fra søyler med tverrsnitt $b \times d = 800 \times 3900\text{mm}$, hvor b er siden vinden angriper på. Høyden er satt $h = 6500\text{mm}$

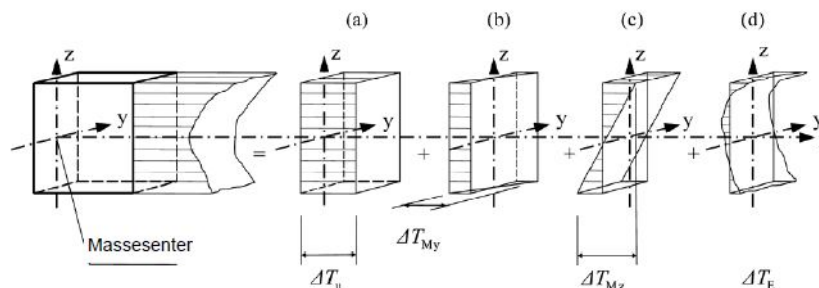
$$f_x \cdot s = 0,41\text{kN/m} \quad (12)$$

En utfordring for den parametriske modellen er dersom tverrsnittet for søyle eller brukassen endres. Lastarealet kan av den grunn bli større eller mindre og feil vindlast blir lagt inn dersom dette ikke endres i beregningene. Forenklet ligger geometri fra COWI til grunn for beregning av størrelse på vindlaster.

Temperaturpåvirkning

Termisk påvirkning bestemmes etter NS-EN 1991-1-5:2003 + NA:2008 [25]. Ved beregning av termisk påvirkning skal enn se på følgende fem virkninger, illustreres ved fire punkter i figur 2.7. Virkningene gjelder for en type 3 betongbru, samsvarende med oppgavekonstruksjonen.

1. Jevnt fordelt temperaturandel.
2. Vertikalt lineært eller ikke-lineært varierende temperaturandel.
3. Horisontalt lineært varierende temperaturandel.
4. Forskjell i jevnt fordelt temperaturandel mellom konstruksjonsdeler.
5. Temperaturdifferanse over veggtykkelsen og mellom utvendige og innvendige vegger i kassetverrsnitt.



Figur 2.7: Fordeling av temperatur i et tverrsnitt [25]

Ved prosjektering av enkel betongbru, med massivt tverrsnitt, hvor tverrmoment er minimalt vil kun de to første virkningene være relevante. Vertikal varierende temperaturandel antas derfor å være lineær [20]. Denne oppgaven fokuserer på enkel brudimensjonering og vil derav kun ta høyde for de to første punktene.

1. Jevnt fordelt temperaturandel:

Jevnt fordelt temperaturandel henger sammen med bruens lavest og høyeste temperatur. I dette området gis en jevnt fordelt temperaturendring. For en ikke-fastholdt konstruksjonsdel vil dette føre til endring av dens lengde.

For bestemmelse av $\Delta T_{N,con}$ og $\Delta T_{N,exp}$ trengs initialtemperaturen til bruene på det tidspunktet den blir fastholdt (T_0), samt øvre- og nedre representativ lufttemperatur T_{maks} og T_{min} . Disse kan uthentes fra det nasjonale tillegget i EK-1-5, og vist i vedlegg F. For bruene gitt av COWI, vil det etter geografiske hensyn gi verdiene i likning 13 og 14

$$\Delta T_{N,con} = T_{min} - T_0 = -37 \quad (13)$$

$$\Delta T_{N,exp} = T_{maks} - T_0 = 23 \quad (14)$$

2. Vertikalt lineært varierende temperaturlast:

Vertikal lineært varierende temperaturlast tar hensyn til temperaturdifferansen mellom oversiden og undersiden av betongbruene. For innhenting av verdier $\Delta T_{M,heat}$ og $\Delta T_{M,cool}$ må forutsetninger knyttet til tverrsnittet være gitt, samt overflatebelegg slik at korrigeringsfaktoren k_{sur} kan bestemmes. Bruene tilknyttet denne oppgaven er av typen platebjelketverrsnitt slik at karakteristiske verdier $\Delta T_{M,heat}$ og $\Delta T_{M,cool}$ bestemmes ut fra det. Uten god nok informasjon til tykkelse av belegg har man tatt utgangspunkt i anbefalinger fra «Beregningsveileder for etteroppspente betongbruer» [20]. Positiv verdi indikerer varm overside.

$$\Delta T_{M,heat} = k_{sur} * \Delta T_{M,heat} = 10.5 \quad (15)$$

$$\Delta T_{M,cool} = k_{sur} * \Delta T_{M,cool} = -8 \quad (16)$$

Temperaturlast i analyse

Temperaturlastene kombineres til åtte ulike kombinerte lasttilfeller etter ω_n og ω_m i gitte formler nedenfor. De åtte ulike kombinasjonene modelleres inn i den analytiske modellen som karakteristiske laster [20]. Kombinasjonene kan ikke opptre samtidig. Hvert tilfelle blir derfor vurdert etter hvor ugunstig det er for hvilket resultat man er ute etter.

$$\Delta T_{M,heat}(\text{eller} \Delta T_{M,cool}) + \omega_n * \Delta T_{N,exp}(\text{eller} \Delta T_{N,con}) \quad (17)$$

$$\omega_m * \Delta T_{M,heat}(\text{eller} \Delta T_{M,cool}) + \Delta T_{N,exp}(\text{eller} \Delta T_{N,con}) \quad (18)$$

Komb	$\Delta T_{M, heat}$	$\Delta T_{M, cool}$	$\Delta T_{N, exp}$	$\Delta T_{N, con}$
1	1,0		0,35	
2	0,75		1,0	
3	1,0			0,35
4	0,75			1,0
5		1,0	0,35	
6		0,75	1,0	
7		1,0		0,35
8		0,75		1,0

Figur 2.8: Kombinasjonstabell for jevnt fordelt- og vertikalt lineært varierende temperaturandel [20]

Karakteristiske verdier for temperatur er beskrevet i vedlegg F. Laster for horisontal temperaturredifferanse, samt temperatur i søyler er ikke tatt hensyn til under modelleringen.

Tabell 2.5: Oppsummert karakteristiske laster

Lasternavn	Verdi	Benevning
Egenvekt	159,5	kN/m
Egenvekt belegning	26,25	kN/m
Egenvekt kantdragere x2	7,4	kN/m
Vindlast horisontal	7,2	kN/m
Vindlast vertikal	4,88	kN/m
Vindlast søyle	0,41	kN/m
Temp 1	8,05/10,5	°C jevnt/lineært
Temp 2	23/7,875	°C jevnt/lineært
Temp 3	-12,95/10,5	°C jevnt/lineært
Temp 4	-37/7,875	°C jevnt/lineært
Temp 5	8,05/-8	°C jevnt/lineært
Temp 6	23/-6	°C jevnt/lineært
Temp 7	-12,95/-8	°C jevnt/lineært
Temp 8	-37/-6	°C jevnt/lineært

Egenvekt av bruene er beregnet av Sofistik (første i tabell 2.5. For trafikklaster se tabell 2.4.

3 Element i analyse

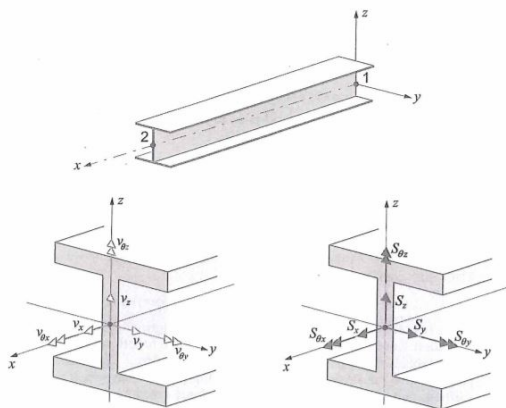
FEM-modellen defineres ut fra bruvalget, hvor man primært beskriver konstruksjonen ut fra tre hovedvalg av elementer i elementprogrammer; bjelkeelementer, skallelementer og solidelement.

I tillegg til elementvalg bør man ved modellering kontrollere programmets definisjon av lastvalg. Dette innebærer valget mellom *last lumping* eller *konsistent last vektor*. Enkelt forklart betyr lastlumping at laster flyttes inn til nodene og uttrykkes deretter. Metoden trenger mindre beregningstid, men er samtidig mindre nøyaktighet. Konsistent lastvektor bestemmes ut fra integralet til elementets formfunksjoner og lastvirkning på elementet. Anvendelsen av konsistent lastvektor i beskrivelse av lasten vil gi bedre nøyaktighet, men ved et tett *mesh* vil lastlumping gi tilnærmet like gode resultater. Programvaren Sofistik tar i bruk lastlumping, en fordelaktig metode ved tunge beregninger deriblant trafikklast.

For modelleringen sees det vekk fra bruken av solid elementer og er derfor ikke beskrevet ytterligere. Kapittelet vil videre ta for seg kort beskrivelse av resterende to elementene, samt vurderinger knyttet opp mot elementvalg.

3.1 Bjelkeelement

Bjelkeelementer, ofte omtalt som rammeanalyse, vil være enklere å modellere, forstå og kvalitetsikre gjennom elementets enkelhet. Bjelkeelementene forutsetter udeformert tverrsnitt, en forenkling man må ta høyde for. Ved å analysere bruene isolert i tverretningen og deretter summere samlet armeringsmengdene tas «feilen» med, men en slik metode kan være svært tidkrevende [20].



Figur 3.1: Frihetsgrader og krefter bjelkeelement [6]

$$\begin{bmatrix} S_x^* \\ S_{\theta y1}^* \\ S_{\theta y2}^* \\ S_{\theta z1}^* \\ S_{\theta z2}^* \\ S_{\theta x}^* \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4+\alpha_y b_y}{1+\alpha_y} & \frac{2-\alpha_y b_y}{1+\alpha_y} & 0 & 0 & 0 \\ 0 & \frac{2-\alpha_y b_y}{1+\alpha_y} & \frac{4+\alpha_y b_y}{1+\alpha_y} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{4+\alpha_z b_z}{1+\alpha_z} & \frac{2-\alpha_z b_z}{1+\alpha_z} & 0 \\ 0 & 0 & 0 & \frac{2-\alpha_z b_z}{1+\alpha_z} & \frac{4+\alpha_z b_z}{1+\alpha_z} & 0 \\ 0 & 0 & 0 & 0 & 0 & t \end{bmatrix} \begin{bmatrix} v_x^* \\ v_{\theta y1}^* \\ v_{\theta y2}^* \\ v_{\theta z1}^* \\ v_{\theta z2}^* \\ v_{\theta x}^* \end{bmatrix}$$

Figur 3.2: Stivhetsmatrise 3D-bjelkeelement [6]

Et bjelkeelement er bestående av seks frihetsgrader per node, som ved analyse medfører tilhørende seks krefter i bjelkesnittet, se figur 3.1. Stivhetsmatrisen til bjelkeelementet er gitt i figur 3.2, hvor beskrivelsen av variablene knyttet til matrisen er gitt i figur 3.3 [6].

$$a = \frac{EA}{L}, \quad b_y = \frac{EI_y}{L}, \quad \alpha_y = \frac{12\kappa_z EI_y}{GAL^2}, \quad b_z = \frac{EI_z}{L}, \quad \alpha_z = \frac{12\kappa_y EI_z}{GAL^2}, \quad t = \frac{GI_t}{L}$$

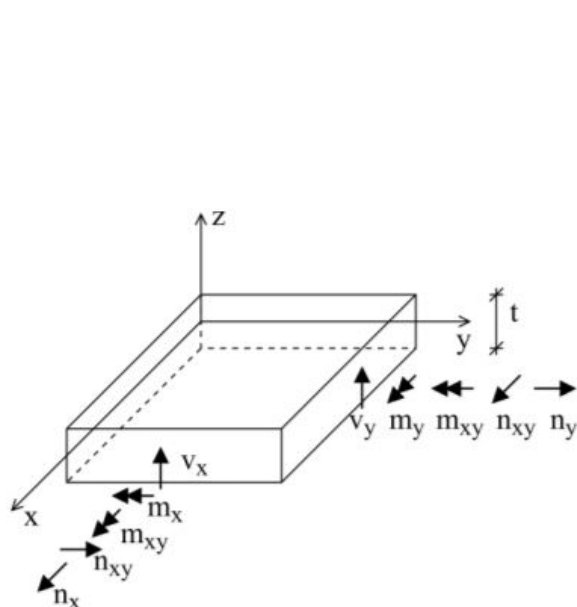
Figur 3.3: Bestemmelser av variabler gitt i bjelkestivhetsmatrisen [6]

Ved analyse av tre-dimensjonalt bjelkeelement vil det oppstå krefter i hvert bjelkesnitt beskrevet i punktene nedenfor.

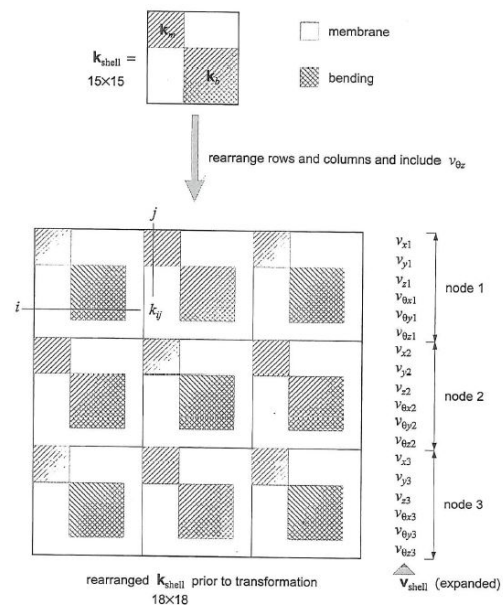
- Aksialkraft $N(S_x)$
- Bøyemoment $M(S_{\theta_y})$
- Vertikal skjær $V(S_z)$
- Torsjon $T(S_{\theta_x})$
- Tverrmoment $M_t(S_{\theta_z})$
- Tverrskjær $V_t(S_y)$

3.2 Skallelement

Skallelement er en kombinasjon av plate- og membranelementer. Det resulterer i et element med fem frihetsgrader per node og tilhørende fem resultanterkrefter per flate, se figur 3.4 [6][26].



Figur 3.4: Krefter i skallelement



Figur 3.5: Triangulær skallmatrise [6]

I figur 3.5 er k_m stivhetsmatrisen til membrandelen og k_b stivhetsmatrisen til bøyingsdelen. For et triangulært element kan k_m og k_b henholdsvis være CST-element og DKT-element [6]. Kraftene for et rektangulært skallelement er beskrevet av:

- Aksialkraft i lengderetning n_x
- Aksialkraft i tverretning n_y
- Skivekjærkrefter n_{xy}
- Bøyemoment om lengderetningen m_x
- Bøyemoment om tverretningen m_y
- Skjærkraft i lengderetningen v_x
- Skjærkraft i tverretningen v_y
- Torsjonsmoment m_{xy}

Kreftene og momentene er angitt som henholdsvis kN/m og kNm/m .

En skallmodell oppdeles i *mesh* for å gi mer nøyaktighet, hvor anvendelsen av *adaptiv mesh* er hensiktsmessig i kritiske områder. Et mesh er en oppdeling av analytiske elementer, hvor elementet er bestående av n -antall frihetsgrader basert på elementvalget. Et adaptiv mesh er et tettere mesh i kritiske områder, men ikke over hele konstruksjonen slik at beregningstiden ikke forstørres unødvendig. Man oppnår tilnærmet nøyaktige spenninger dersom bruken av element og mesh er riktig. Et element som er uendelig oppdelt eller har uendelig med frihetsgrader vil gi helt nøyaktig resultat, men uendelig oppdeling eller uendelig frihetsgrader gir lengre beregningstid. Ved søyleopplegg eller store tverrsnittsendringer vil det være fordelaktig å ta i bruk skallelementer, og ha et mer oppdelt mesh grunnet mer konsentrerte krefter.

Med skallelementer bør spenningene hentes fra integreringspunktene. Det vil si at man bør hente dem fra punkter mellom elementet. Deformasjon er nøyaktig ved noden. Verdier for deformasjon bør derfor hentes fra nodene.

3.3 Valg av element

Klassifisering av plate- eller bjelkebru er ikke alltid intuitivt. EK2 punkt 5.3(1) gir en kort klassifisering av konstruksjonsdeler, men det er likevel ikke alltid enkelt å bestemme hvordan bruene skal modelleres. Derfor bør tre forhold vurderes når en velger å modellere bruene med skall- eller bjelkeelement:

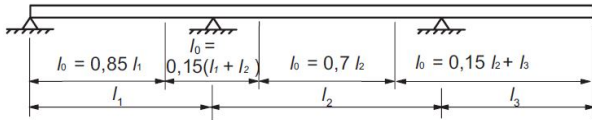
- Forhold mellom tverrsnittets høyde og bredde
- Forhold mellom tverrsnittets bredde og spennvidden i lengderetningen til brua
- Opplagerbetingelser

En bru med et lite forhold mellom spennvidde og bredde, det vil si vesentlig lengre enn bred, kan betraktes som bjelkebru siden kreftene bæres i lengderetningen [20]. Et tverrsnitt med vesentlig større bredde enn høyde vil betraktes som platebru og modelleres deretter.

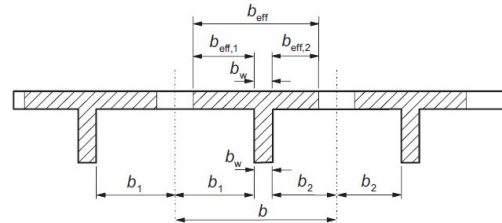
Det kan foretrekkes å ta i bruk skallelementer i bruanalyse, da dette gir flere relevante krefter direkte. Bruk av skallelementer bidrar til større nøyaktighet i deformasjon og spenninger. Likevel vil bruk av skallelement ofte gi en mer komplisert FEM-modell. Bjelkeelementer vil som nevnt bidra til enkle og forståelige resultater. Å tolke krefter modellert med bjelkeelement er derfor ofte en mer foretrekkende praksis.

Virkingen av effektiv flensbredde

For rammeanalyse må effektiv flensbredde bestemmes. Metoden for hvordan dette bestemmes er gitt i EK2 punkt 5.3.2.1, se figur 3.7



Figur 3.6: Bestemmelse av l_0
EK2 pkt [5.3.2.1]



Figur 3.7: Effektiv flensbredde
EK2 pkt [5.3.2.1]

Bøyesspenninger vil bli ujevnt fordelt over tverrsnittet, hvor spenningene er størst ved steget og ikke eksisterende helt ytterst ved flensen. Denne effekten er størst der skjærkraften i lengderetningen er størst, altså ved opplegg. Ved skalleanalyse benyttes hele bredden, mens for bjelkeelementer vil man som nevnt benytte effektiv flensbredde for å bestemme kreftene. Det tilsier at dersom $b_{eff} < b_{full}$ vil det oppstå større spenninger over steget i tverretning. Dette betyr at med en skallelementanalyse oppnår man mindre bøyesspenninger i dette området. Det resulterer i mindre nødvendig armering, samt tverrsnittshøyde.

Elementoverføring mellom Revit-Sofistik

Innledningsvis ble ulike konstruksjonerdelar og tilhørende elementer i Revit introdusert. Kort rekapitulering av dette:

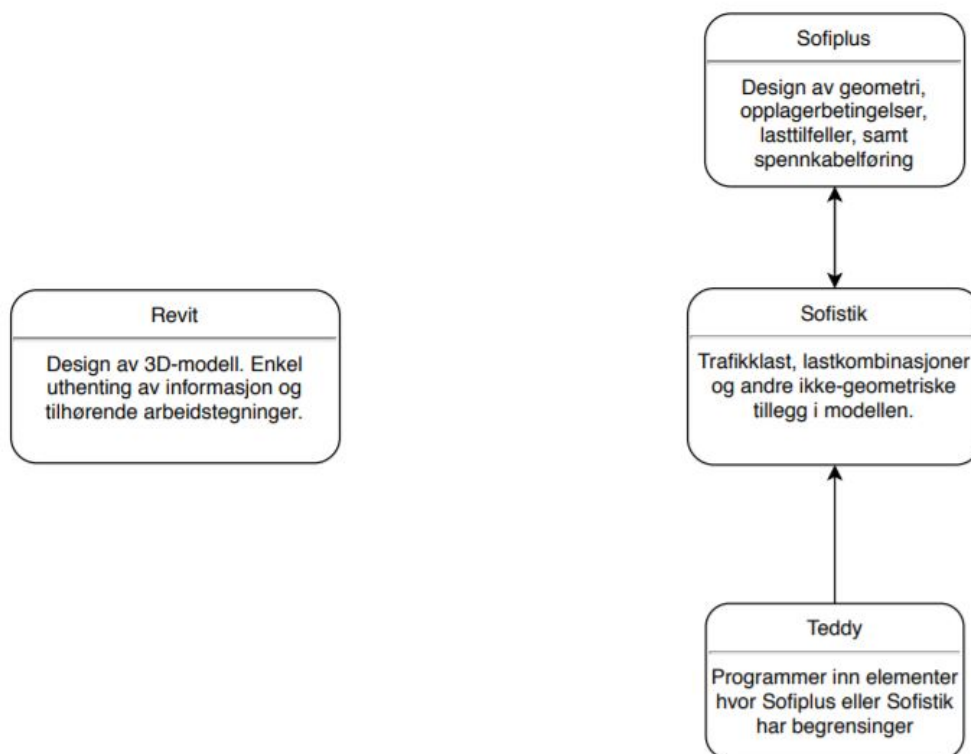
- Bjelker gir bjelkeelement
- Dekke/vegg gir skallelement

I oppgaven vil det etableres en parametrisk modell. I denne fasen trengs et parametrisk tverrsnitt hvor det opprettes en analytisk del. Revit har begrenset familier hvor dette er mulig. Den familien best egnet for å skape et slikt tverrsnitt er *Structrual Framing*. Familien er en linje-basert bjelke hvor det opprettes et tilhørende bjelkeelement i den analytiske modellen. Dette gjør at bruene kan lages ved å følge bestemt veilinje, mer om dette i senere kapitler.

De analytiske familiene for dekker har begrensinger hvor tverrsnittet ikke er spesielt anlagt for utforming, samt at de ikke kan følge en bestemt veilinje. På grunn av disse begrensningene vil oppgaven kun fokusere på brumodellering med bjelkeelement. For bruene gitt av COWI med lite forhold mellom spennvidde og bredde er modellering ved bruk av bjelkeelement et godt valg, av hensyn til tverrsnittsforholdene nevnt over.

4 Tradisjonell modellering

Som grunnlag til parametrisk modellering og flyt mellom programvare er det vesentlig å modellere brumodeller på tradisjonell måte i Sofistik, for å senere ha mulighet til sammenligning av resultat. Hensikten er å senere etablere en tilsvarende modell gjennom parametrisk design. Sammenligningsgrunnlaget vil være om geometri, laster og resultater overensstemmer med modellen designet på tradisjonell metode.



Figur 4.1: Tradisjonelt flytdiagram

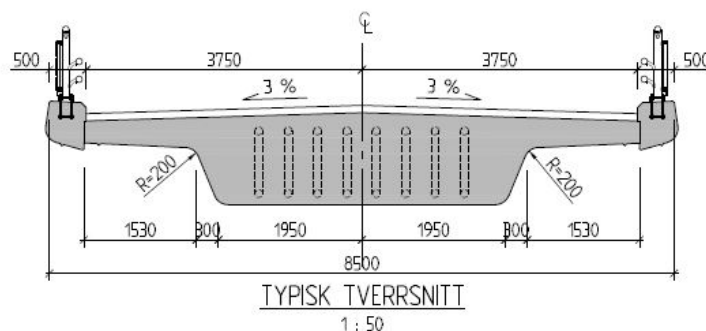
4.1 Modelleringsgang i Sofistik

Tidligere nevnt er et av modelleringsvalgene knyttet til bjelke- eller skallelementer. Lengde/bredde-forholdet for bruene tilsier at anvendelse av bjelkeelementer er tilsynelatende uproblematisk, da fokusområdet til modellen hovedsakelig er momentkrefter om lokal y-akse. Samtidig ble det nevnt tidligere at en overføring av skallelement mellom Revit og Sofistik er problematisk. Modellering med skallelementer vil derfor være mindre egnet i direkte sammenligning. På bakgrunn av disse faktorene velges det å modellere bruene med bjelkeelementer.

Utgangspunkt for modelleringen er inspirasjonsbru gitt av COWI. Her blir tverrsnittsdimensjoner og spennavstander hentet ut, se vedlegg A. Laster beregnet i tidligere kapittel er grunnlag for lastpåvirkningen.

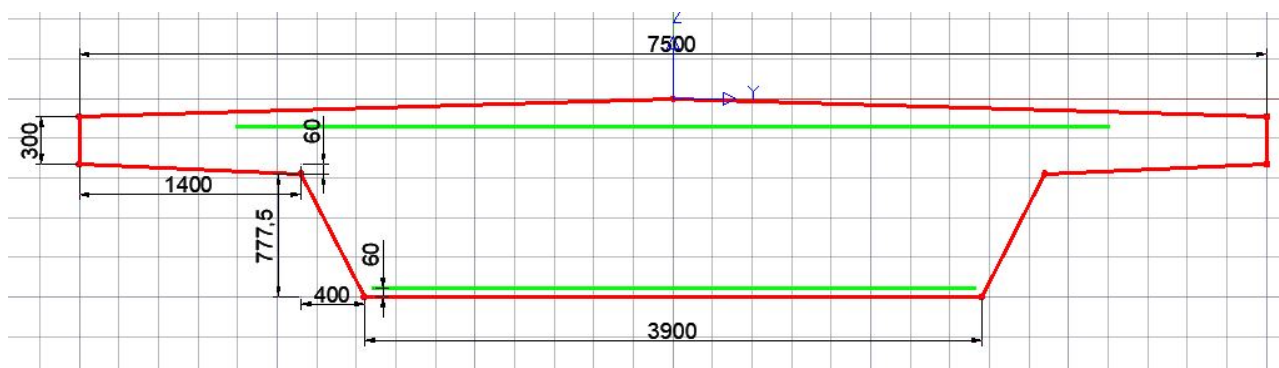
4.1.1 Designfasen

Designfasen starter i programvaren Sofiplus, nevnt innledningsvis. Sofiplus er et linjebasert og arealbasert tegneprogram som gir grunnlag til geometri, opplagerbetingelser og laster for Sofistik. Av hensyn til at Sofiplus er et tegneprogram vil det foreligge individuelle tegningsmetoder for å etablere ønsket analytisk modell.



Figur 4.2: Typisk tverrsnitt gitt av COWI, benyttet som inspirasjon under modellering

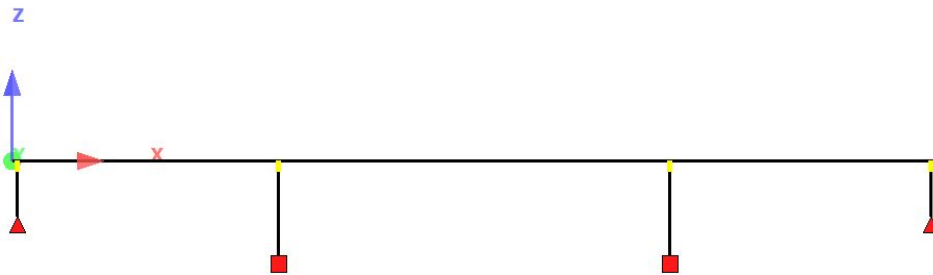
Tverrsnittinspirasjon fra COWI er vist i figur 4.2. Tverrsnittet forenkles i analysemodellen ved å fjerne rekkverk med kantdrager, samt avrunding mellom flens og steg. Disse forenklingen utgjør ingen nevneverdig analytisk forskjell. Utgangstverrsnitt for analysemodellen illustreres i figur 4.3. Sofistik har egenskapen til å beregne nødvendig armering for et gitt tverrsnitt og det er valgt å ta i bruk dette verktøyet til å sammenligne opp mot egne håndberegninger. Dette medfører at i design av tverrsnitt må det tilrettelegges for disse beregningene ved at det legges til armeringsområder i tverrsnittet.



Figur 4.3: Tverrsnitt benyttet i analytisk modell, forenklet versjon

Etablering av akser og plasseringer bestemmes for å henholdsvis ha linjer og referansepunkter i modellen. Aksen kan senere benyttes til generering av trafikklaster, samt spennkabelføring og er dermed vesentlig for brumodellen. Aksegrunnlaget er forenklet til rettlinjert veiprofil hvor bruspennet er hentet fra gitt bru. Bruprofilen plasseres i henhold til gitt akse, samt søyler etter gitt plassering i forhold til bruprofilen. Bestemmelser av forbindelsen mellom bruprofil

og søyler er basert etter spennteknikk sin brosjyre for lagersystem [27]. Lagersystemene deles opp i tre typer: fastlager, ensidig bevegelig lager og allsidig bevegelig lager. Ettersom modellen er bjelkemodellert vil det kun være ett fastholdingspunkt mellom konstruksjonsdelene. Bruen utføres som et monolittisk system i midtre søyleforbindelse, uten behov for lager. Første og siste forbindelse anvendes ensidig lager slik at konstruksjonen kan utvides i bruens lengderetning. Spennteknikk-brosjyre side seks redegjør for alternativt anvende fastlager i midtre søyleforbindelse med usikre grunnforhold [27]. Foruten forbindelse mellom konstruksjonen er det i søyleender valgt fastlager på ytre søyler, samt fast innspenning på midtre søyler. Endelig statisk system illustreres i figur 4.4.



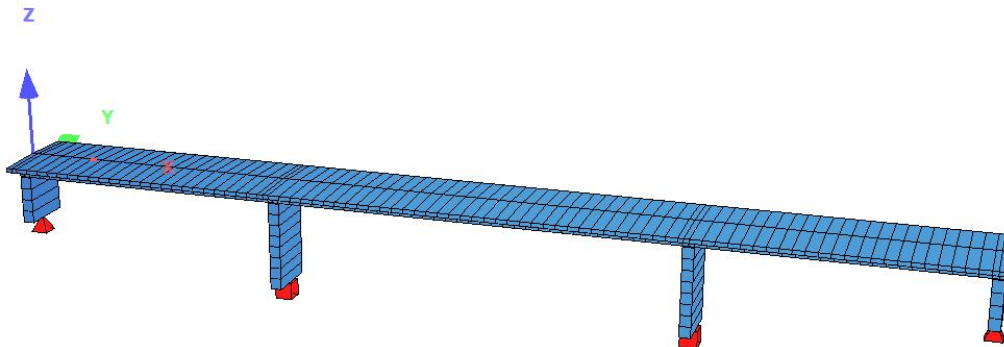
Figur 4.4: Statisk system for brumodell

Lastilfeller med tilhørende laster, beregnet i kapittel 2.2, legges inn i Sofiplus. Dette er ikke inkludert trafikklast, men lasttilfelle for trafikklast må likevel opprettes i Sofiplus for å senere tilegne trafikklasten. Karakteristiske laster er oppgitt i tabell 4.1.

Tabell 4.1: Karakteristiske laster

	<i>Nummer</i>	<i>Navn</i>	<i>Verdi</i>
Egenvekt	1	G_1	159.5 kN/m
Egenvekt-aditional	2	G_2	41.1 kN/m
Vind horisontal	20	W	7.20 kN/m
Vind vertikal	21	W	4.80 kN/m
Vind søyle	22	W	0.41 kN/m
Temperatur kombinasjon	190	T	$\Delta T_{M,heat} + \omega_n * \Delta T_{N,exp}$
Temperatur kombinasjon	291	T	$\Delta T_{M,heat} * \omega_m + \Delta T_{N,exp}$
Temperatur kombinasjon	392	T	$\Delta T_{M,heat} + \omega_n * \Delta T_{N,con}$
Temperatur kombinasjon	493	T	$\Delta T_{M,heat} * \omega_m + \Delta T_{N,con}$
Temperatur kombinasjon	594	T	$\Delta T_{M,cool} + \omega_n * \Delta T_{N,exp}$
Temperatur kombinasjon	695	T	$\Delta T_{M,cool} * \omega_m + \Delta T_{N,exp}$
Temperatur kombinasjon	796	T	$\Delta T_{M,cool} + \omega_n * \Delta T_{N,con}$
Temperatur kombinasjon	897	T	$\Delta T_{M,cool} * \omega_m + \Delta T_{N,con}$
Trafikklast TS	101-112	L_T	
Trafikklast UDL	201-212	L_U	

4.1.2 Resultatfasen



Figur 4.5: Sofistikmodell etablert med Sofiplus

Modellen tegnet i Sofiplus eksporteres så videre til Sofistik, se figur 4.5. I Sofistik anvendes *Traffic Loader*, et innebygd verktøy for beregning av krefter basert på trafikklast. Traffic Loader krever en referanseakse og at lineær analyse er beregnet på forhånd slik at stivhetsberegningene i modellen er unnagjort. I Traffic Loader bestemmes ulike lasttilfeller fra lastmodell 1, deriblant akslinglast, jevnt fordelt trafikklast og jevnt fordelt trafikklast med bremselast.

Lastkombinasjoner for brudd- og bruksgrensetilstand legges inn i egen oppgave i Sofistik. Fra kombinasjonene evalueres høyeste opptredende momenter i brukonstruksjonen, samt deformasjon i lengderetning. Resultat er oppgitt i tabell 4.3. For å kontrollere at modellen gir nøyaktig nok resultat, lages syv forskjellige mesh med varierende elementstørrelse. Dette gir også grunnlag til å bestemme elementstørrelse på mesh i den parametriske modellen. Samtidig evalueres tidsomfanget av beregningene for de forskjellige elementstørrelsene, for å unngå unødvendig tett mesh. Påvirkningen av elementenes tetthet er gitt i tabell 4.2.

Tabell 4.2: Effekten av mesh

Mesh størrelse	Maks M_y [kNm]	Min M_y [kNm]	Tid analyse [s]
10m	21030	-29385	18
4.0m	21030	-29385	23
2.0m	21030	-29385	34
1.0m	21030	-29385	48
Automatisk	21030	-29385	53
0.5m	21055	-29385	84
0.25	21055	-29385	271

Mesh med minst elementstørrelse tar klart lengst tid og dette er mye på grunn av trafikklastberegningen. Trafikklasten beregnes for flere plasseringstilfeller. Dette er en langvarig iterasjon

avhengig av antall elementer. Store endringer i elementoppdelingen gir likevel ingen nevneverdig forskjell i minimumsmomentet, som opptrer over støtte. Bakgrunnen for dette er når plasseringer for referansepunkt legges inn i Sofiplus, vil Sofistik dele opp element i nærheten i mindre element, og derav god nøyaktighet for alle elementstørrelser av mesh. I feltet mellom søylene hvor maksimum moment oppstår er det liten endring knyttet til tettheten av elementoppdelingen.

Tabell 4.3: Avleste verdier

	Verdi	Plassering
Med.min	-29385 kNm	Over midtre søyler brustart.
Med.max	21055 kNm	Felt mellom midtre søyler.
Ux.max	14mm	Brustart.
Ux.min	-14mm	Bruende
Uz.max	2.0mm	Rett før midtre søyle, ca 18m fra brustart
Uz.min	-35mm	I felt mellom midtre søyler, ca 35m fra brustart.

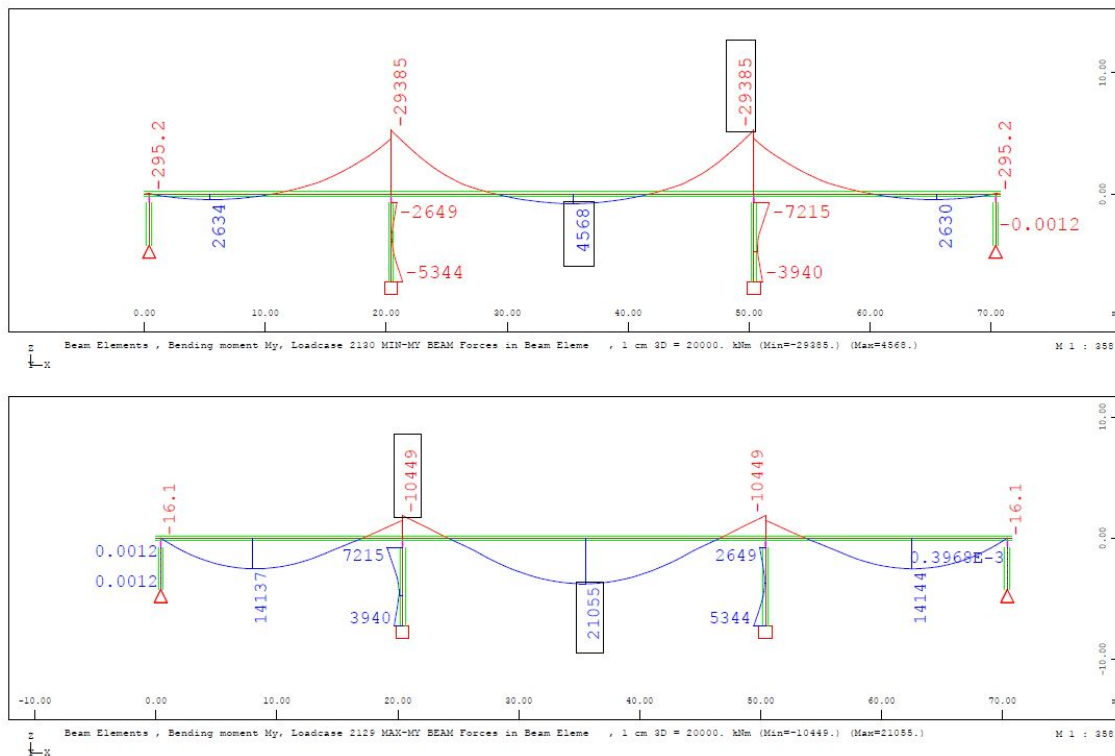
Deformasjon i lengderetning, samt vertikallast i området kontrolleres opp mot lagerbetingelser. Ensidiglager, gitt av spenntechnik ”*TOBE FR4 potlagere*”, kontrolleres for deformasjon og vertikallast [27]. Deformasjon på under 20mm i lengderetning utgjør ingen problem for lager-systemet da produkter for ensidiglager valgt fra spenntechnik har mulighet til opp mot 100mm deformasjon i lengderetning. Ensidiglager kan samtidig bestemmes for å ta opp en vertikallast på maksimalt 33400kN. Opptredende last ligger vesentlig under dette, slik at lagerløsning er gjennomførbar og modellens oppbygning følgelig realistisk.

Tverrsnittet dimensjoneres på bakgrunn av momentkreftene om y-aksen i bruddtilstand. Av hensyn til modelleringsvalg av et monolittisk system i midtre søyler, er det grunnlag til å redusere dimensjonerende støttemoment, jf EK2 punkt (5.3.2.2(3)). Momentreaksjonen kan reduseres etter likning 19, hvor $F_{ed.sup}$ er dimensjonerende opplagerreaksjon og t søyletykkelse. Av hensyn til at modelleringen brukes til sammenligning av resultat benytter man ikke reduserte verdier i videre beregning.

$$\Delta M_{ed} = F_{ed.sup} * t/8 \quad (19)$$

Beregning basert på T-tverrsnitt med konservativ metode angir mengde slakkarmering nødvendig for tverrsnittet. Videre benyttes momentverdiene for bestemmelse av spennarmering. Ved plassering av spennkabler har man to valg i Sofiplus; enten langs en linje, eller i forhold til en akse, da enten senteraksen til tverrsnittet eller egenprodusert akse. Spennarmeringen legges inn for å motvirke momentkreftene før en ny analyse blir foretatt. Med spennarmering reduseres momentene slik at behovet for slakkarmering gjennom konstruksjonen også reduseres. Videre kontrolleres tverrsnittet for samtidig virkning av moment og aksialkraft. Armeringsberegningene er utført etter Sørensens bok for betongkonstruksjoner [18]. Beregningene knyttet opp mot tverrsnittsdimensjoneringen er gitt i vedlegg I. og oppsummert i tabell 4.4

4. TRADISJONELL MODELLERING



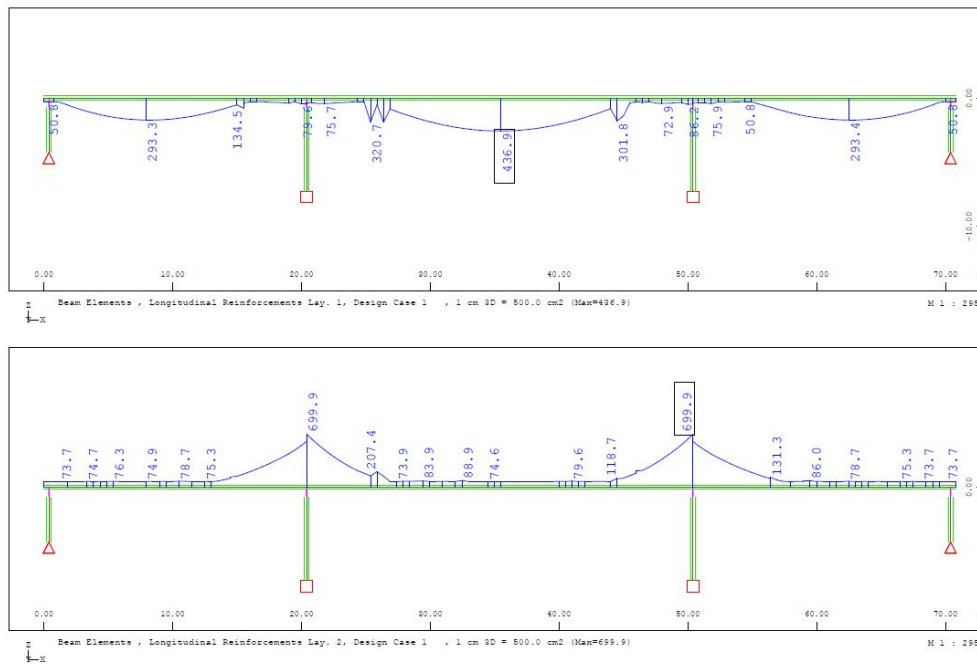
Figur 4.6: Maksimal og minimum momentdiagram beregnet fra Sofistik

Tabell 4.4: Slakkarmering uten spennarmering

	<i>Beregnet OK</i>	<i>Beregnet UK</i>	<i>Sofistik OK</i>	<i>Sofistik UK</i>
Slakkarmering	74533(69942) mm ²	50403(47586) mm ²	69990 mm ²	43690 mm ²

Tabell 4.4 indikerer mulig overdimensjonering ved håndsberegninger. Håndsberegningene er utført på en konservativ metode da høyden er redusert i henhold til fallhøyden av tverrsnittet. Dette gir vesentlig mindre momentarm og derfor større nødvendig armeringsmengde. Ved bruk av middelvei for høyden oppnår man mer nærliggende armering beregnet i Sofistik, gitt av parentesverdier i tabell 4.4. Verifisering av kvaliteten til beregnet armering i Sofistik utførtes en enkel test, med en fritt opplagt bjelke med rektangulært tverrsnitt. Resultatet gir lik håndsberegnet armering som resultat beregnet i Sofistik, se vedlegg G.

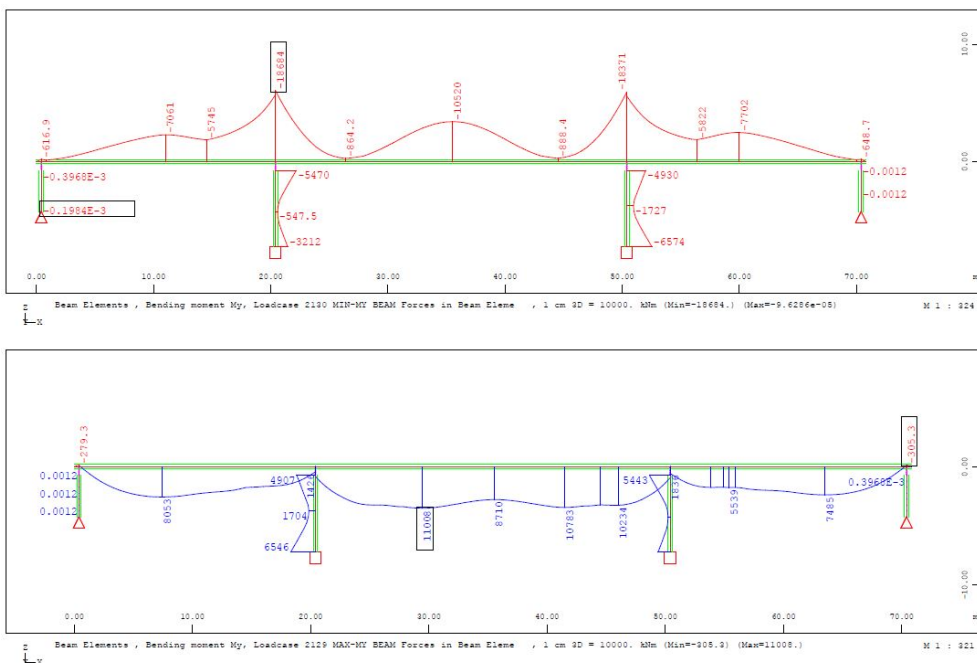
4. TRADISJONELL MODELLERING



Figur 4.7: Beregnet Armering fra Sofistik uten spennarmering

Beregninger tilsier nødvendig spennarmering på $A_p = 31500\text{mm}^2$, 14 spenntau av typen BBV ETA L15 blir tatt i bruk i Sofistik. Føringen blir etter best evne plassert for motvirking av momentforeløpet til permanente laster.

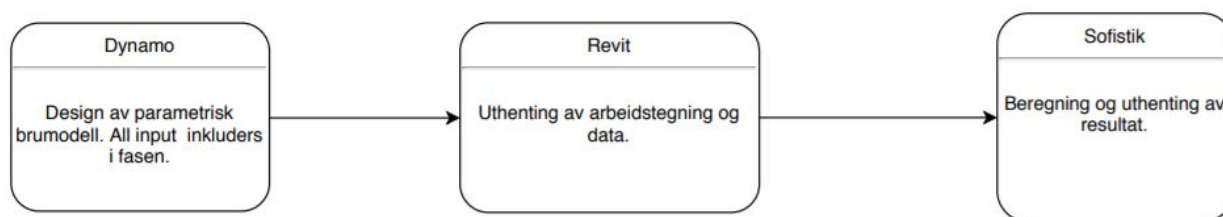
Spennarmeringsberegninger er primært anvendt som grunnlag i forbindelse med parametriske modellering.



Figur 4.8: Maks/min moment om y-aksen etter utplassering av spennarmering

5 Arbeidsflyt i programvare

Oppgavens mål er å benytte parametrisk design til å effektivisere arbeidsflyten mellom modellproduksjon og beregningsproduksjon i bru-prosjektering, se figur 5.1. Dette avhenger blant annet at overgangen mellom tegningsprogram og beregningsprogram fungerer godt. I denne sammenheng forstås arbeidsflyt for hvordan modellen lages og overføres mellom programmene som benyttes. En direkte overgang byr på flere utfordringer i programvaren. Av den grunn kartlegges alternative løsninger for arbeidsflyt mellom programmene. Alternativet med størst potensial utdypes og testes i påfølgende kapitler.

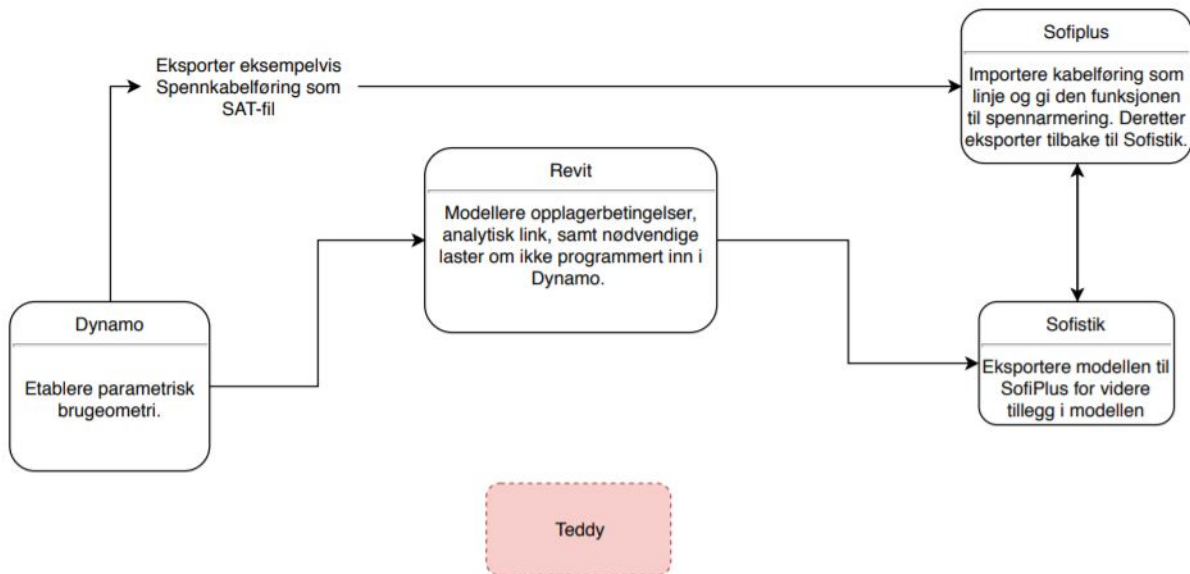


Figur 5.1: Ideell arbeidsflyt for effektivisering av enkel bru-prosjektering

Proessen med å etablere egnet metode for arbeidsflyt er bestemt gjennom testing av hvorvidt flyten eksisterer, kvalitet og brukervennlighet. Konsept om en parametrisk modell hvor både tegnings- og beregningsmodell blir etablert under samme prosess er grunnleggende mål for arbeidsflyten.

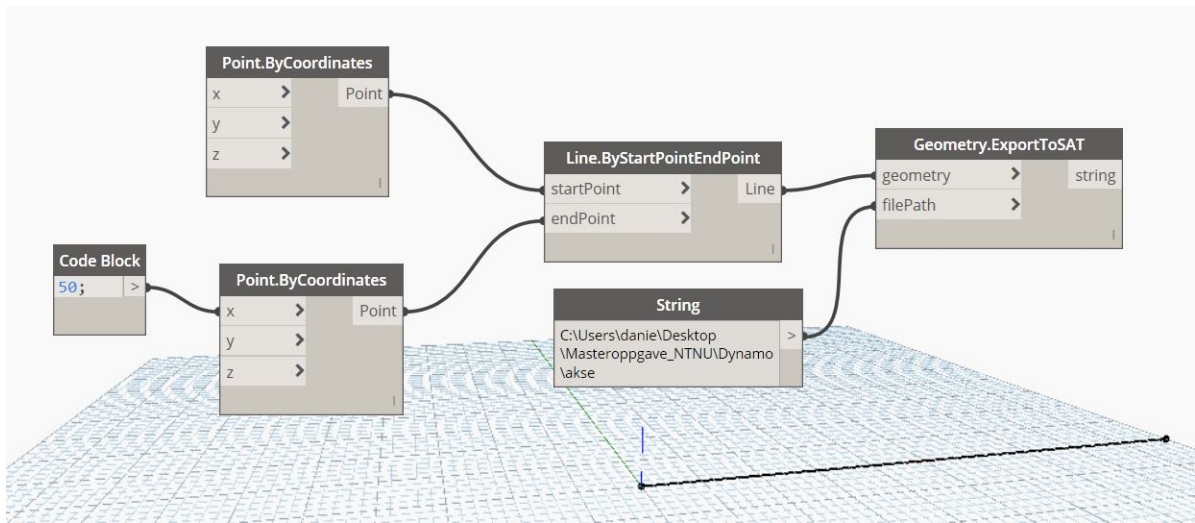
5.1 Flytalternativ I - Parametrisk modell med bruk av Sofiplus for tilleggsinformasjon

Metode én betrakter en direkte overgang mellom Revit og Sofistik, hvor Dynamo anvendes for å parametrisere brukeometrien og nødvendig spennkabelføring. Kabelføringen eller andre vesentlige elementer kan til senere tid importeres som linjer til tegningsprogrammet Sofiplus ved hjelp av SAT-filer. Filformatet SAT konverterer modeller til linjer og objekter og er et filformat støttet av flere tegningsprogrammer. Prosessen er illustrert i figur 5.2. Tanken bak metoden er en parametrisk brumodell hvor tilleggsopplysningen legges til i Sofiplus, slik at modellen blir komplett før analyse kjøres i Sofistik.



Figur 5.2: Flytdiagram I. Tilleggsinformasjon gjennom Sofiplus

Modellering ved bruk av Dynamo er forholdsvis lik som for alternativ IV. Hvordan modelleringen gjennom Dynamo gjøres beskrives i senere kapitler. Forskjellen i denne flytmetoden er anvendelsen av en geometrioverføring av diverse linjer og punkt fra Dynamo til Sofiplus. Dette gjøres ved å eksportere linjer og punkt til SAT-filer og deretter åpne SAT-filene i Sofiplus. Eksport til SAT-fil gjøres med noden *Geometry.ExportToSAT*, illustrert i figur 5.3. Geometri og filplassering legges inn.



Figur 5.3: Overføring av geometri til SAT-fil i Dynamo

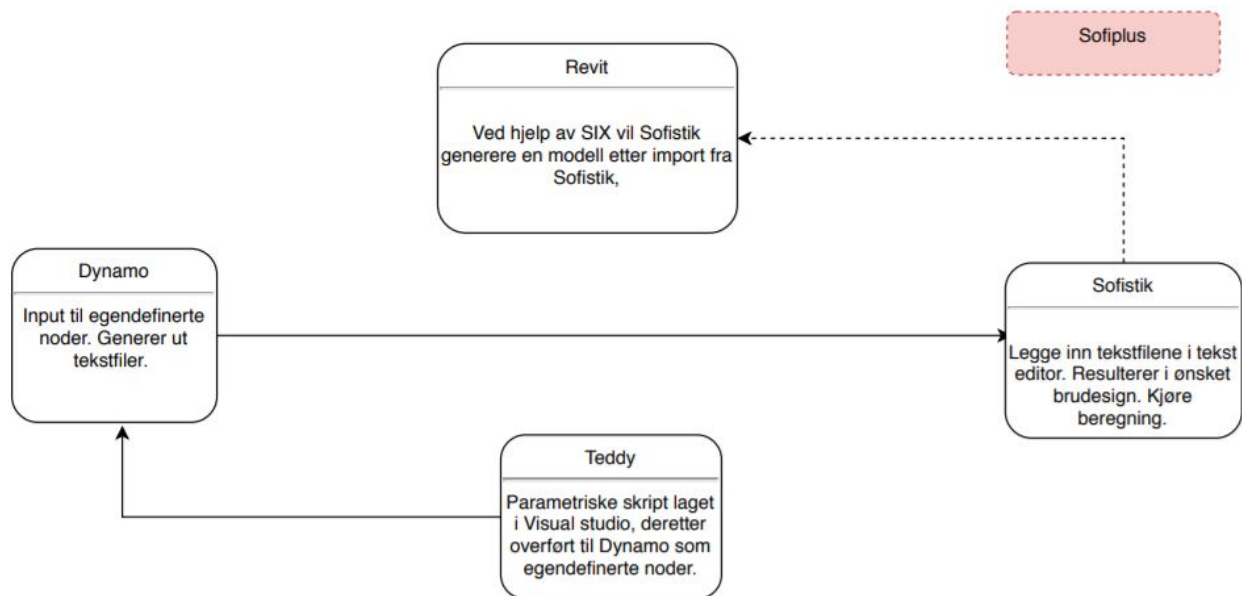
Geometrien i SAT-filen vil deretter kunne tildeles funksjon i Sofiplus. Et antatt egnet alternativ er å tildele en kurvefunksjon til spennarmering. Samtidig kan en slik operasjon alternativt benyttes til importering av komplett modell, for deretter tildele ulike elementer deres tenkte funksjon i Sofiplus.

Importering av kurve til Sofiplus med hensikt å tildeles spennarmering setter krav til flere operasjoner. Spennarmering i Sofiplus må følge en definert akse. Dette medfører at den importerte kurven må tildeles aksestatus før det deretter bestemmes at spennarmeringen skal følge akse. Eksentrisiteter må ha verdien null i alle punkt slik at føringen bestemmes direkte fra aksegeometrien. Dette på grunn av spennarmeringen vil vike fra kurven i punkter hvor verdien ikke er null. Dersom en «komplett» tegningsmodell med flere linjer fra Dynamo overføres til å tegne brukeometri, oppstår problemet at geometrien overføres som et enkelt objekt. Dette gjør det vanskelig å tildele en funksjon for hver linje.

Ettersom arbeidsflyten består av to designprogram oppstår det flere problemer. Enhver endring i tegningsmodellen vil ikke gi automatisk endring i Sofiplus. Dette medfører at parallelt med enhver endring i Revitmodellen, må de også gjøres manuelt i Sofiplus. En ny importering må forekomme og følgelig nye operasjoner opp mot analysen er nødvendig. I tillegg til manglende arbeidsflyt ved endring i modell, er endringer i flere stadier lite hensiktsmessig. Et av målene var at tyngden av modellering foregår på samme plass. Bruk av to designprogram byr samtidig på generell flytproblemer mellom programvarene. Etter at modell er overført til Sofistik fra Revit og videre importert til Sofiplus forekommer det hyppige feilmeldinger dersom begge tegningsprogrammene er åpne samtidig. Løsningen krever at man konsekvent kun har et designprogram åpent, eller hyppig restarting av lisensen.

5.2 Flytalternativ II - Parametrisk modell gjennom Teddyskript

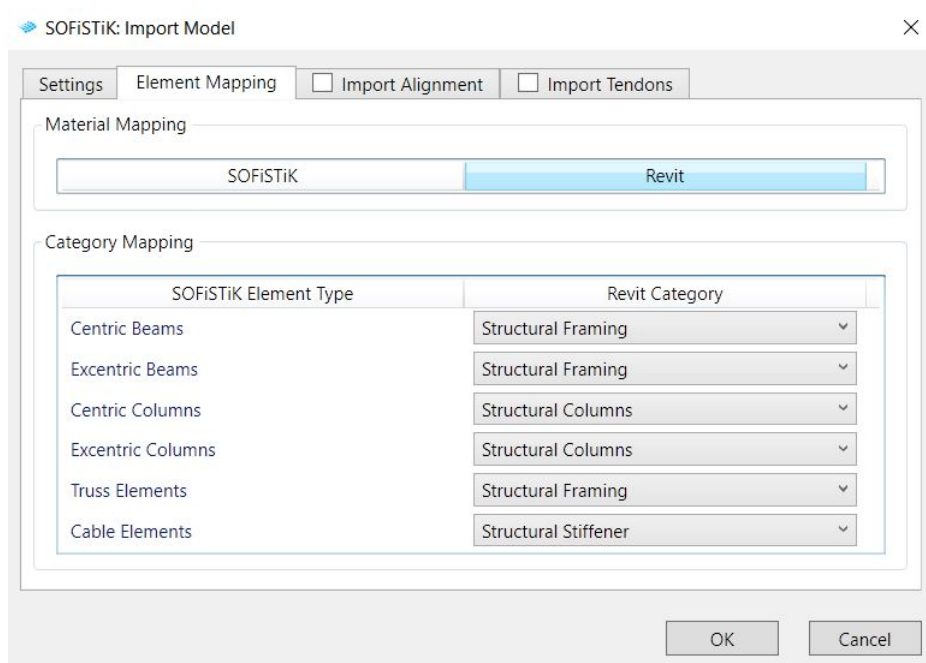
Begrensninger knyttet til overføringsevnen i Revit ga opphav til en metode hvor direkte overføring mellom Revit og Sofistik uteble. For å tilfredsstille målsetning om en designdatabase i Dynamo egnet for parametrisk manipulasjon er det naturlig å rette fokuset mot bruk av Teddy-verktøyet. Tanken er at egne Teddyscript designes og etter behov parametriseres. Teddyscriptene vil dermed være grunnlaget for etablering av geometri og laster i denne arbeidsflyten. Beregningsmodellen etableres derfor først i Sofistik før den overføres til tegningsmodell i Revit, illustrert i figur 5.4. Dette er til forskjell fra arbeidsflyt I og IV, hvor tegningsmodell overføres til beregningsmodell.



Figur 5.4: Flytdiagram II. Modellering gjennom bruk av Teddy

For å parametrisere Teddyscriptene benyttes *Visual Studio* til å designe egne noder til Dynamo. Typiske Teddyscript hvor parametriseringen gjøres er for geometri og laster. Nodene bygges slik at output-porten, det vil si resultatet, i noden tilsvarer tekstdokumentet i teddyformat. Outputen eksporteres deretter til en tekstfil og ved enkel «klipp og lim» kan tekstfilene legges inn i Sofistik ved bruk av «tekstfileditor». På denne måten etableres modellen i Sofistik. Metoden utdypes ytterligere i kapittel 6 for blant annet temperaturlast.

Overføringen mellom Sofistik og Revit gjøres gjennom Sofistik sitt tilleggsprogram *Sofistik Infrastructure*, omtalt SIX [28]. Et krav til overføringen er at modellen er bjelke- eller kabelbasert, se figur 5.5. Dette medfører at design med eksempelvis skallelement ikke lar seg overføres. Det er ingenting som antyder at overføring fra Sofistik til Revit ved bruk av SIX gir en bedre overgang eller flyt mellom programvarene, enn den motsatte overgangen.



Figur 5.5: Overførbare elementer Sofistik-Revit

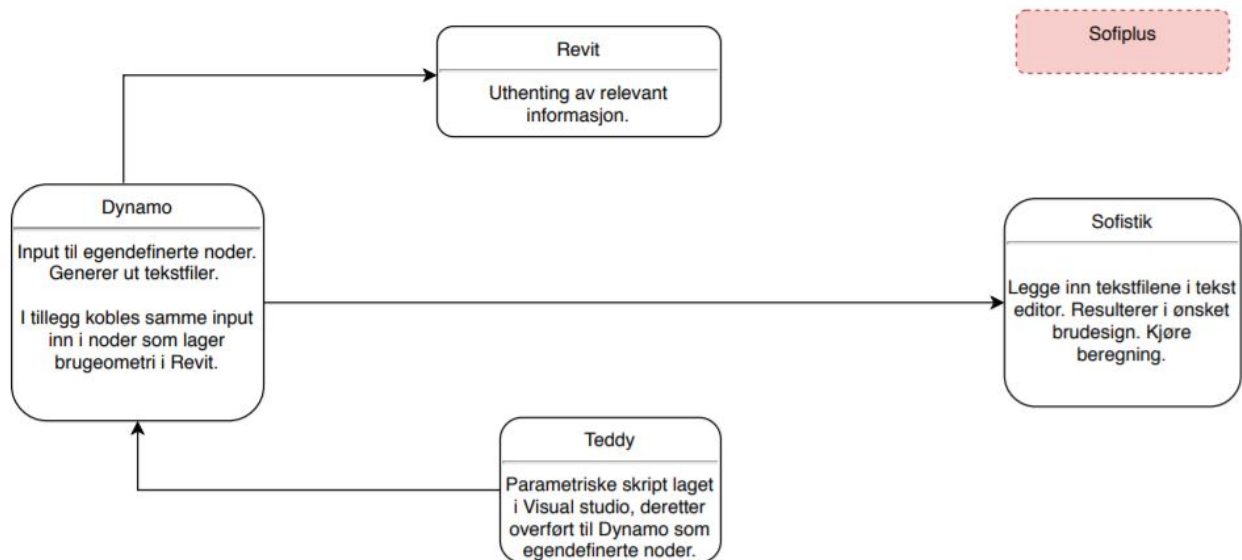
Teddyfiler gir mange muligheter til manipulering av design, hvor Revit og Sofiplus forøvrig har sine begrensninger. Likevel gir Teddy ingen visualisering under selve prosessen, det vil si at for å få en slik egendefinert Teddyfil til å fungere trengs gode kunnskaper innenfor dette området. Benyttelse av slike noder gir heller ingen visualisering i Dynamo, slik at denne metoden utelukkende generer tekstfiler ut i fra gitte inputverdier.

Videre luker bruken av Teddy vekk begrensninger i overgangen Revit til Sofistik. Men ved design utelukkende med bruk av Teddy trengs et stort antall noder hvor flere av nodene trenger store mengder inputverdier for å oppnå en god parametrisering. Metoden vil derfor være uoversiktlig, samt bære preg av dårlig brukervennlighet. Å skape noder med bedre utgangspunkt enn eksisterende teddyfiler er krevende. Foreligger det god forståelse i bruk av teddyfiler, vil endringer direkte i disse trolig være like effektivt. Nødvendigheten av parametrisk design i Dynamo blir tilnærmet ikke-eksisterende, men bistår mer som et databaseverktøy for input verdier med mål om større brukervennlighet enn tilhørende teddyfiler.

Metoden hvor visualisering er ikke-eksisterende under prosessen, samt stiller krav til programmeringssevne for generering av brukervennlige noder virket mindre egnet. Man valgte derfor å ikke ta metoden videre.

5.3 Flytalternativ III - Dynamo som database for separate modeller

Da forgående alternativ manglet visualisering i designfasen, gir denne metoden en aktuell løsning på nettopp dette. Konseptet er tilnærmet identisk med alternativ II. Overføringen fra design til analytisk modell skjer ved hjelp av Teddyfiler generert på lik linje som tidligere. Dynamo anvendes som et databaseverktøy for inputverdier, men forskjellig fra tidligere deles Dynamo inn i to deler. Del én der inputverdien legges til gir ut teddybasert tekstfiler. Del to etablerer den visuelle geometrien med direkte tilknytning til Revit. Opprettelse av en slik parametrisert modell krever god koordinering ved at verdier i parametrisert teddyfiler kan knyttes til tilhørende lik mulighet i Revit. Ambisjonen vil derav være en visualisering som overstemmer med hva teddyfilen generere i analytisk modell. En slik koordinering trenger forøvrig kun være forhold knyttet til brugeometri. Metoden ble derimot ikke tatt med videre grunnet mengde design ved bruk av teddyfil ansees som krevende å få til. Likevel nevnes metoden som et alternativ i hvordan brudesign kan oppbygges, der man designer en modell med opphav i Dynamo hvor resultatet er både tegnings- og beregningsmodell. Fordelen med en slik metode er at begge modellene er relativt uavhengige, samt at teddykripting tolkes perfekt av Sofistik slik at begrensninger i Revit-Sofistik overgangen uteblir. Dette kan samtidig være en ulempe i og med at man i teorien jobber med to forskjellige modeller. Metoden illustreres i figur 5.6.

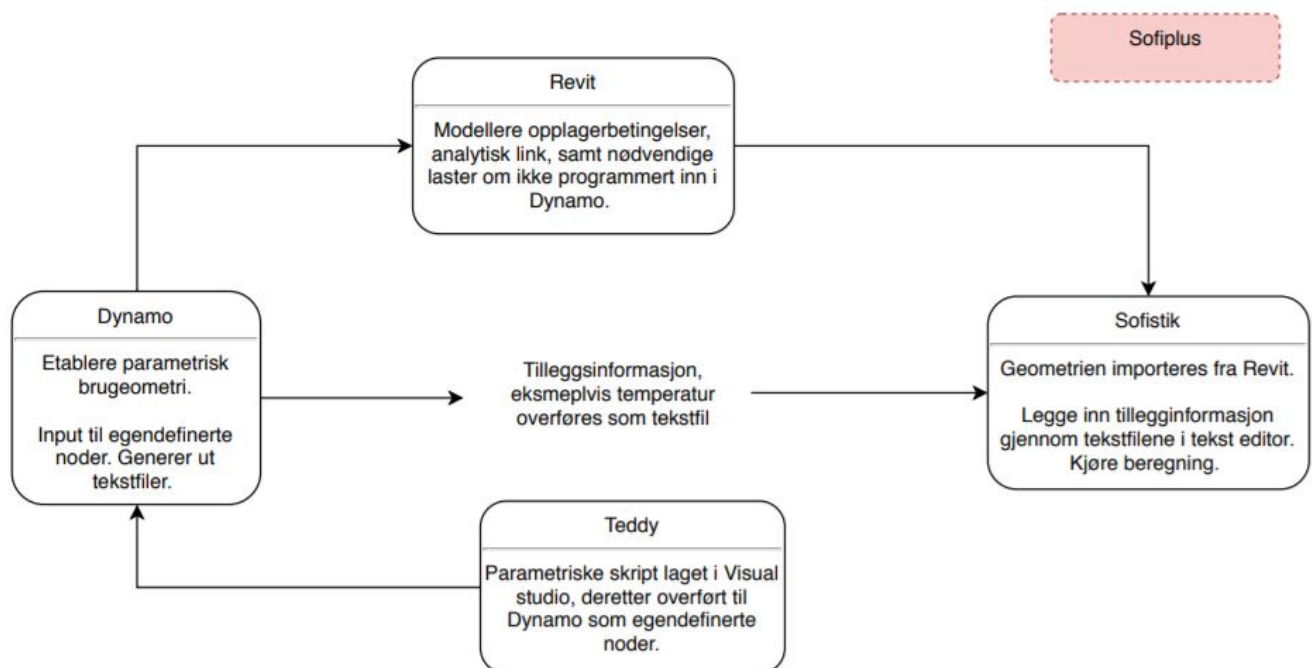


Figur 5.6: Flytdiagram III. Dynamo som database for to separate modeller

5.4 Flytalternativ IV - Parametrisk modell med bruk av Teddy for tilleggsinformasjon

Målsetningen om en egnet arbeidsflyt basert på brukervennlighet og kvalitet resulterte i metoden illustrert i figur 5.7. Det er praktisk å utnytte den direkte overføringen mellom tegnings- og beregningsmodellen, slik at Dynamo kan benyttes til å etablere en parametrisk modell. Her opprettes en dynamisk modell, med direkte visualisering.

Arbeidsflyten bygges rundt dynamoscript der geometri og laster tilknyttet Revit etableres. Modellen i Revit overføres deretter til Sofistik for videre bearbeiding av analysen. Aspekter med bruken der det ikke er gjennomførbart å ta i bruk koblingen mellom Dynamo og Revit, anvendes parametriske teddyscript implementert i Dynamo.



Figur 5.7: Flytdiagram IV. Parametrisk modellering i Dynamo, hvor Teddy benyttes for tilleggsinformasjon

Målet i denne metoden er at største parten av inputen foregår i Dynamo. Grunnen til dette er for at man skal slippe å gjøre endringer underveis og på flere steder/programmer. Endringer som gjøres i Dynamo vil dermed fanges opp i både tegningsmodellen (Revit) og beregningsmodellen (Sofistikk).

6 Parametrisk modell

Alternativ fire vurderes til å ha størst potensial vedrørende arbeidsflyt. Det er derfor naturlig å ta dette alternativet videre og utforske, utdype og teste hvordan det fungerer. Utgangspunktet for metoden er å kunne lage en parametrisk modell nøyaktig lik brumodellen designet på tradisjonelt vist. Videre skal bruene enkelt kunne endres ved å bestemme parametere som bygges inn i den parametriske modellen. Deretter utføres parametriske studier i påfølgende kapittel hvor metoden testes for forskjellige utforminger og design.

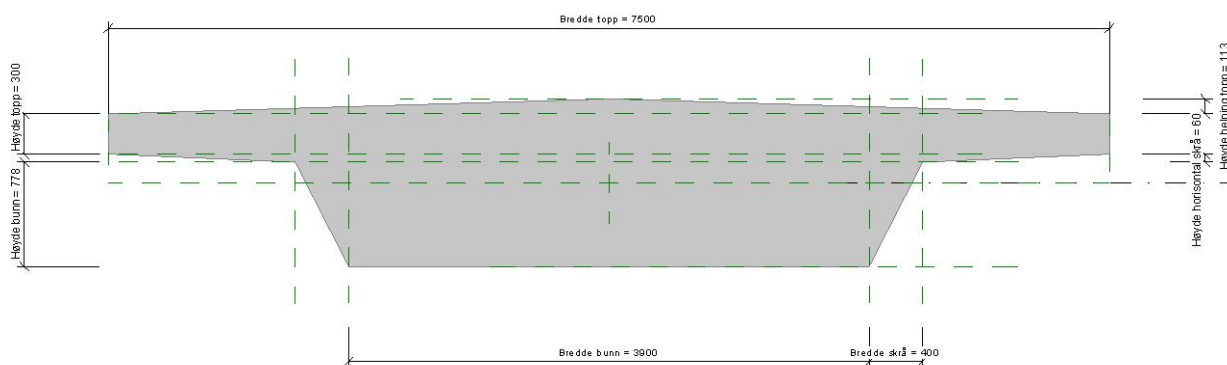
En detaljert beskrivelse med nummerert rekkefølge er laget for arbeidsflyten for å vise hvert steg i prosessen med tekst og bilde. Dette gir en bedre forståelse av oppbygningen til modell og hvordan arbeidsflyten fungerer. Parametrisk design er et relativt nytt verktøy og det er dermed viktig å beskrive stegene nærmere for å forstå den grunnleggende prosessen. Arbeidsprosessen viser i tillegg hvilke steg de forskjellige inputene må utføres i. Personer uten erfaring med brukte programvarer i arbeidsflyten vil kunne følge fremgangsmåten uten store problemer.

6.1 Parametrisk modellering

6.1.1 Dynamo

Gjennom Dynamo foreligger det mange måter å bygge opp script til å løse de problem brukeren har. I denne oppgaven er det gjort på én måte, men det vil finnes andre alternative løsninger for å løse samme problemet i Dynamo.

For å lage et parametrisk brutverrsnitt benyttes Revitfamilien for structural framing. På denne måten vil familien lage analytisk modell i Revit. Parametere kan enkelt legges inn i familien og endres i Dynamo.



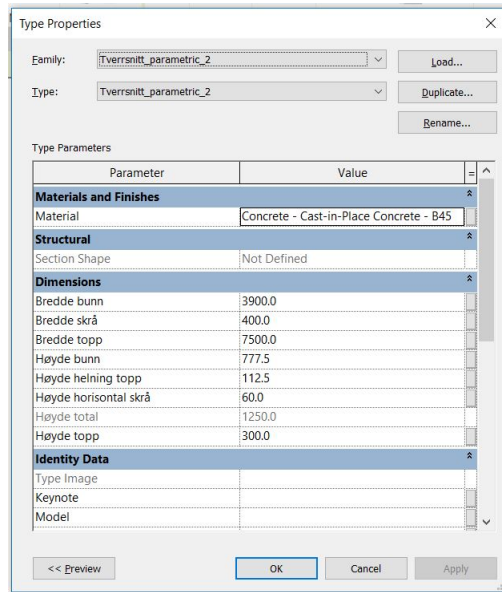
Figur 6.1: Parametere brutverrsnitt

Parametriseringen av tverrsnittet gjøres ved å legge dimensjonene inn som parametere med tilhørende navn. Siden tverrsnittene ofte er symmetrisk om senteraksen legges det også inn faste

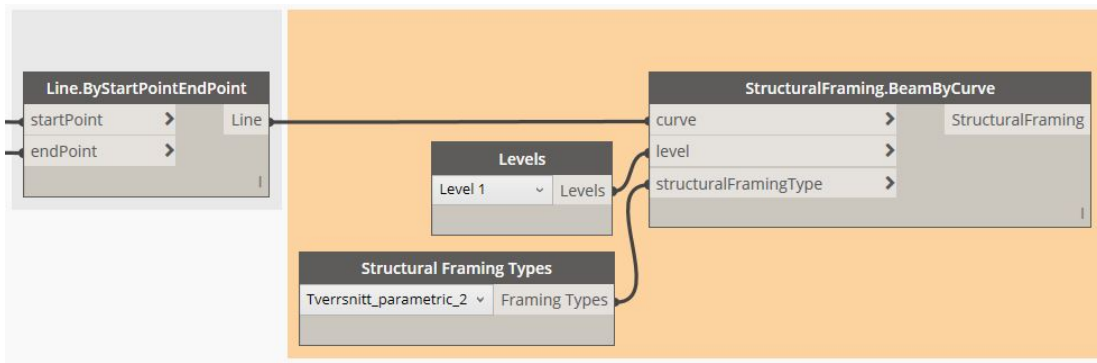
6. PARAMETRISK MODELL

begrensninger slik at symmetri opprettholdes. Færre parametere vil dermed være nødvendig for å lage tverrsnittet, da tverrsnittet alltid vil forbli symmetrisk om senteraksen. Samme prinsippet brukes for søylene, men her brukes en structural column familie. Nærmere beskrivelse av de parametriserte dimensjonene blir ikke gjort.

Familiene for brutverrsnitt og søyletverrsnitt lastes inn i Revit-prosjektet slik at Dynamo kan bruke dem. Dynamo finner kun familier og element som er lastet inn i prosjektet Dynamo opererer på.



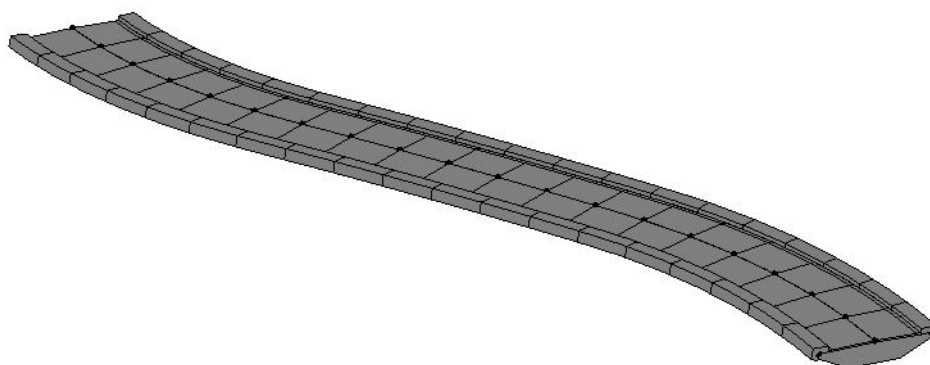
Figur 6.2: Parametere i Revit



Figur 6.3: Structural framing-node med tilhørende input

For at Revit skal kunne generere en modell i Sofistik er det avhengig av at modellen i Revit er en analytisk modell. Gjennom Dynamo er dette bare oppnåelig ved bruk av en *Structural Framing.BeamByCurve*-node (figur 6.3) eller *Structural Framing.ColumnByCurve*-node. Dette har begrensninger som medfører at ikke alle typer veigeometrier tilfredsstilles i Revit. Det største problemet er dersom veilinjen ikke er rett. Det er kun gjennomførbart å lage geometri med krumning i et plan av gangen med disse nodene. En kan dermed ikke få en bru som både heller

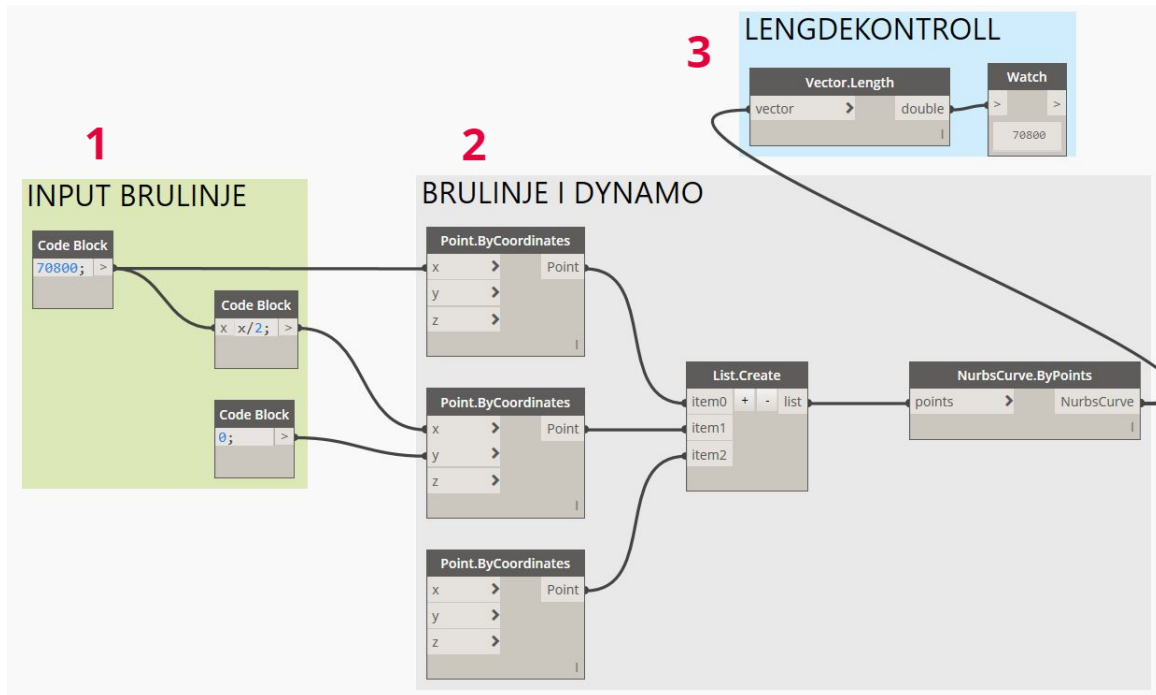
ut av planet og svinger i planet samtidig. Det er fullt mulig å lage en bru i Dynamo som gjør dette uten en analytisk modell, men da vil den ikke kunne overføres til Sofistik, pga. manglende analytisk modell. En løsning på dette er dersom man forenkler og deler veilinjen opp i mange mindre, rette linjer. På den måten kan structural framing-familien fange opp små endringer i veilinjen. Forenklingen der krummet veilinje erstattes med korte, rette linjer resulterer i en tegningsmodell oppdelt av mange rette profiler. Dette medfører flere «åpninger» mellom profilene der veilinjen har stor krumning. Kvaliteten til tegningsmodellen svekkes derfor på bakgrunn av denne løsningen.



Figur 6.4: Bru i Revit laget med adaptive family

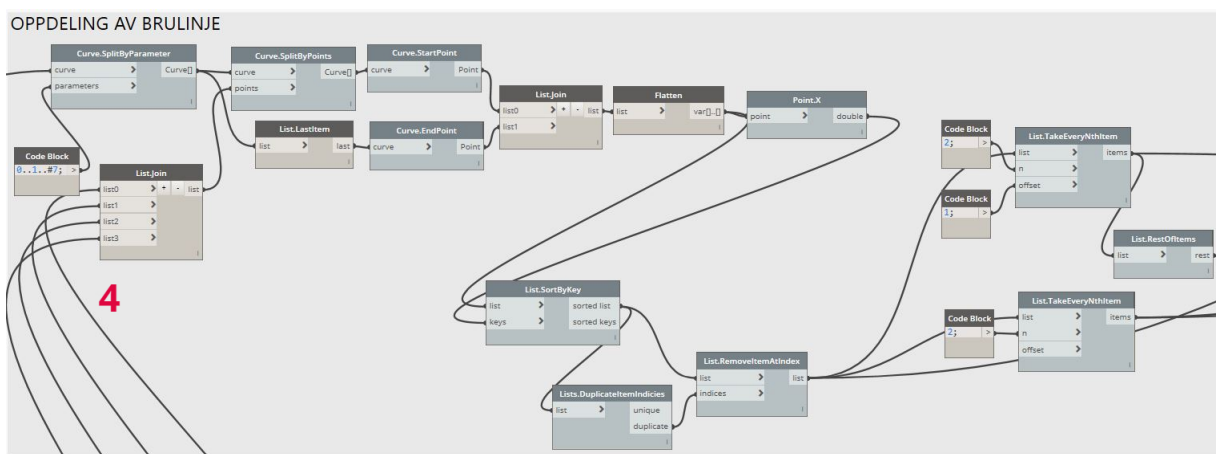
Inndeling av *mesh* i Revit til Sofistik vil være uavhengig av hvor mange linjer veilinjen deles opp i, da dette bestemmes i Revit og ikke i Dynamo. Videre er det ikke mulig for tverrsnittet å endre geometri langs veilinjen. Tverrsnittet vil dermed være låst til en geometri langs hele veilinjen. I motsetning til bruken av *adaptiv family* hvor tverrsnittet forholder seg til flere punkter og gir mulighet til lokal rotering. Med det oppnår man visuelt bedre tegninger. Problemet med adaptiv familie er at det ikke lages en analytisk modell som kan overføres til Sofistik.

Skript for bruoverbygning



Figur 6.5: Brulinje i Dynamo

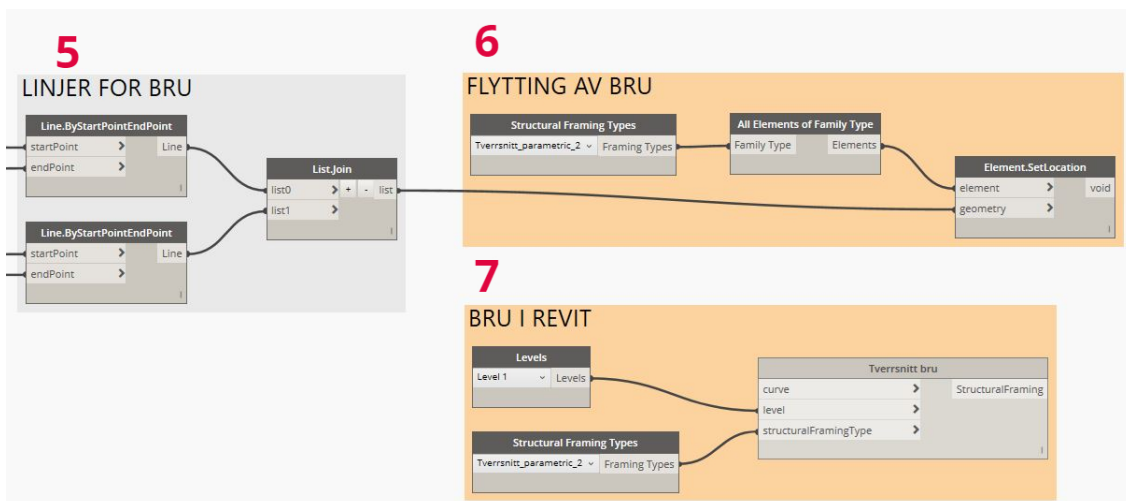
1. Input for brulinje. Her er inputen lagt inn i flere *Code Block*-noder
2. Brulinjen lages med f.eks. tre punkter i *Point.ByCoordinates* som kobles inn i *NurbsCurve.ByPoints*. Tilsvarende etablering kan forekomme med utallige punkter for å skape en veikurve.
3. Kontroll av lengden på bruens gjøres ved å finne lengden av kurven med *Vector.Length*



Figur 6.6: Oppdeling av brulinje

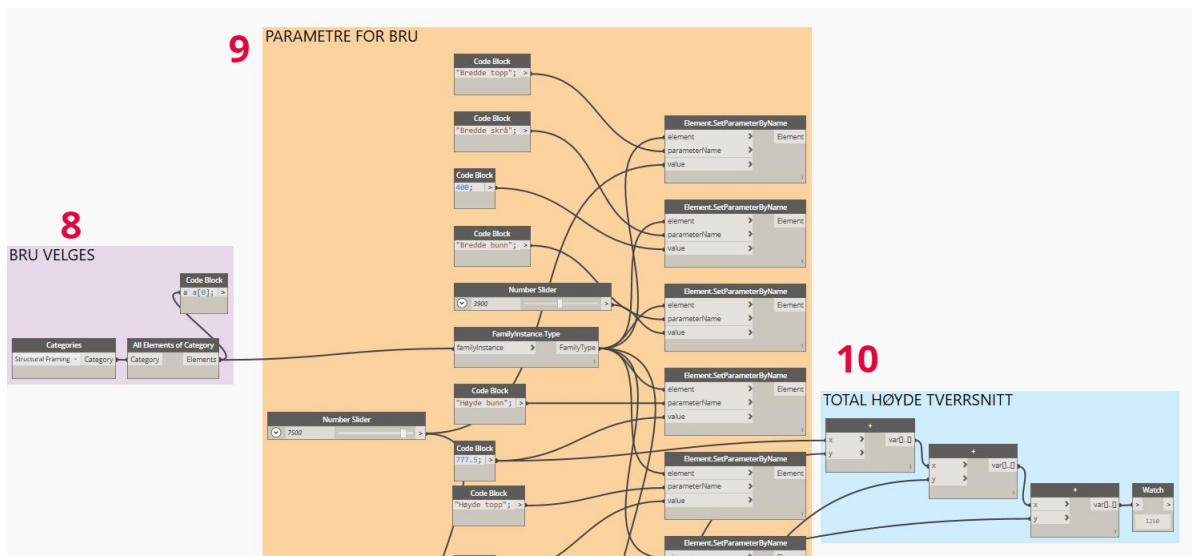
4. Veilinen deles opp i flere elementer slik at bruens kan håndtere kurvet veilinje bedre.

Samtidig er oppdeling nødvendig for å tilrettelegge for analytiske link mellom søylene og brudekket. Veilinen legges inn i *Curve.SplitByParameter* sammen med antall elementer man vil ha for å dele opp bruene. Samtidig splittes bruene der hvor den analytiske linken tiltenkes. Dette gjøres ved at punktene for søyleplassering legges inn i *Curve.SplitByPoints*. Videre sorteres alle punktene bruene er delt inn i slik at linjer kan lages mellom dem. Sorteringen kan foregå på ulike måter. Ved rettlinjete bru er det anvendt sortering basert på x-koordinater, men sortering basert på den skapte *Nurbs.Curve.ByPoints* kan være like hensiktsmessig.



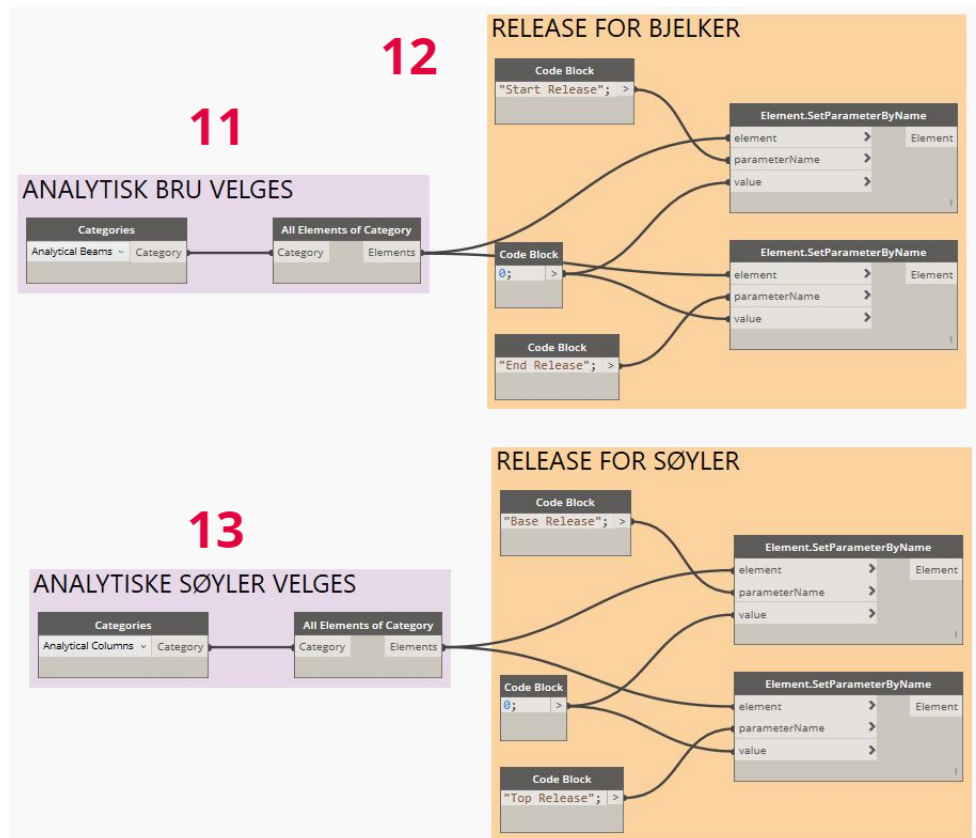
Figur 6.7: Overføring til Revit

5. Linjer lages fra punktene med *Line.ByStartPointEndPoint*.
6. Dersom det allerede er bruprofil i Revitprosjektet er det kun nødvendig å bruke den lagde Dynamogeometrien til å flytte på dem. Dette gjøres ved å hente bruene fra Revit og flytte dem med *Element.SetLocation*.
7. Dersom det ikke allerede er bruelement i Revit, brukes *StructuralFraming.BeamByCurve* til å lage elementene med ønsket linje, nivå i Revit og tverrsnittstype. Årsaken til at man heller vil flytte elementer, er at *StructuralFraming.BeamByCurve* kan i noen tilfeller lage elementer oppå andre eksisterende elementer. Det er derfor lettere å flytte på elementer man allerede har etablert. At Dynamo ikke er satt til å kjøre automatisk, men heller at man kjører scriptet manuelt er dermed relativt viktig.



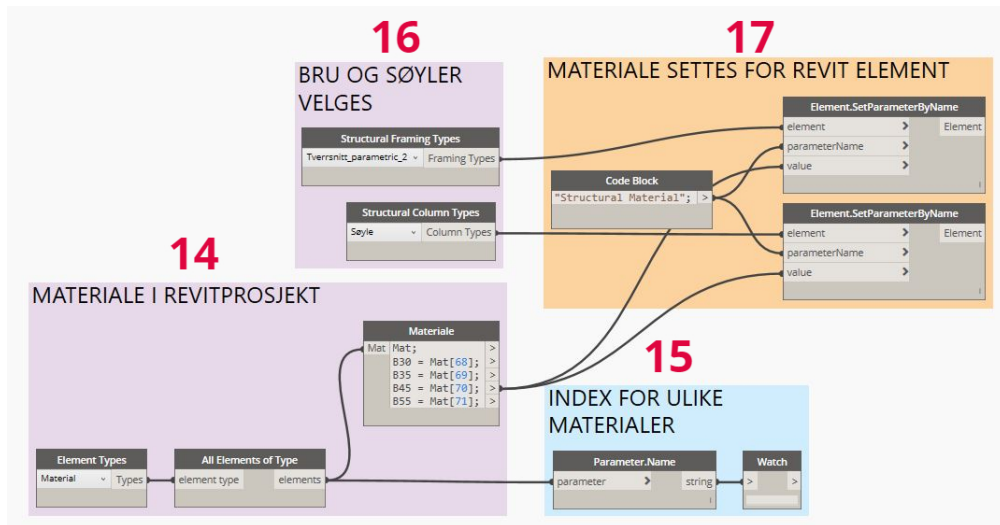
Figur 6.8: Bruparametere i Dynamo

8. For å endre på bruprofilet hentes komponenten fra Revit ved å først finne kategorien med *Categories* og deretter hente alle elementer i kategorien med *All Elements of Category*.
9. Videre setter man verdi på parameterne til bruelementene ved å bruke *Element.SetParameterByName*. Alle parameterne i et bruelement kan enkelt og effektivt endres i Dynamo på denne måten.
10. For å kontrollere tverrsnittshøyden adderes alle høydeparametere for å få total høyde. Denne kontrollen er nødvendig på bakgrunn av at det parametriske tverrsnittet er utformet slik at det ikke er én parameter for total høyde, men heller individuelle høydeparametere for deler av tverrsnittet.



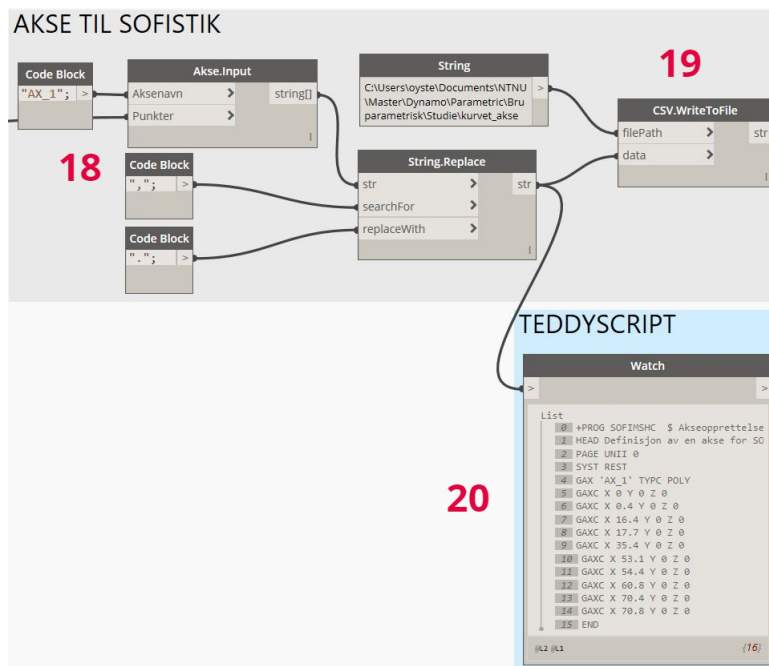
Figur 6.9: Fastholding i Dynamo

11. Fastholdingsparametere kan enten være fastholdt eller fri. I Revit bestemmes fastholdingen ved at den enten er slått på (fri) eller av (fastholdt). For å bestemme fastholdingen gjennom Dynamo benyttes noden *Boolean*, hvor true er fri og false er fastholdt. I Scriptet er det benyttet verdiene 1 og 0 for henholdsvis true og false, da dette er det samme i programmeringsspråket.
12. Verdi og parameternavn legges inn i *Element.SetParameterByName* for å bestemme fastholding endene av elementene.
13. Fastholding for søylene bestemmes på tilsvarende måte.



Figur 6.10: Materiale i Dynamo

14. For å definere materiale til brudekket og søylene må alle materialer tilgjengelig hentes fra Revit. Disse er definert som element og finnes ved *Element Types* og *All Elements of Type*. Når materiale hentes fra Revit på denne måten er det materialnummeret Dynamo finner og ikke navnet på materialet.
15. Måten man finner riktig materiale er ved å finne navnet materialenummeret er koblet til. Dette gjøres med noden *Parameter.Name* og en *Watch*-node for å se outputen. Riktig materiale med index (nummer i listen) kan nå enkelt finnes.
16. Elementtypen materialet skal defineres til, velges med *Structural Framing Types* og *Structural Column Types* for henholdsvis bru og søyler.
17. Tar man igjen i bruk *Element.SetParameterByName* og indexen for materialet, kan parameteren *Structural Material* bestemmes.



Figur 6.11: Akse til bruk i Sofistik

18. Etersom Revit inneholder mangler i blant annet lasttilfeller som temperatur- og trafikklast samt spennarmering, bestemte man å ta i bruk egendefinerte noder i Dynamo. Her vil de egendefinerte nodene genererer tekstfiler i Teddyformat. Alternativt kan tillegg knyttet til modellen modelleres inn ved bruk av Teddyscript direkte i Sofistik. Da målsetningen er en arbeidsflyt hvor design skjer i et ledd, blir det naturlig å designe egne noder, slik at dette kan realiseres.

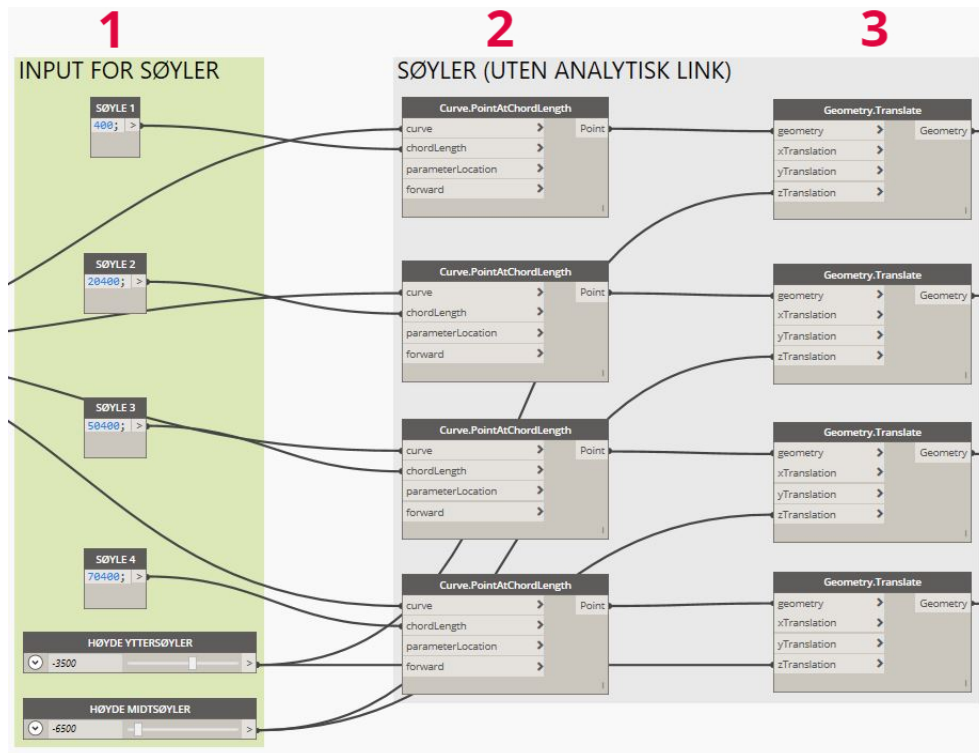
For å oppnå korrekt design trengs kunnskap om Teddy. Teddyprogramvaren er bestående av flere gode eksempler, hvor nyttig informasjon kan uthentes. I tillegg inneholder teddytekstformat en brukerveiledning der kodene kommer frem. Dette hjelpemidlet er praktisk i opprettelsen av slike teddygeneraliserte noder.

For å designe en node anvendes Visual studios. Her programmers noder etter ønsket input av variabler, samt fremstilling av output. Vist i dette punktet er aksenode. Inputen vil være aksnavn, samt liste med punkter. Aksnavnet er en *string*, dermed må inputen være i likt format. For å tilfredsstille krav i Sofistik, kan aksnavnet ikke være større enn fire bokstaver/tall. Listen med punkter må være i «stigende» rekkefølge med tanke på at aksen blir en linje mellom dem. Ved rett bru er det kun nødvendig å benytte start og endepunkt i akseopprettelsen. Aksenoden, samt andre øvrige designede noder er gitt i vedlegg I.

19. *CSV.WriteToFile* har funksjon å lagre tekstfilen i ønsket plassering.
 20. Anvendes en *Watch*-node, kan innholdet i tekstfilen kontrolleres.

Et alternativ til nodebasert akse er bruken av gridlines. Gridlines i Revit resulterer i akseystem i Sofistik.

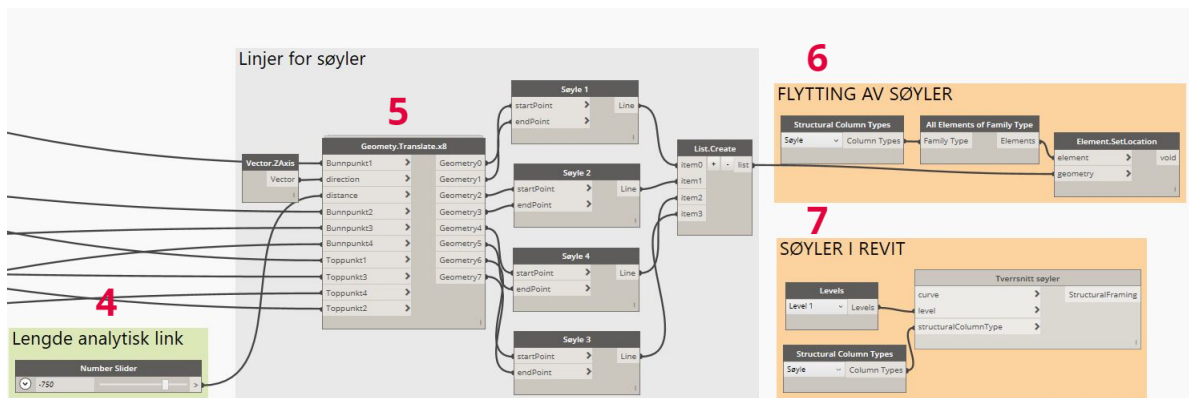
Script for søyler



Figur 6.12: Søyler i Dynamo

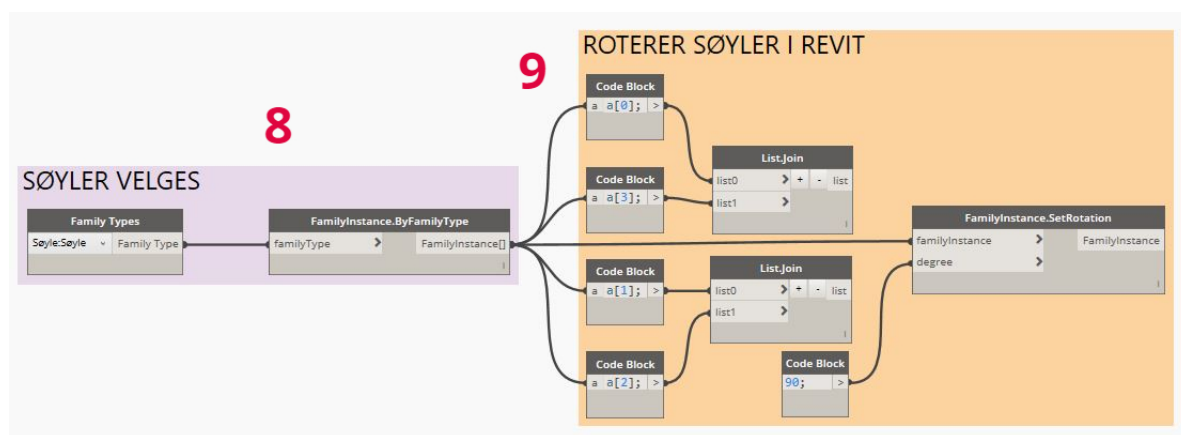
1. Avstander fra veilinjestart og høyder for søylene defineres. Her ved hjelp av *Code block* og *Number Slider*.
2. *Curve.PointAtChordLength* lager punkter på veilinjens ut fra definerte avstander i foregående punkt. Dette vil være søylenes toppunkt.
3. Bunnpunkt for søylene opprettes ved å kopiere toppunkt med *Geometry.Translate*. Høyden legges inn i input-porten for *zTranslation* i noden.

Presisert innledningsvis i kapittelet kan utforming av script foregå på ulike metoder. Dette gjelder naturligvis ved oppføringen av søylene. Den valgte metoden bygges søylene opp basert på veilinjens. Dette medfører få tydelig inputverdier gitt i punkt 1 og 4, men man mister litt spillerom da de ikke er fullstendig uavhengige. Dersom søylegeometrien bygges rundt *Points.By.Coordinate* vil søylene være uavhengige og dermed forholdsvis mer fleksible dersom dette kreves.



Figur 6.13: Søyler i Dynamo 2

4. Lengden for den analytiske linken legges inn med en *Number Slider*. Denne gir en åpning i modellen hvor den analytiske linken legges manuelt til i Revit.
5. For å lage søylen med start under den analytiske linken, flyttes topp- og bunnpunkt fra forrige script (figur 6.12) i z-retning med tilsvarende avstand som lengden til den analytiske linken. Dette gjøres med noden *Geometry.Translate* og i figur 6.13 er åtte slike noder satt sammen til én, slik at alle åtte punkt legges inn i en node, sammen med lengden av analytisk link og retningen de skal flyttes. Med disse punktene lages så fire linjer med *Line.ByStartPointEndpoint*.
6. Dersom det allerede er søyler i Revitprosjektet er det kun nødvendig å bruke den lagde Dynamogeometrien til å flytte på dem. Dette gjøres ved å hente søylene fra Revit og flytte dem med *Element.SetLocation*.
7. Dersom det ikke allerede er søyler i Revit, brukes Dynamogeometrien til å lage dem med noden *StructuralFraming.ColumnByCurve*. Nivå og tverrsnittsgeometri legges inn i samme node for å lage elementet i Revit.



Figur 6.14: Rotasjon av søyler i Dynamo

8. Når søylene lages i Revit kan det forekomme at de er rotert feil vei, eller at lokalt akse-systemet ikke er plassert riktig i forhold til hva man vil ha i Sofistik. Rotering av søyler i

Dynamo gjøres derfor enkelt ved noden *Family Types* for å finne søyletypen, og deretter alle familietilfellene av ønsket søyle med *FamilyInstance.ByFamilyType*.

9. Rotasjonene settes så med *FamilyInstance.SetRotation* og ønsket rotasjon i grader. Denne noden roterer familien kun om lokal x-akse.

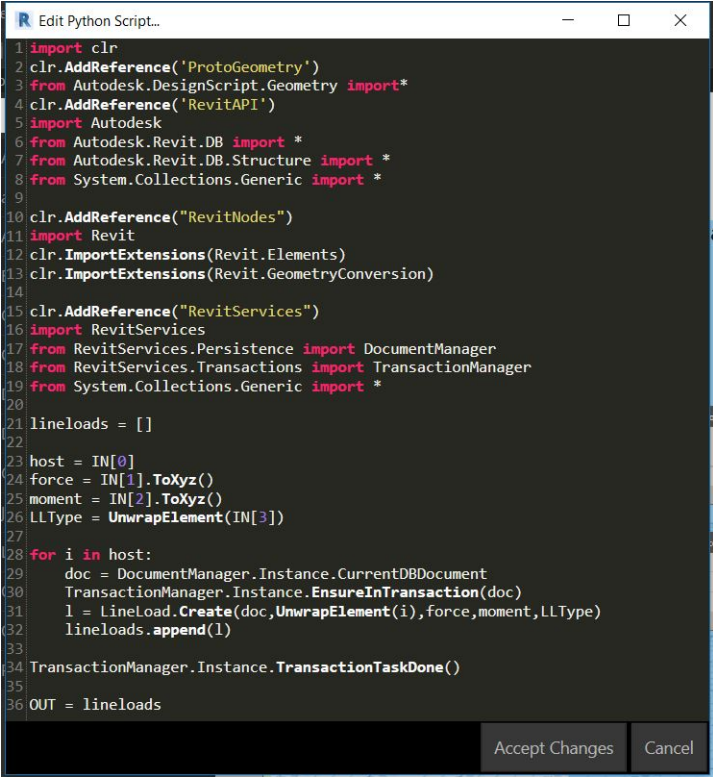
Parametere for søyler settes på lik måte som for bruen og vises derfor ikke. Her inngår tverrsnittsgeometri, materiale og fastholding.

Laster på brukonstruksjonen

Skript for laster er laget i en egen fil, for å unngå at mange krevende operasjoner kjøres samtidig i Dynamo. Det er i tillegg mer oversiktlig å arbeide med én del av modellen av gangen.

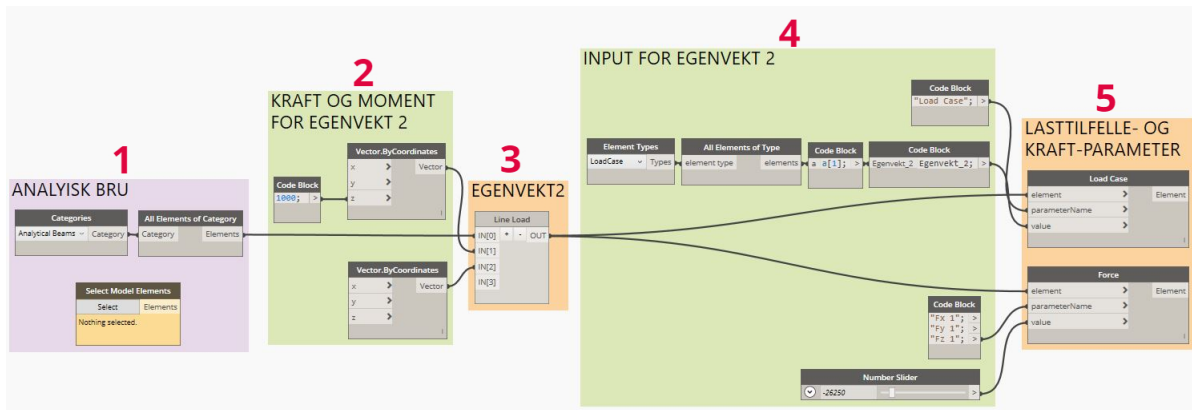
Dynamoskriptet inneholder et python skript for *hosted line load*, linjelast som følger et element, og et python skript for en linjelast som følger gitte linjer.

Dynamo har ingen innebygd node hvor laster i Revit lages. Egne noder kodes derfor gjennom Python til å lage for eksempel linjelast i Revit [29]. De første linjene (1-19) i python scriptet (figur 6.15) gjør at Dynamo kommuniserer med Revit og blant annet importer elementer Dynamo skal arbeide med. Dersom bruen er delt inn i flere elementer er det lagt inn en *for loop* i scriptet hvor en ny linjelast lages for hver host man legger inn. På denne måten lager Dynamo en hosted linjelast på alle elementer som velges. Element hentet fra Revit må «pakkes ut», for at Dynamo kan operere på dem. Dette gjøres ved å skrive «*UnwrapElement*» i Python scriptet, vist i linje 31 for hostelementene i figur 6.15.



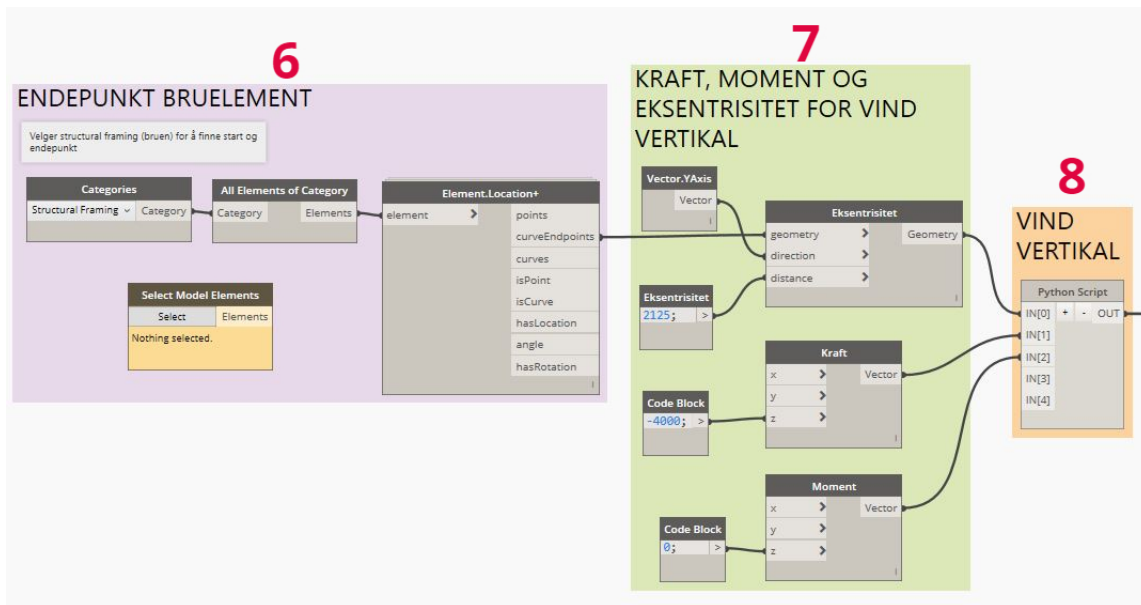
```
1 import clr
2 clr.AddReference('ProtoGeometry')
3 from Autodesk.DesignScript.Geometry import *
4 clr.AddReference('RevitAPI')
5 import Autodesk
6 from Autodesk.Revit.DB import *
7 from Autodesk.Revit.DB.Structure import *
8 from System.Collections.Generic import *
9
10 clr.AddReference("RevitNodes")
11 import Revit
12 clr.ImportExtensions(Revit.Elements)
13 clr.ImportExtensions(Revit.GeometryConversion)
14
15 clr.AddReference("RevitServices")
16 import RevitServices
17 from RevitServices.Persistence import DocumentManager
18 from RevitServices.Transactions import TransactionManager
19 from System.Collections.Generic import *
20
21 lineloads = []
22
23 host = IN[0]
24 force = IN[1].ToXYZ()
25 moment = IN[2].ToXYZ()
26 LLType = UnwrapElement(IN[3])
27
28 for i in host:
29     doc = DocumentManager.Instance.CurrentDBDocument
30     TransactionManager.Instance.EnsureInTransaction(doc)
31     l = LineLoad.Create(doc, UnwrapElement(i), force, moment, LLType)
32     lineloads.append(l)
33
34 TransactionManager.Instance.TransactionTaskDone()
35
36 OUT = lineloads
```

Figur 6.15: Hosted line load



Figur 6.16: Hosted last i Dynamo

1. Alle analytiske bjelker (bruoverbygningen) hentes fra Revit ved å finne kategorien Analytical Beams. Dette gjennomføres med *Categories* og *All Elements of Category* nodene. Det er samtidig oppnåelig å manuelt velge hvilke bjelkeelementer i Revit lasten skal påføres. Dette gjøres med *Select Model Elements*.
2. Størrelse på kraft og moment legges inn med noden *Vector.ByCoordinates*. Dette fordi kraft og moment i Pythonscriptet er definert med vektorer. Dersom det ikke er angitt verdi for input settes den automatisk lik null.
3. Python-node med det egendefinerte Python-scriptet. Denne lager linjelast i Revit.
4. Lasttilfeller må defineres manuelt i Revit, før laster kan tildeles lasttilfeller i Dynamo. I Dynamo hentes lasttilfellene med *Element Types* og *All Elements of Type*. Her er det viktig å hente ut riktig indeks for lasttilfellet lasten skal representere. Dette utføres på samme måte som materialdefineringen.
5. Lasttilfellet tildeles lasten med *Element.SetParameterByName*. Tilsvarende node er laget for størrelse på lasten, slik at det er enklere å se hvilken retning lasten har (Fx, Fy, Fz). Øvrige laster direkte tilknyttet søyle- og bruelementene kan legges til med lik fremgangsmåte beskrevet for egenvekt 2 (brubelegning).



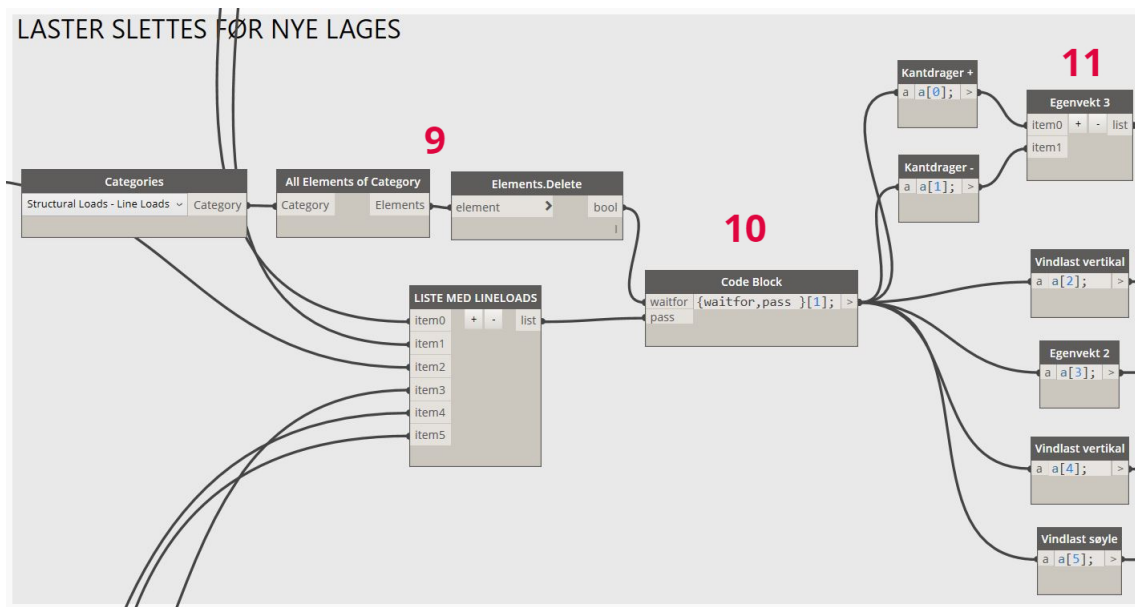
Figur 6.17: Eksentrisk last i Dynamo

6. Laster eksentrisk i forhold til hovedaksen til bruene må legges inn som en linjelast uten host. Dette fordi det finnes ingen alternativ til å legge inn eksentrisitet for hosted linjelast. Måten dette løses på er ved å lage et Python script hvor linjelasten lages mellom et start- og endepunkt.

Med *Element.Location+* fra Clockwork pakken finner man start- og endepunkt for alle elementene, hentet på samme måte som i punkt 1.

7. Eksentrisiteten legges inn ved å koble start- og endepunkt i *Geometry.Translate* sammen med ønsket distanse og retning. Størrelse på kraft og moment legges samtidig inn som input.
8. All input legget inn i Python skriptet. Videre angis lasttilfelle og eventuell korrekt størrelse, tilsvarende hosted linjelast.

I figur 6.17 er det vist eksentrisk vertikal vindlast. Samme operasjon gjøres for Egenvekt 3 (vekt av kanndragere).

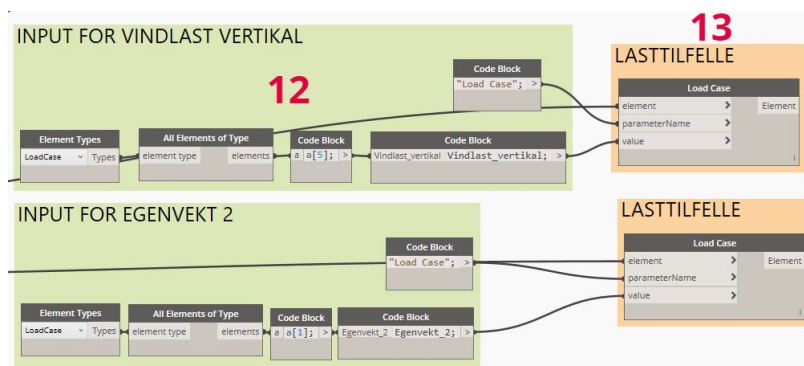


Figur 6.18: Sletting av eksisterende laster før nye lages

9. Et problem med Pythonscriptene for linjelastene er at de lager nye laster hver gang Dynamo kjøres på nytt. Dette betyr at dersom det gjøres endringer for laststørrelser, vil det lages nye laster i plasseringer der det allerede eksisterer laster i Revit.

En måte å unngå dette er ved å lage et skript hvor eksisterende laster først slettes, før nye lages. Innenfor kategorien *Structural Loads - Line Loads* hentes alle eksisterende laster.

10. For at Dynamo først skal slette eksisterende laster og deretter lage nye brukes en egen *Code Block*. Her bestemmes rekkefølgen for hvordan Dynamo skal kjøre funksjoner. Første funksjon sletter eksisterende laster med *Elements.Delete* og kobles inn i inputen *waitfor*. Andre funksjon er en liste med alle linjelastnodene. Listen kobles til inputen *pass*. Måten noden fungerer på er at den andre funksjonen venter på at den første funksjonen skal kjøre, før den passerer igjennom og kjører selv.
11. Linjelastene sorters ut, og tildels tilhørende lasttilfeller.

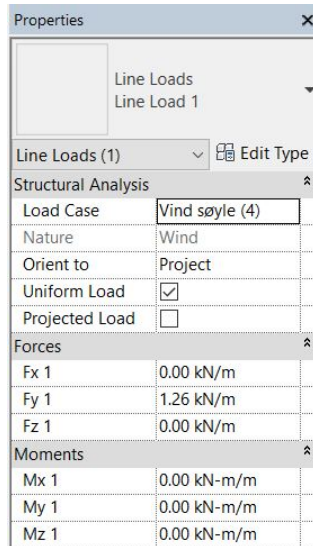


Figur 6.19: Lasttilfelle angis last

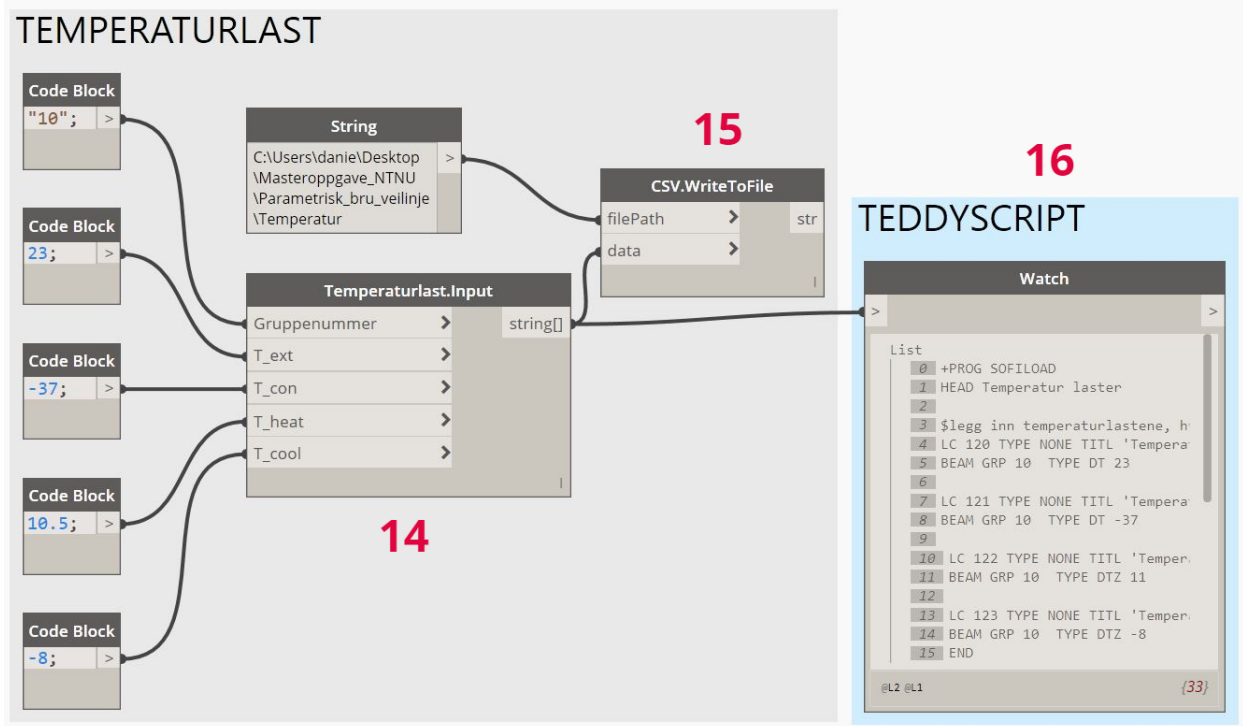
12. For å legge til lasttilfelle for lastene må alle lasttilfeller i Revit hentes. Dette gjøres ved å

hente alle lasttilfeller med *Element Types* og *All Elements of Type*. På samme måte som materiale har lasttilfellene et nummer og ikke et bestemt navn i Dynamo. Det er dermed indeksen i listen med lasttilfeller som legges til lasten.

- Tildeling av lasttilfeller gjøres på samme måte som for andre elementparametere med *Element.SetParameterByName*. I figur 6.20 vises parametere for vindlast på søyler, inkludert lasttilfelle og størrelse og retning på krefter og moment.



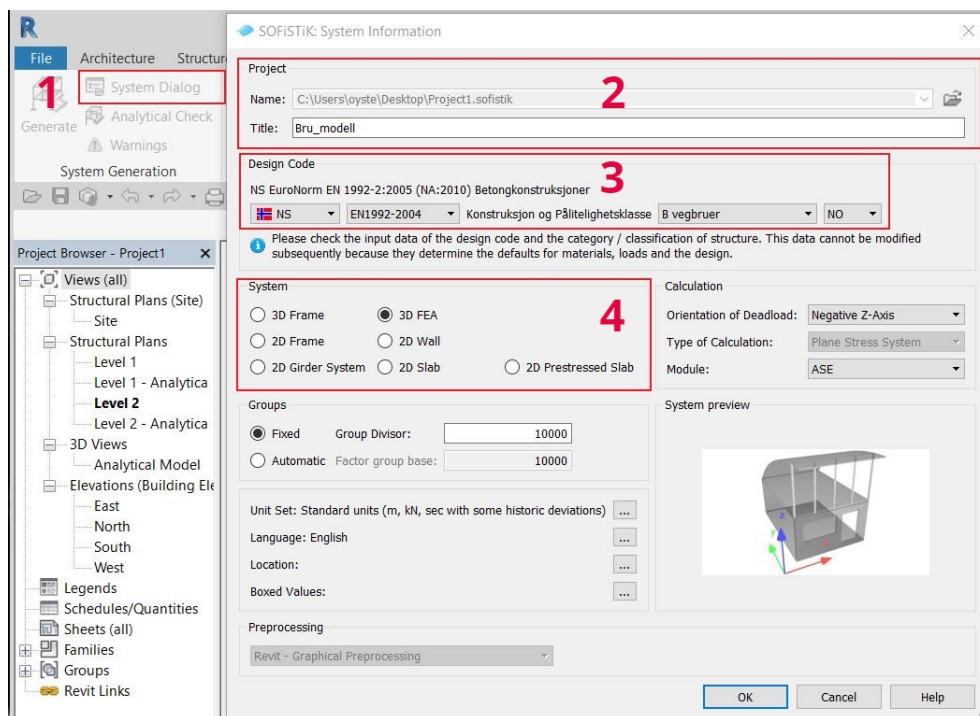
Figur 6.20: Egenskaper/parametere for laster i Revit



Figur 6.21: Temperaturlast for Sofistik i Dynamo

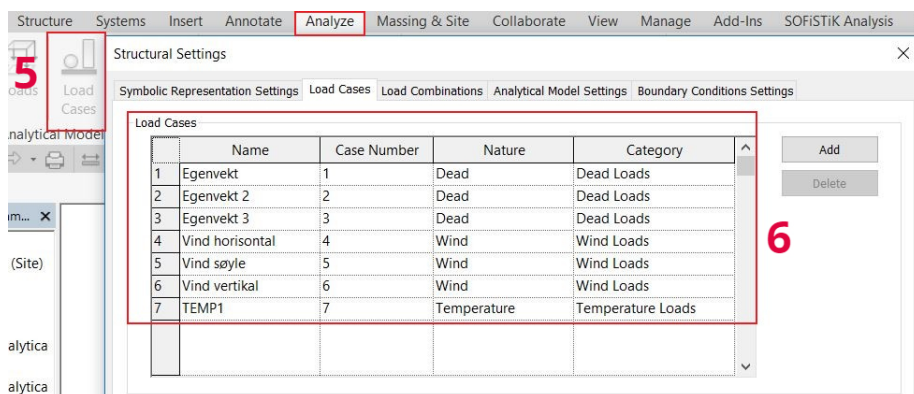
14. Egendesignet temperaturlastnode. Input *Gruppenummer* må stemme overens med gruppen man vil påføre lasten. Her er det fullt oppnåelig å benytte en rekke gruppenummer om nødvendig. Øvrige inputverdier er jevnt fordelt, samt linært variende temperatur beregnet fra Eurokode.
15. Node med lik funksjon forklart i akseopprettelse. Her overføres tekstfilen til en gitt plassering.
16. *Watch*-node for kontroll av tekstfilen. Her kan en enkelt se om designet node er av tilfredsstillende format, med ønskede verdier.

6.1.2 Revit



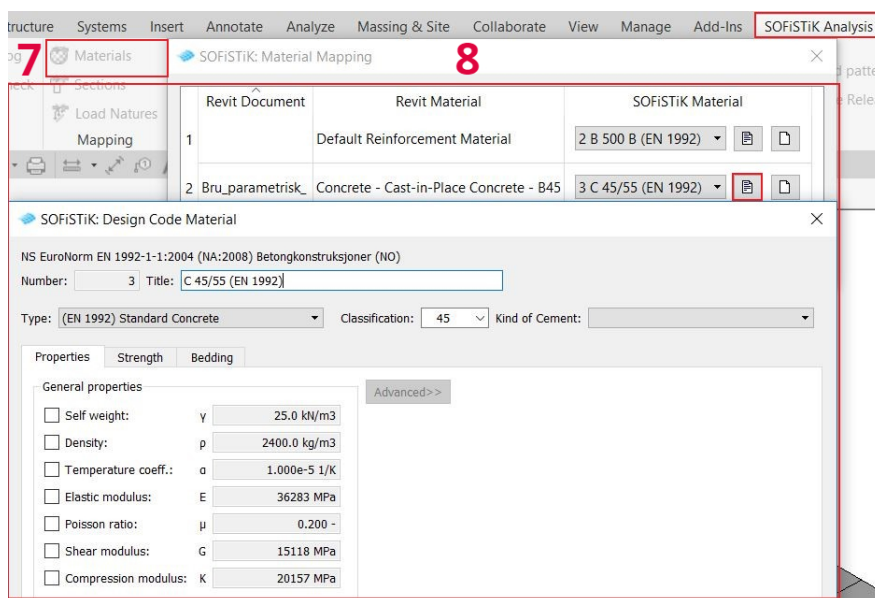
Figur 6.22: Systemdialog i Revit

1. I Revit startes systemdialogen til Sofistik for å legge inn relevant informasjon til prosjektet. Det finnes flere inputer i systemdialogen og de mest relevante beskrives her.
2. Hvor Sofistik-filen lagres med fil-navn legges inn.
3. Hvilken eurokode og konstruksjonsklasse Sofistik skal beregne etter legges inn.
4. Hvilken type system konstruksjonen skal genereres som i Sofistik legges inn. Dette innebærer blant annet rammekonstruksjoner i 2D/3D, 3D FEA og 2D dekkessystem.



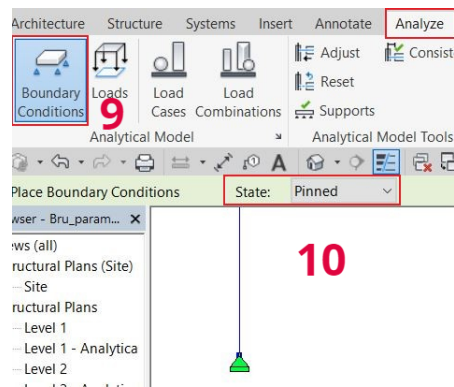
Figur 6.23: Lasttilfeller i Revit

5. I Revit må, som nevnt tidligere, lasttilfeller legges inn slik at lastene man lager i Dynamo kan tildeles riktig type last.
6. Lasttilfeller med navn, natur og kategori legges inn til bruk på lastene som lages i Dynamo. I tillegg legges lasttilfeller for laster som må legges direkte i Sofistik til. I denne oppgaven er disse temperatur- og trafikklaste.



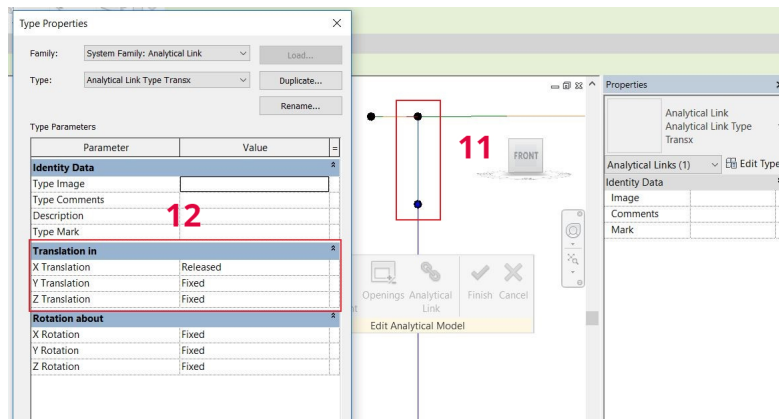
Figur 6.24: Materiale i Revit

7. Materialer til bruk i Sofistik må også defineres i Revit. Disse må stemme overens med materialene anvendt i Revit/Dynamo.
8. Material defineres. Alle valgene for materiale i Sofistik er tilgjengelig i Revit.



Figur 6.25: Opplagerbetingelser i Revit

9. I Dynamo er det mulig å legge inn opplagerbetingelser i Revit, men problemet her er at disse opplagerne ikke fungerer ved overgang til Sofistik. Opplagerbetingelser må derfor legges manuelt inn i Revit.
10. Fastlager, rullelager fastinnspenning eller egendefinert opplager legges inn. I figur 6.25 vises fastlager.



Figur 6.26: Analytisk link i Revit

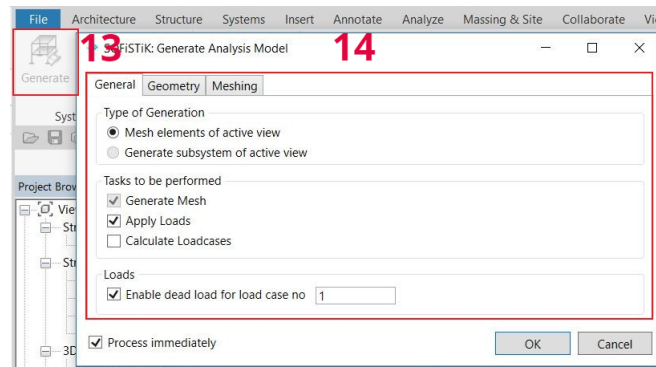
11. Analytiske linker mellom søyler og bruoverbyggingen må legges inn i Revit, dersom dette brukes.

Figur 6.26 viser en analytisk link mellom bruoverbyggingen og en søyle. Her kan man samtidig se analytiske noder, illustrerte med svarte prikker. En analytisk link kan opprettes mellom to analytiske noder for å skape en uendelig stiv forbindelse.

Her er bruene splittet i punkt over søylen. Dette gjøres for at det automatisk dannes en analytisk node. Dette skjer fordi Dynamo oppretter en analytisk node i hver ende av hvert analytisk element.

12. Fastholding av translasjon og/eller rotasjon for den analytiske linken vises i figur 6.26. Her kan translasjon/rotasjon enten være fri eller fastholdt. Dersom alle translasjoner og rotasjoner er fastholdt betyr det at linken verken kan bevege eller rotere. I figuren er x-translasjon fri og dermed vil linken kunne bevege seg i x-retning. Dette representerer

ensidig lager i brumodellen.

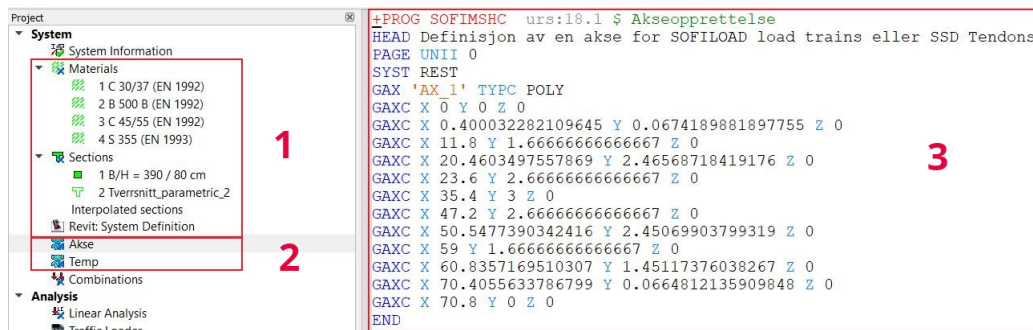


Figur 6.27: Generering av Sofistikk-fil i Revit

13. Vinduet for å generere Sofistik-filen finnes under Sofistik Analysis-fanen.
14. Input man kan bestemme blant annet størrelse på mesh, om laster skal overføres, om egenvekt av modell skal tas med, og hvilket lasttilfelle egenvekten skal tilegnes.

6.1.3 Sofistik

I Sofistik må beregningsmodellen ferdigstilles før beregningene kan kjøres. Disse stegene beskrives nærmere under.



Figur 6.28: Informasjon og oppgaver i Sofistik, samt akse

1. Materiale og profil definert i Revit er lagt til i Sofistik.

Et mindre problem med overføringen fra Revit til Sofistik er for tverrsnittsendringer. Dersom det gjøres endringer i tverrsnittet, må de først slettes i Sofistik og deretter overføre de nye tverrsnittene fra Revit. Dette er fordi Sofistik ikke oppdaterer tverrsnittene med likt navn automatisk.
2. Teddy-script med opprettelse av akse og temperaturlast legges inn gjennom egne oppgaver
3. Innhold i scriptet for akse kopieres inn fra Dynamo. Det samme gjøres for temperaturlast.
4. Videre legges enkelte andre oppgaver inn for å ferdigstille beregningsmodellen. Dette er hovedsakelig lastkombinasjoner, trafikklast, superposisjons-kombinasjoner og dimensjo-

nering for bruks- og bruddgrense. Alle disse har egne oppgaver definert i Sofistik, hvor man fyller inn data og inputen man har.

5. Etter at all input er lagt inn, kjøres og kalkuleres modellen. Deretter kan rapporter og diagrammer enkelt hentes med Sofistik sine tilleggsprogrammer.

6.2 Sammenligning av parametrisk- og tradisjonell modell

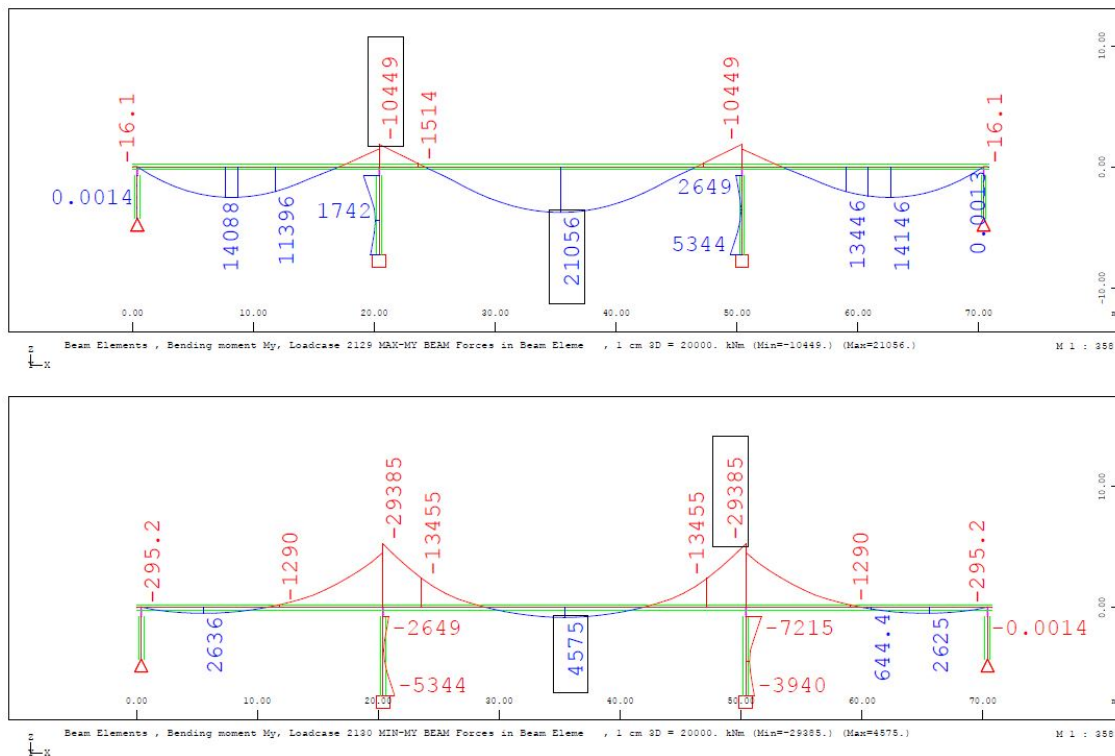
For å kvalitetssikre metoden er det naturlig med en sammenligning av resultater fra en parametrisk modell bygd rundt en rett veilinje, og tilsvarende tradisjonell modell. Denne sammenligningen bistår i kontroll av kvaliteten til den analytiske delen av arbeidsflyten. Deretter vil den parametriske modellen bli manipulert for å ta i bruk gitte XML-koordinater, slik at bruene kan følge en mer realistisk kurve.

- Rett veilinje – Kvalitet av parametrisk modell
- Veilinje med koordinater – Oppnå en modell mer nærliggende virkeligheten.

Rett veilinje

Ved parametrisk modellering av en rettlinjert bru er det ikke nødvendig med stor oppdeling av de strukturelle linjene i Dynamo. Et lite antall elementer benyttes derfor for rettlinjert bru.

Momentdiagram for maksimum og minimum bøyemoment om lokal y-akse for den parametriske modellen laget med flytmetode IV er vist i figur 6.29. Dette momentdiagrammet skal tilsvare momentdiagram man oppnår i modellen etablert på tradisjonell metode. Etersom likheten mellom diagrammene er tilnærmet eksakt velger man å gå videre med den parametriske modellen i parametrisk studie.



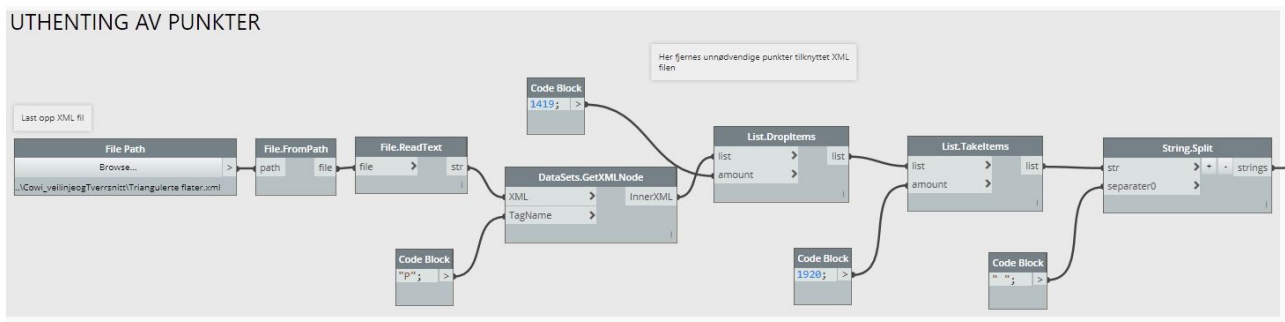
Figur 6.29: Maks/min momentdiagram parametrisk bru

Veilinje med koordinater

Det er ønskelig å teste bruene for en veilinje basert på oppmålte koordinater. Dette kan gi en tegning- og beregningsmodell vesentlig mer nærliggende virkeligheten. Da koordinatene ble gitt i XML-format fra COWI, benyttet man metode for importering av denne type format. Det er også gjennomførbart å importere inn punkter fra for eksempel excel- eller tekstdokument.

Nodefunksjonen `DataSets.GetXMLNode` er grunnlag i en slik import av XML-koordinater. Dette er ikke en innebygd node i Dynamo, men hentet fra pakken `Lunchbox`. Innhentede punkter må skaleres med 10^3 for å gi ønskelig geometriforhold opp mot Revit.

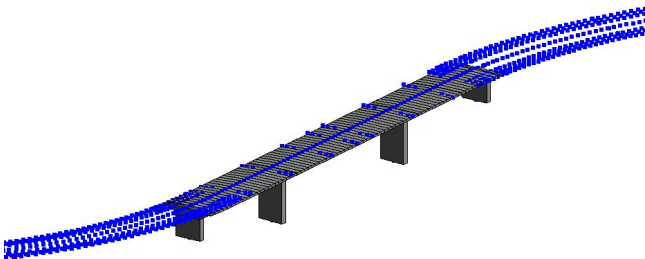
Problematikken ved denne importen lå i XML-filen. Her var det vesentlig mange irrelevante punkter, uten direkte tilknytning til bruens senterlinje. Alle punktene lå under samme navn slik utskilling av unødvendige punkter var mindre bra. I tillegg ble det krevende å finne start- og endepunkt for bruprofilet da disse heller ikke var unike.



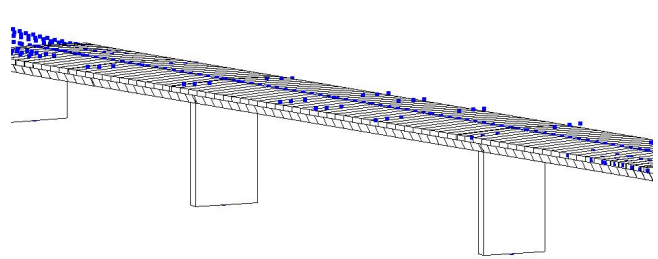
Figur 6.30: Uthenting av punkter fra XML-fil

De gitte tegningene antyder en brulengde på 70m . Da avstanden mellom punkter man betrakter som bruens start- og endepunkt tilsvarte 67m blir det vanskelig å fastslå nøyaktig start- og endepunkt for bruene. Likevel er metoden viktigere enn selve plasseringen og man ender opp med brulengde på 71m for å tilfredsstille avstandsmål gitt av tegning. Veilinjen, gitt som *NurbCurve*, er kurvet i begge plan og bruene må derfor deles inn i flere element (tidligere beskrevet). På grunn av at veilinjen nå er laget med et stort antall punkt, med flere potensielle svinger og høydeforskjeller, må elementinndelinger være svært tett.

Ut fra gitte tegninger er midtre søyler rotert slik at lokal y-retningen er parallell med underliggende vei. Dynamoskriptet tillater uavhengig rotering av søyler. Forholdet mellom denne vinklingen og bruprofil er uvisst, og blir dermed antatt. Av figur 6.32 illustreres hvordan et linjebasert element gir dårlig tegningsnøyaktighet. Oppmålte veiskuldre vil ikke knyttes til profilet. Dette resulterer i en tegningsmodell med redusert kvalitet.



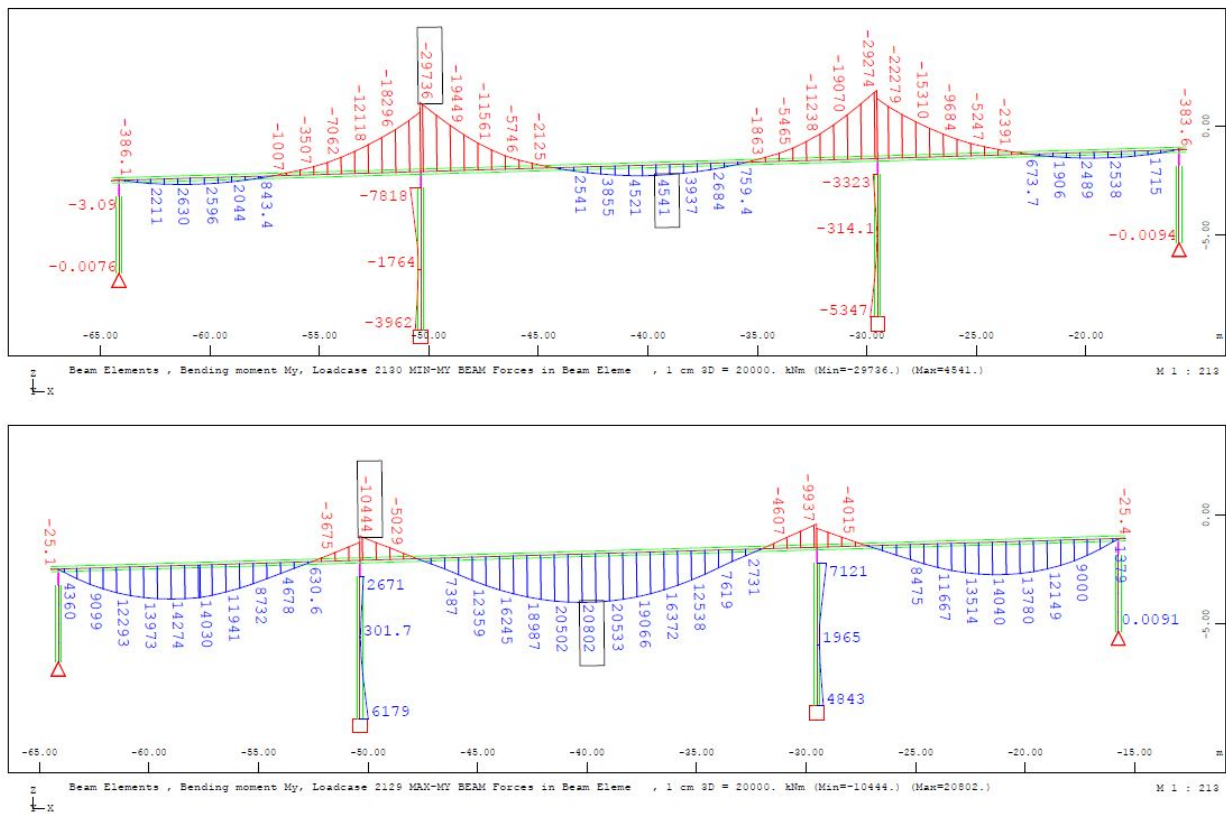
Figur 6.31: Punkter (blått) med innplassert bruprofil



Figur 6.32: Unøyaktighet ved linjebasert element

Under opprettelse av analytiske link mellom konstruksjonsdelene eksisterer det ingen lokal fastholdning, slik at for å oppnå lik forbindelse av den modellert i Sofistik antar man allsidig lager, hvor translasjon i global x- og y-retning er fri. Modellene blir deretter overført til Sofistik hvor resterende elementer for analyse blir foretatt, herunder dannelse av akse, temperatur, trafikklast og kombinasjon av lastilfeller. Verdier for maks- og min moment om y-aksen blir deretter uthentet. Disse verdiene er gitt i figur 6.33 og oppsummert i tabell 6.1.

6. PARAMETRISK MODELL



Figur 6.33: Maks- og min moment for bruprofil som følger veilinje

Tabell 6.1: Sammenligning av My-verdier

	<i>Tradisjonell metode</i>	<i>Tilhørende rett brulinje</i>	<i>Følge veilinje</i>
Maks My	21055 kNm	21056 kNm	20801 kNm
Min My	-29385 kNm	-29385 kNm	-29736 kNm

Den rettlinjede brumodellen gir tilnærmet nøyaktig like verdier for moment om lokal y-akse, sammenlignet med beregnet verdier under tradisjonell metode. Basert på dette kan man argumentere for at kvaliteten til den paramertiske modellen, med fokus på beregningsmodellen, er god. Da den analytiske rettlinjede modellen innehar god kvalitet er det tenkelig at beregningsmodellen til bruen tilknyttet en veilinje er realistiske. Verdier for minimums moment (støtte) er i denne modellen litt større enn for rettlinjete geometri. Å forenkle den analytiske brumodellen til rettlinjete vil derfor ikke nødvendigvis være en konservativ forenkling i dette tilfellet.

Brumodellen basert på en gitt veilinje ble etablert for å gi mer indikasjon på hvorvidt kvaliteten til tegningsmodellen ville være opprettholdt. Her kom det tydelig frem at en slik modell ikke inneholder ønsket kvalitet, dette mye av hensyn til valg man må gjøre for å skape en analytisk god modell.

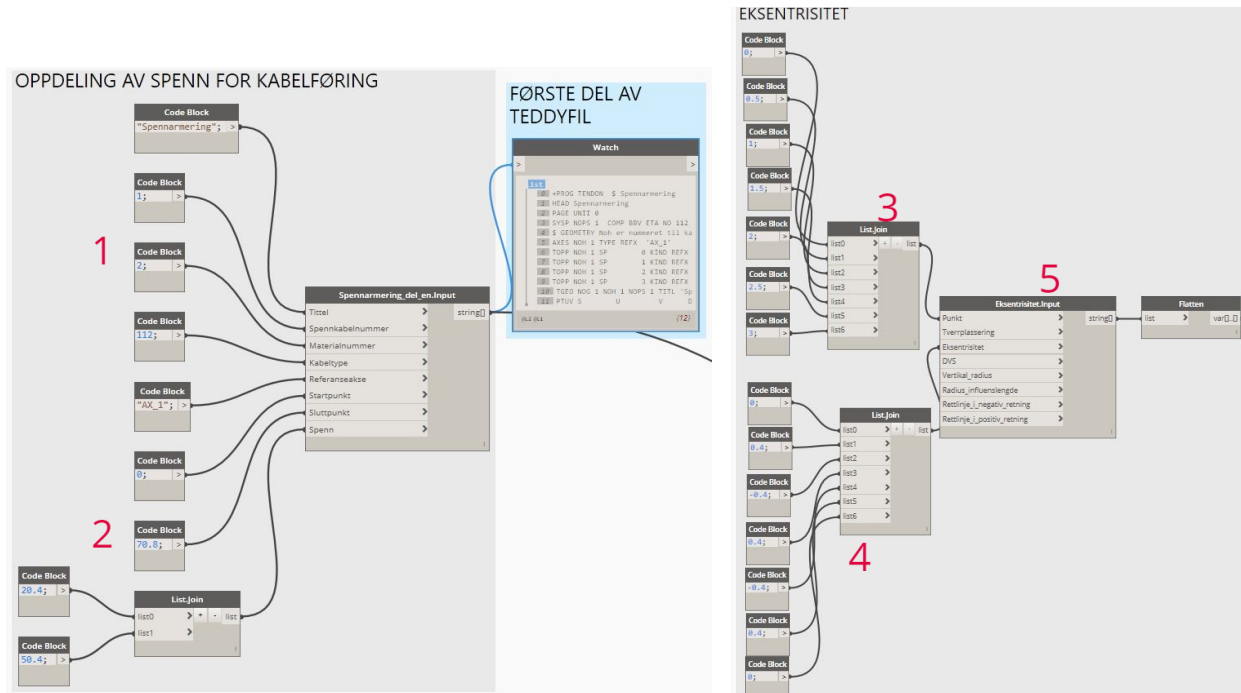
6.3 Spennarmering i den parametriske modellen

Opprettelse av spennarmering i den parametriske modellen er det på lik linje som temperaturlast, mest egnet å ta i bruk Teddyprogrammering. Gjennom slik filbehandling kan en enkelt bestemme kabelføringen. Den gjennomgående prosessen er at hovedinputten til den parametriserte brumodellen foregår i Dynamo. Det er derfor designet en egen node for spennarmering ved hjelp av Visual Studios.

Spennarmeringen er bestemt av en stor andel parametere, slik at designet vil bære preg av mindre brukervennlighet med tanke på omfanget av detaljstyringen. Designet foregikk derfor i tre typer: ”tilnærmet full styring”, ”tilnærmet automatisk dannelse”, ”eksentrisitetsbasert”. På bakgrunn av at designede noder lages i Visual Studio bør det igjen nevnes at parametriserte inputverdier velges etter behov. Dette medfører at en slik node vil kunne utbedres eller endres for å tilfredsstille ulike nytteverdi. Vedlegg I inneholder skript for spennarmering i Visual Studios format.

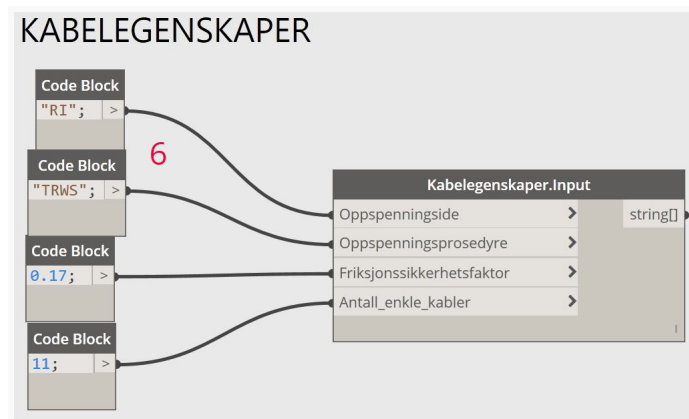
- **Tilnærmet full styring:**

I tilnærmet full styring er det designet en tre-delt node hvor flere elementer kan parametriseres. De tre ulike delene sammenslås til en enkelt tekstfil for å gi ønskelig resultat.



Figur 6.34: Spennarmeringsskript 1

Figur 6.35: Spennarmeringsskript 2

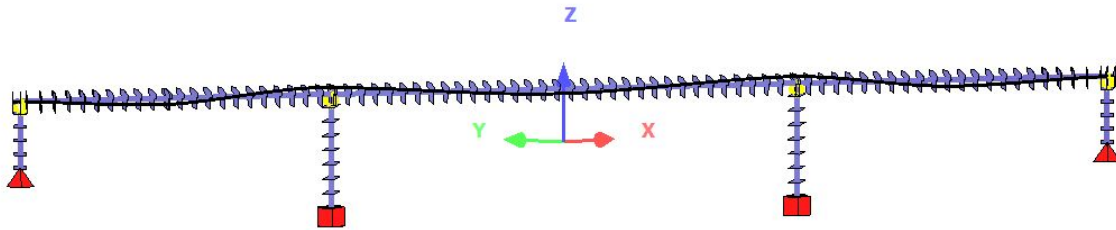


Figur 6.36: Spennarmeringskript 3

1. Her legges materialnummeret tilknyttet spennarmering i Sofistik til. Samtidig hvilken type spennarmering det skal tas i bruk. Eksempelvis eksistere BBV L3,7,9 osv. Det er nødvendig med et ett tall foran for at Sofistik gjenkjenner spennarmeringen. Med andre ord vil BBV L12 være kabeltype 112.
 2. Referanseaksen er tildelt. Aksen deles deretter opp i flere punkter basert på lengden av aksen.
 3. Viser innlagt liste av punkter
 4. Viser innlagte eksentrisiteter til tilhørende punkter. Øvrige verdier er ikke lagt til, dermed blir verdiene automatisk null i filbehandlingen.
 5. Eksentrisitetnoden. Bestående av blant annet punkt plassering der det velges referansepunkt knyttet til spennoppdelingen. Tverrplassering (U), beskriver hvor spennarmeringen ligger i tverretningen. Eksentrisitet (V) er bestemt fra aksens sentrert i bjelkesnittet hvor avstanden er gitt i millimeter. DVS er vertikal inklinasjon dv/ds hvor stor stigningen kabelen skal ha. Denne settes oftest lik null. Videre har noden inputverdier for vertikal radius, influenslengde av radius, samt eventuell om føringen skal være rett.
 6. Her velges hvilken side oppspenningen skal foregå, RI = høyre, LE = venstre, RILE = Høyre deretter venstre, LERI = Venstre deretter høyre. TWRS er oppspenningsprosedyren, hvor T = tensioning W = restressing, R = release, S = slip.
- **Tilnærmet automatisk dannels**
I tilnærmet automatisk dannels vil eksentrisiteten bestemmes fra høyden til tverrsnittet. Fordelaktig burde den designes etter tyngdepunktet, men forenklet kan dette gi en grei forslagsføring. Denne designede noden er vesentlig mer brukervennlig, samt ryddig en foregående. Men med tanke på at endring i eksentrisitet ikke kan forekomme i Dynamo er dette primært egnet for et utgangspunkt til spennarmeringens føring langs bruprofilet.
 - **Eksentrisitetsbasert**
Det tredje alternative til designet node er hvor parametrisering av eksentrisiteter kan forekomme. Her velges deler av Teddytekstfilen til å være faste for å bedre brukervennlighet til nodefunksjonen. Den designede noden er dermed en enkelt node og ikke slik øverste

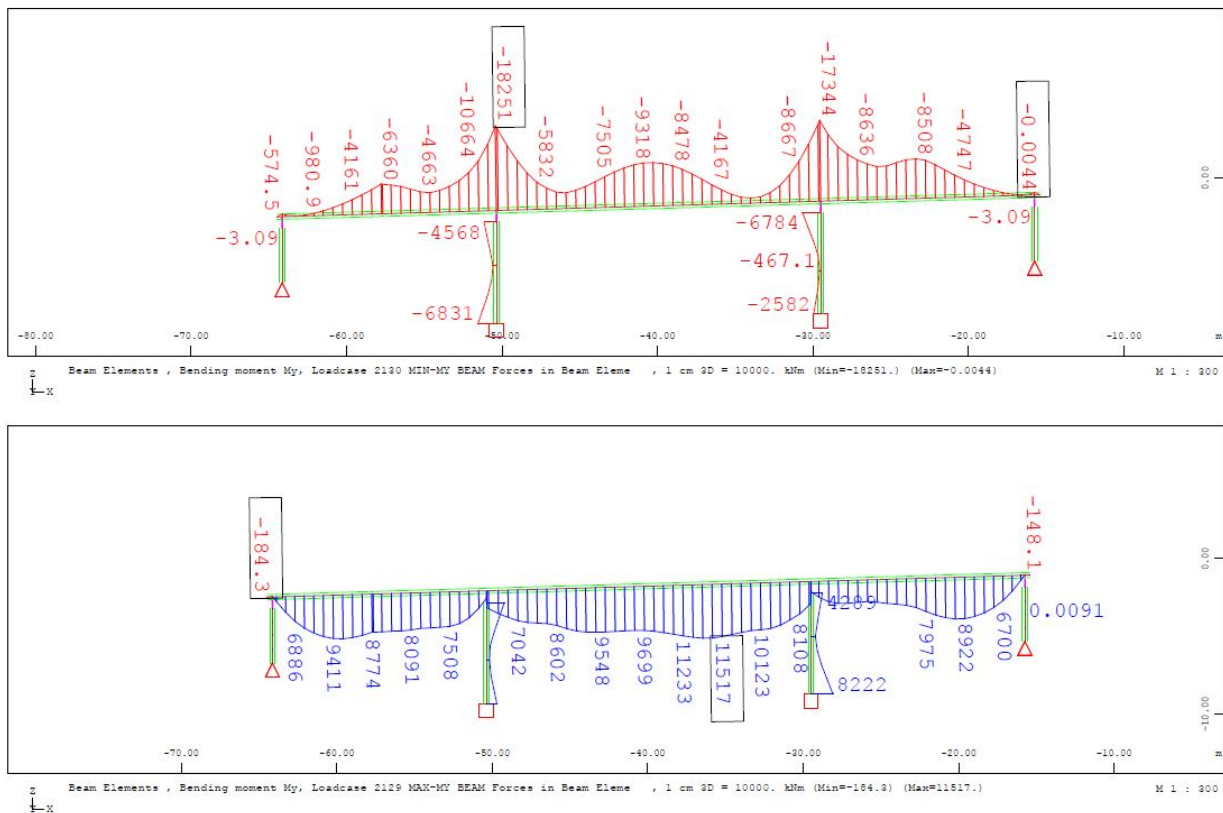
6. PARAMETRISK MODELL

alternativ, tilknyttet tre ulike deler. Noden vil være vesentlig enklere å ta i bruk, men mangle deler av parametriseringsmulighetene til tilnærmet full styring. For ytterligere forståelse av designet node se vedlegg.



Figur 6.37: Viser utplassert spennarmering ved bruk av nodealternativ 3 på bru som følger veilinje

Tidligere nevnt generer designet node ut en tekstfil. Filen er da i leselig Teddyformat slik at den deretter overføres til opprettet tekst editor i Sofistik. Ved å kjøre analysen opprettes spennarmeringen og tilhørende resultater. Dette er illustrert i brumodellen for gitt veilinje, se figur 6.37 og 6.38.



Figur 6.38: Viser maks- og minimumsmoment ved utplassering av mulig spennarmering illustrert i figur 6.37

Tabell 6.2: Spennarmering i modellen

	<i>Maks My [kNm]</i>	<i>Min My [kNm]</i>
Tradisjonell metode	18684	11088
Innsatt i parametrisk modell	18255	10831
Modell som følger veilinje	18251	11517

Sammenligning av spennarmering ved bruk av designet spennarmeringsnode og tradisjonell modelleringsgang i Sofiplus er vist i tabell 6.2. Tabellen viser at designet node ikke gir eksakt likt resultat som ved etablert kabelføring i Sofiplus. Sofiplus generer et spennkabelskript litt ulikt den egendesignede noden. Dette kom frem da man knyttet den tradisjonelle modellen opp mot egendesignet node og ikke gjennom spennarmering dannet i Sofiplus. Ved bruk av egen node oppnådde man likt resultat gitt fra den parametriske modellen.

På bakgrunn av at spennarmering ikke har den ønskelige parametriske effekt valgte man å ikke gå dypere inn på spennarmering.

7 Parametrisk studie

For å vise at den parametriske modellen faktisk fungerer etableres flere studier av bruene. Fremgangsmåten blir å teste ulike geometriske endringer for å fremme styrken av en parametriske modell. Samtidig demonstrerer studiene hvordan en parametriske modell kan bli tatt i bruk. Studiets utgangspunkt er den rettlinjede parametriske modellen uten spennarmering. Fullstendig prosjektering av en bru er ikke hovedhensikten med oppgaven, men heller det å teste hvordan den parametriske modellen fungerer. Resultatene fra beregningene er derfor mindre viktig, sammenlignet med å teste og undersøke den parametriske modellen.

7.1 Studie 1 – Spennendringer

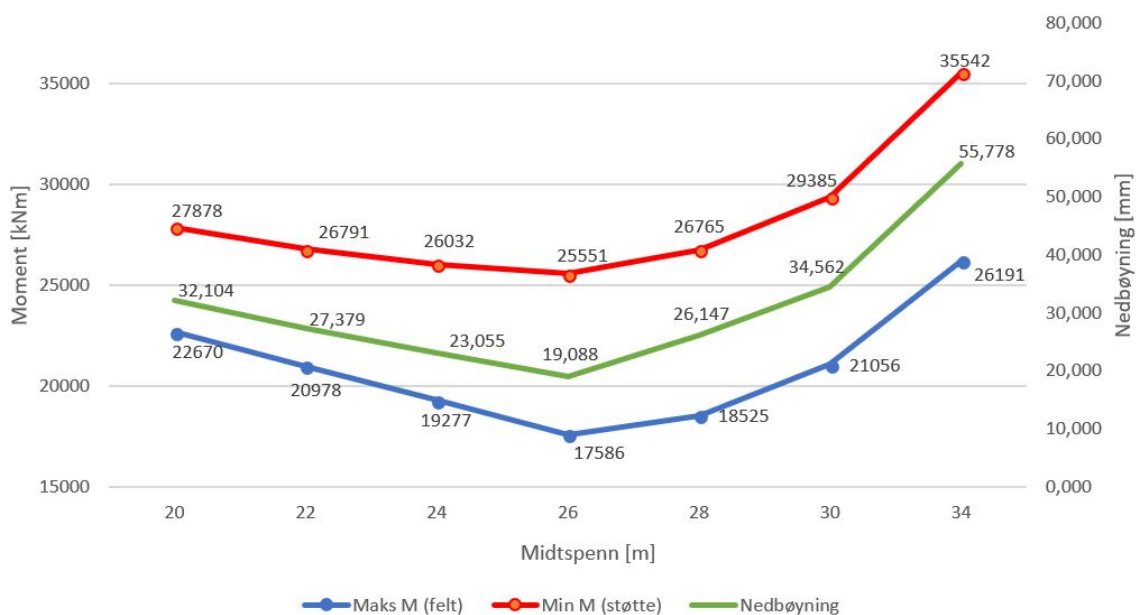
Plassering av søyler kan ofte endres underveis i et prosjekt og det utføres derfor et studie på hvordan dette gjøres med hjelp av den parametriske modellen og hvilke effekter dette har. Fra håndbok N100 Veg- og gateutforming følger det krav om 20 meter bredde for vei under bru, og midtre spenn kan dermed ikke settes mindre enn dette [30].

Plassering av søylene i modellen er enkelt å endre gjennom Dynamo. Dette kan gjøres ved å gi nye koordinater til søylene i Dynamo og deretter kjøre geometriscriptet. Deretter overføres brumodellen fra Revit til Sofistik. Lastscript er ikke nødvendig å kjøre dersom veilinjen ikke endres på. Total tid tar rundt 1 min og 40 sekunder, hvor beregning i Sofistik tar ca. 80 sekund med automatisk elementstørrelse for mesh.

Endringer av søyleplassering gir naturligvis endring av spennlengde i de tre spennene i modellen. Resultat fra syv (inkludert original plassering) er oppgitt i tabell 7.1.

Tabell 7.1: Resultater spennendringer

Spennlengder [m]	Maks M [kNm]	Min M [kNm]	Maks N [kN]	Min N [kN]	Maks Uz [mm]	Min Uz [mm]
20/30/20 (original)	21056	-29385	965	-1880	1,771	-34,562
25/20/25	22670	-27878	2306	-2,6	3,654	-32,104
24/22/24	20978	-26791	2032	-319	2,218	-27,379
23/24/23	19277	-26032	1738	-651	1,932	-23,055
22/26/22	17586	-25551	1429	-994	1,205	-19,088
21/28/21	18525	-26765	1186	-1421	1,027	-26,147
18/34/18	26191	-35542	528	-2810	4,097	-55,778



Figur 7.1: Maks/min moment og nedbøyning for spennendring

Fra resultatene i tabell 7.1 tydeliggjøres nytten av parametrisk design ved at flere forskjellige søyleplasseringer testes. Her er det lett å velge ut søyleplasseringen hvor det beste resultat oppstår. Spenn på 22/26/22 meter gir tydelig ganske gode resultater med hensyn til maks/min moment og nedbøyning i forhold til original modell. Dette synliggjøres i figur 7.1 med absoluttverdier for maksimum og minimum moment, det vil si henholdsvis felt- og støttemoment. Figur 7.1 viser videre at spenn på 22/26/22 gir lav nedbøyning i forhold til de andre. I dette studiet kommer nytten av en parametrisk modell godt fram, hvor optimalisering av bruene enkelt gjøres ved å se på tabeller og diagram.

Figur 7.1 illustrerer i tillegg at lange spenn gir høyere felt- og støttemoment. Dette er forventet basert på enkel statikk. En bru hvor midtspennet er lengst, men ikke mye lengre enn ytterspennene gir best resultat. Følgelig er spenn 23/24/23, 22/26/22 og 21/28/21 å foretrekke.

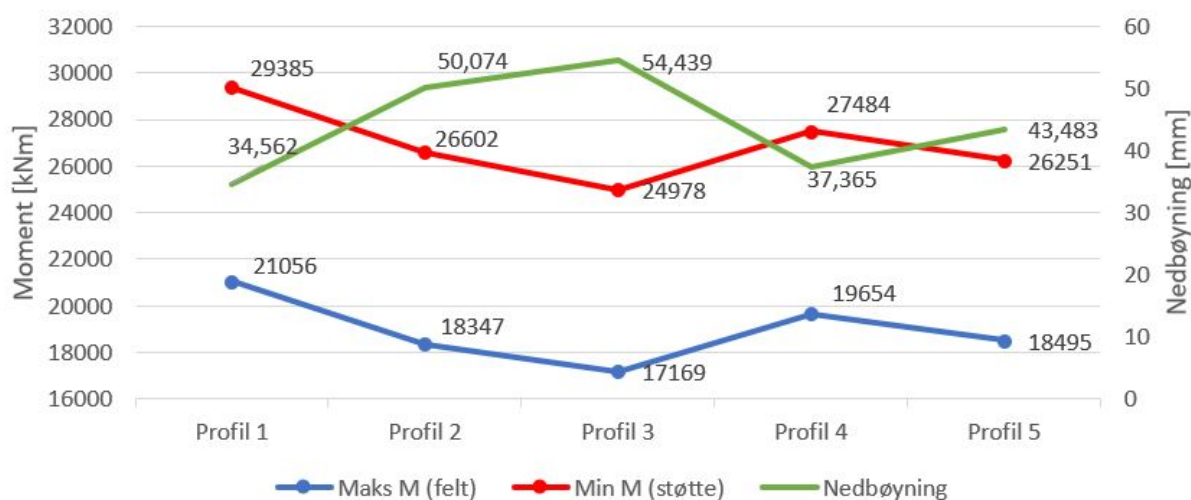
7.2 Studie 2 – Endring av brutverrsnitt

For å teste hvordan den parametriske modellen fungerer for endring av tverrsnitt lages tre alternative profil til det originale. Endringer av tverrsnittsdimensjoner gjøres for total høyde og bredde i bunn. Øvre bredde endres ikke i disse tilfellene grunnet hvordan laster er lagt inn i både Dynamo og Sofistik, men det er likevel fullt mulig. Profil 1 i tabell 7.2 er profil gitt av COWI.

Tiden disse endringene tar er ca. like lang tid som søyleendringene. Det betyr at det ikke er tidskrevende å gjøre endringer på tverrsnittet. Dimensjonene for tverrsnittet endres gjennom parametere i Dynamo og er beskrevet i kapittel 6.1.1. Lastscript er ikke nødvendig å kjøre på nytt dersom lastene allerede er i modellen. Tverrsnittene i Sofistik må i tillegg slettes før nye tverrsnitt overføres, som beskrevet i kapittel 6.1.3.

Tabell 7.2: Resultater for tverrsnittsendringer

Spennlengder [m]	h_{total} [mm]	b_{bunn} [mm]	Volum [m ³]	Maks M [kNm]	Min M [kNm]	Maks V [kN]	Min Uz [mm]
Profil 1	1250	3900	451	21056	-29385	6093	-34,562
Profil 2	1000	3900	372	18347	-26602	5525	-50,074
Profil 3	1000	3000	327	17169	-24978	5205	-54,439
Profil 4	1250	3000	400	19654	-27484	5725	-37,365
Profil 5	1100	3200	363	18495	-26251	5459	-43,483



Figur 7.2: Maks/min moment og nedbøyning for tverrsnittsendringer

Fra figur 7.2 er det synlig at høyden for tverrsnittet utgjør en større forskjell for moment enn bredden i bunn. Samtidig øker nedbøyningen ved mindre høyde. Profil 4 og 5 kan sees på som gode alternativ med lavere moment enn profil 1, men litt høyere nedbøyning. Profil 4 og 5 har henholdsvis 51m³ og 88m³ mindre betong enn profil 1, til stor fordel. Et spørsmål er om profil 4 har nok plass til slakk- og spennarmering, men dette vurderes ikke her. Skjærkreftene er forholdsvis like for alle fem profil, men skjærkapasiteten vurderes ikke her.

Håndbok N400 Bruprosjektering gir krav for nedbøyning på $L/350$ for det betraktede spenn [22]. For lengste spenn med 30m er kravet dermed:

$$L/350 = 30000/350 = 86mm$$

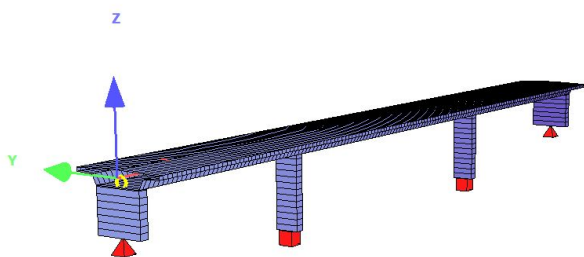
Alle profilene er innenfor dette kravet, men dette er bare et raskt overslag og nærmere beregninger må gjøres.

Sammenlignet med studie 1 er optimalisering gjennom tabeller og diagram vanskeligere, da disse ikke gir et like klart bilde av beste utforming. Av den grunn må tverrsnittet bestemmes på bakgrunn av andre hensyn enn de resultatene brukt her.

7.3 Studie 3 – Endring av søyleverrsnitt

Dette studiet ser på endringer av tverrsnittsgeometri for søyler. Endring av søyleverrsnitt eller søylenengde medfører endringer i stivhetsmatrisen til rammeanalysen. Med fri x-translasjon i søyleforbindelsen vil søylen ikke bidra i stivhetsmatrisen. Momentdiagrammet vil da kun bli påvirket av stivhetsendringer for de fastholdte søylene, det vil si midtre søyler. En økning av stivheten for disse søylene resulterer i større moment over søylene forårsaket av at den økte stivheten «tar opp» flere krefter. Fokuset i dette studiet er av den grunn på midtre søyler, da lagerbetingelsene forblir uforandret, og ytre søyler har dermed ikke påvirkning på momentdiagrammet.

Gjennom studiet er referering til høyde og bredde tilknyttet søyleverrsnittet i xy - planet, hvor høyden er i x-retning. «r» i tabell 7.3 er radius av sirkulære søyleverrsnitt.

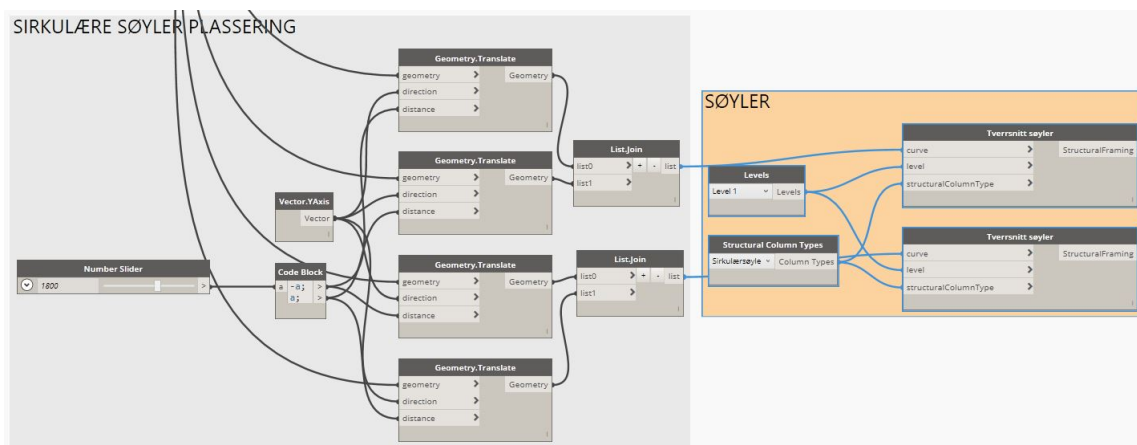


Figur 7.3: Midtre søyler 2000mm x 800mm



Figur 7.4: Søyler med diameter 900 mm

Bruer har ikke nødvendigvis alltid rektangulære søyler, og et modifisert Dynamoscryptet benyttes dermed til å endre søyleprofil til to sirkulære søyler. Modifiseringen er gitt i figur 7.5. Forskjellen fra script med rektangulære søyler er at to linjer legges inn i de fire søyleplasseringene, og en annen egendefinert søylefamilie benyttes. Søylefamilien styres av en parameter for radius slik at tverrsnittet blir sirkulært.



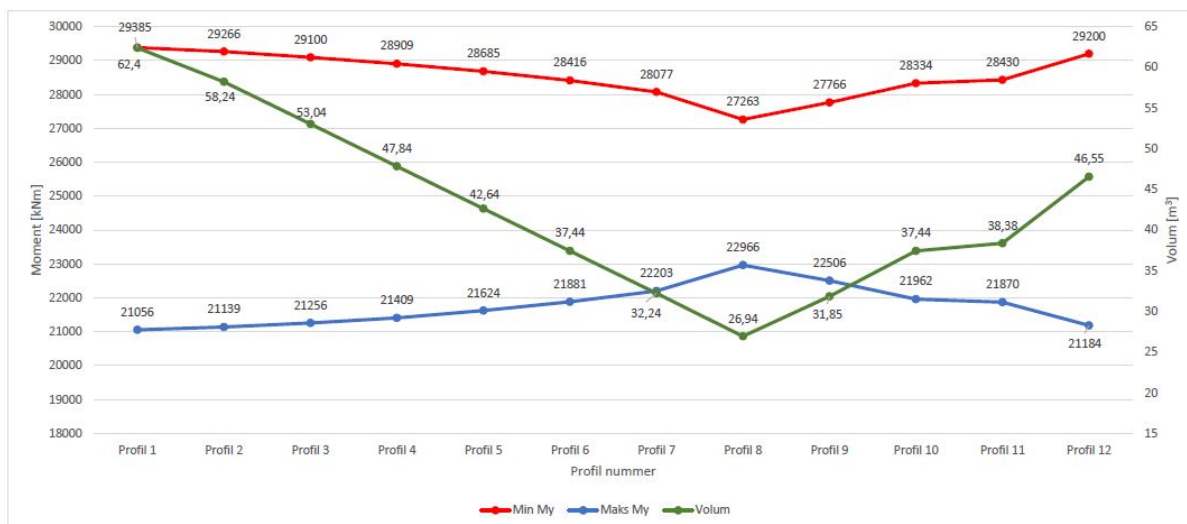
Figur 7.5: Dynamoscrypt modifisert for to sirkulære søyler

7. PARAMETRISK STUDIE

Arbeidsomfanget for endring av søyletverrsnitt er relativt likt de tidligere studiene, og dette er en gjenganger for alle studiene. Generelt over er det ikke spesielt tidskrevende å endre parametere i Dynamo, hvor hovedvekten av arbeidet utføres. På samme måte som i studie 2, må tverrsnittene i Sofistik slettes før oppdaterte overføres.

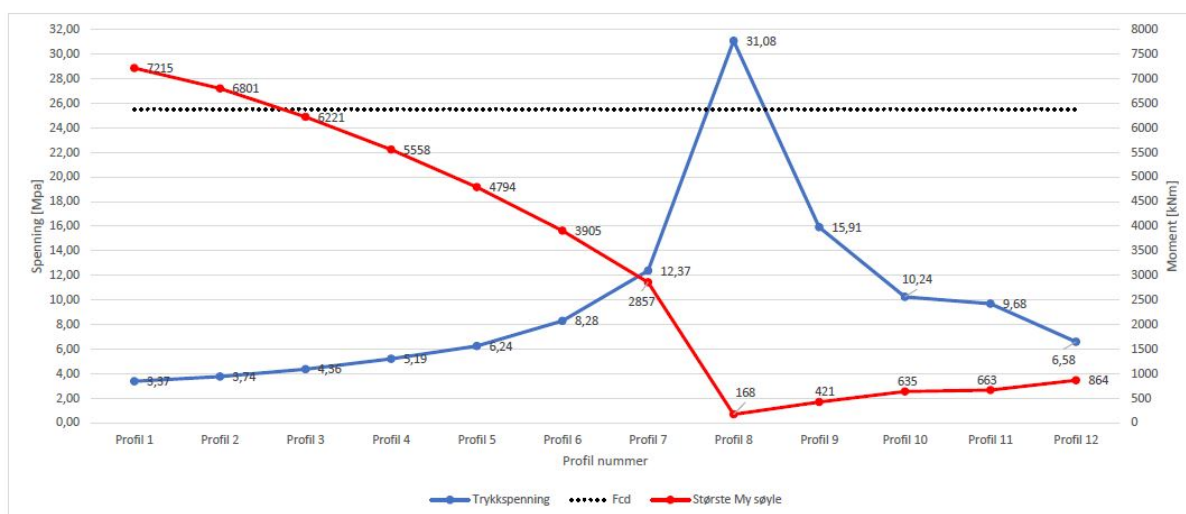
Tabell 7.3: Endring av søyletverrsnitt

	Type bxh, r [mm]	Volum [m ³]	Min M _y	Maks M _y	σ søyle	M _y søyle
Profil 1	3900 x 800	62,4	-29385	21056	3,37	7215
Profil 2	3500 x 800	58,24	-29266	21139	3,74	6801
Profil 3	3000 x 800	53,04	-29100	21256	4,36	6221
Profil 4	2500 x 800	47,84	-28909	21409	5,19	5558
Profil 5	2000 x 800	42,64	28685	21624	6,24	4794
Profil 6	1500 x 800	37,44	28416	21881	8,28	3905
Profil 7	1000 x 800	32,24	28077	22203	12,37	2857
Profil 8	250	26,94	27263	22966	31,08	168
Profil 9	350	31,85	27766	22506	15,91	421
Profil 10	437	37,44	28334	21962	10,24	635
Profil 11	450	38,38	28430	21870	9,68	663
Profil 12	550	46,55	29200	21184	6,58	864



Figur 7.6: Momentinnvirkning for brudekket ved ulike søyleprofil, volumforskjell illustrert i grønt

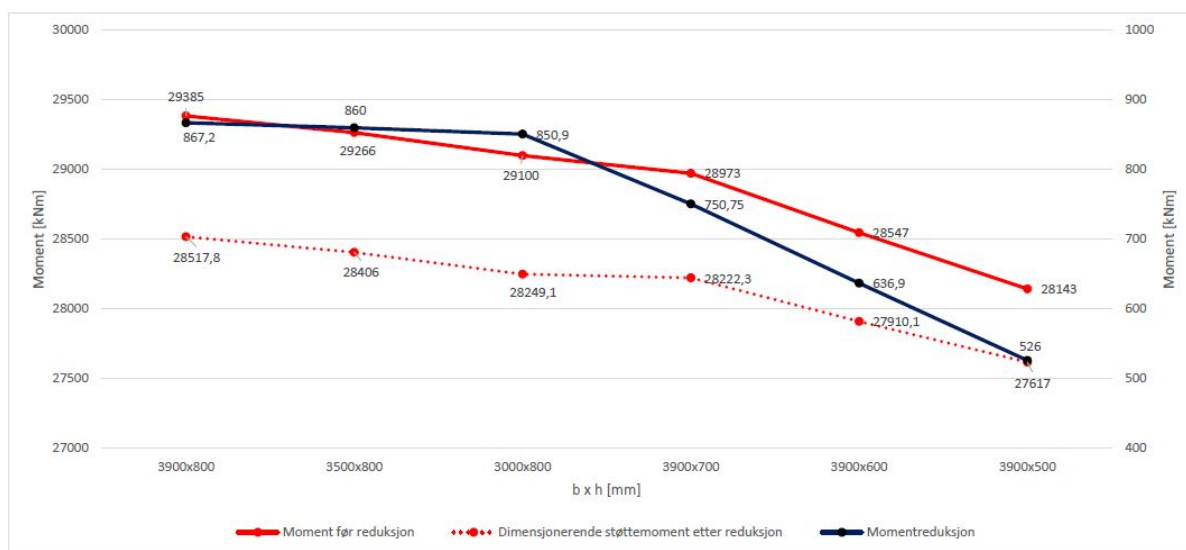
7. PARAMETRISK STUDIE



Figur 7.7: Tykkspenning og moment med ulike søyleprofil

Observert gjennom studiet er det ingen overraskelse at forholdet mellom maksimalt støtt- og feltmoment utjevnes ved mindre stiv søyle. Under studiet er det på rektangulære søyler kun endret på bredden. Dersom høyden av søylen var den mer varierende faktoren ville stivhetforskjellen trolig blitt tydeliggjort i større grad med tanke på at $I = (b \cdot h^3)/12$. Da søyletykkelsen allerede anses som relativ tynn, ble den ikke endret i første omgang.

Likevel ble det valgt å undersøke hvilken påvirkning søyletykkelsen vil ha på minimumsmomentet for brumodellen. På bakgrunn av at det foreligger en mulighet til å redusere støttemoment over søylene, gjennom $\Delta M_{ed} = F_{ed,sup} \cdot t/8$, ble det vesentlig å sammenligne opp mot endelig momentverdier. Innvirkningen søyletykkelsen vil ha på minimumsmomentet er illustrert i figur 7.8.

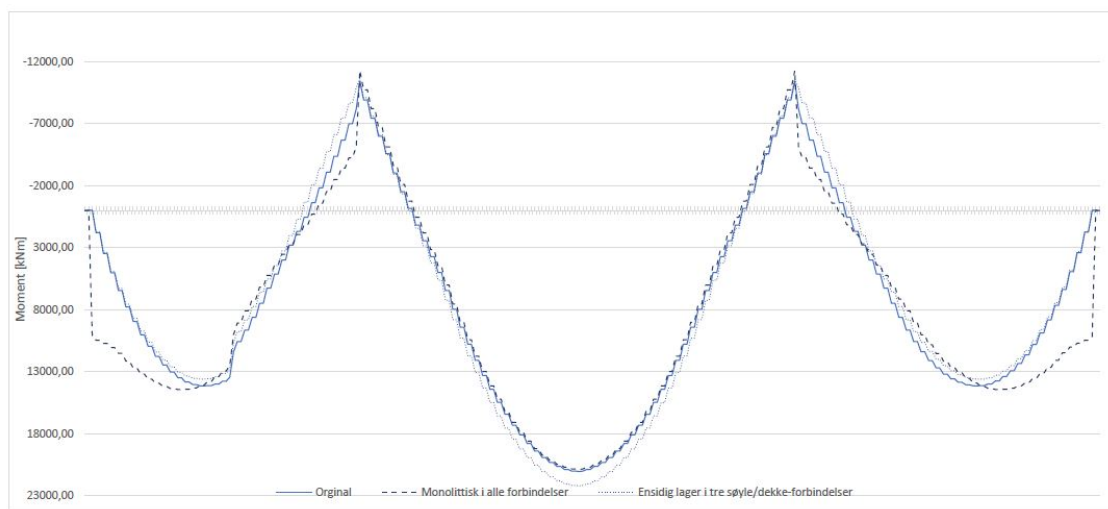


Figur 7.8: Effekt av endring i søyletykkelsen

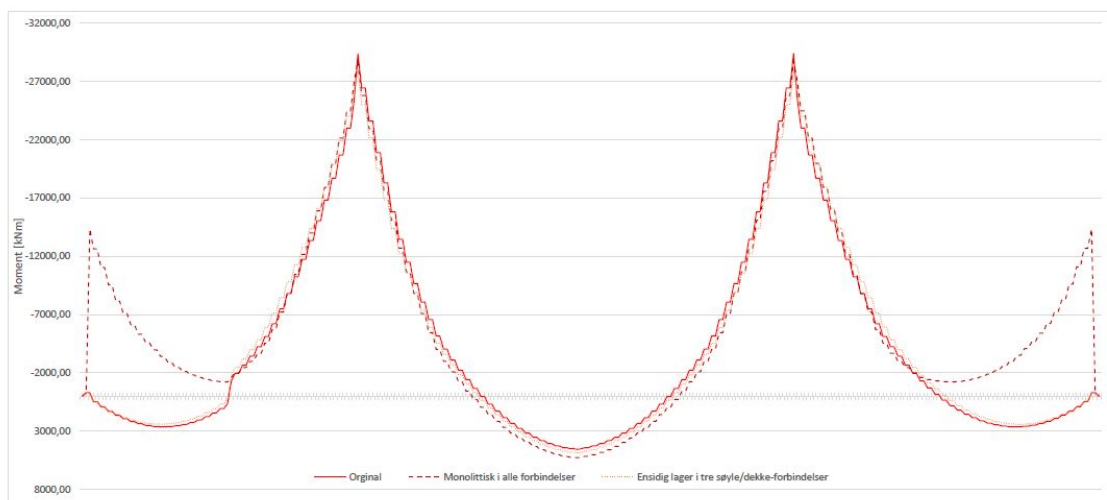
7.4 Studie 4 – Stivhetsendringer

Et mer fordelt momentdiagram i bruprofilet kan være fordelaktig. Da brutverrsnittet oftest har arealmessig mer betong i øvre del vil tyngdepunktet til tverrsnittet bevege seg mot øvre ytterkant. Dette betyr at eksentrisiteten til spennarmeringen blir fortere begrenset i overkant av tyngdepunktet kontra nedsiden, og derav gi mindre mulighet til motvirkning av momentdiagrammet.

Tverrsnittsendring av søyler i studie 3 ga ikke store forskjeller i momentdiagrammet for bruoverbyggingen. Et forsøk på å få mer fordelt momentdiagram gjennom endring av lagerbetingelsene mellom søyle og bruoverbyggingen gjøres derfor i dette studiet. I den originale versjonen er det valgt ensidig lager i endesøylene, følgelig er global x-retning fri for bevegelse.



Figur 7.9: Maks moment i brudekket ved endring av lager, $X = 0$ m t.v, $X = 70,8$ m t.h



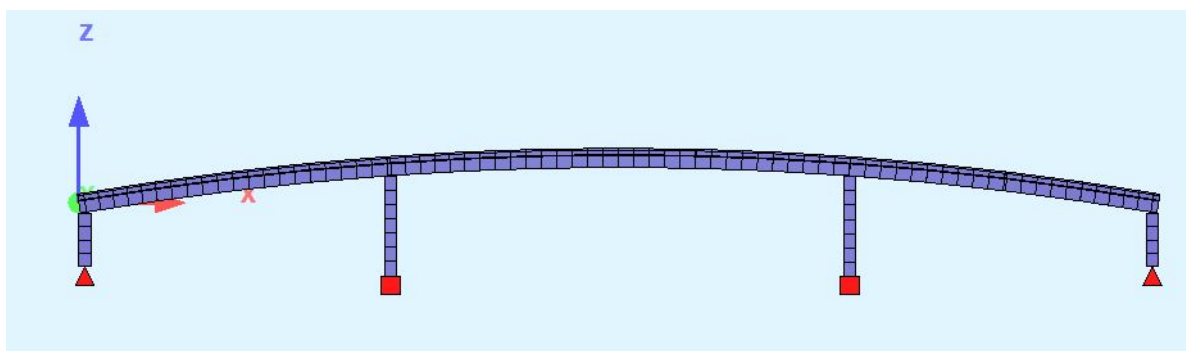
Figur 7.10: Min moment i brudekket ved endring av lager, $X = 0$ m t.v, $X = 70,8$ m t.h

Endring i lagerbetingelser resulterer i små endringer for moment om y-aksen i brudekket. Man ser ved monolittisk system i alle forbindelsene vil det oppstå et stort moment ved ytre søyler og derav ned i søylefoten. En bedre løsning av lagerbetingelser kan være ved bruk av ensidig lager i midtre samt en av ytre søylene. Brudekket blir da fastholdt ved et monolittisk system i den resterende ytre søylen. Dette resulterer i et mer utjevnet forhold mellom maksimum- og minimumsmoment.

Tabell 7.4: Endring i søyle/dekke -forbindelsen

	Min My [kNm]	Maks My [kNm]	Maks/Min Ux [mm]	Min Uz [mm]
Original	-29385	21056	13,9	-35
Monolittisk system	-28756	20885	13,2	-32,8
Ensidig lager i tre forbindelser	-27962	22212	24,6	-36,6

7.5 Studie 5 – Endring i høyde



Figur 7.11: Kurvet bru i Sofistik, 3m høyde

Styring av høyder og svinger er viktige parametere ettersom dette ofte endres. For å vise endring av høyde/stigningsgrad er tegningsgrunnlaget gitt av COWI lagt til grunn. Bruen fra COWI er i vegklasse Sa3, ÅDT 30 med fartsgrense 80 km/h. Etter Statens Vegvesen sine håndbøker om Veg- og gateutforming er maks stigning i prosent for denne bruklassen 8 % [30].

Stigning legges inn ved å endre på for eksempel høyden av midtpunktet i linjen. *NurbsCurve.ByPoints* lager da en buet kurve. Siden bruene lages i flere element får de ytterste elementene mest stigning. Dette begrenser høyden i midten av bruene til å være rundt maks 1.5m for at hele bruene skal ha mindre stigning enn 8 %. På bakgrunn av dette er det laget fire forskjellige høyder for midten av bruene på 0.5m, 1.0, 1.5m og en ekstra høyde på 3.0m for å sjekke om den utgjør store endringer. Det kan legges til flere høydeforskjeller langs veilinjen ved å endre på høyden av flere punkter, men her er det kun gjort med punktet i midten av bruene for å illustrere hvor enkelt det lar seg gjøre og effekten av høydeendringen. Figur 7.15 viser skriptet for endring av punktene i veilinjen.

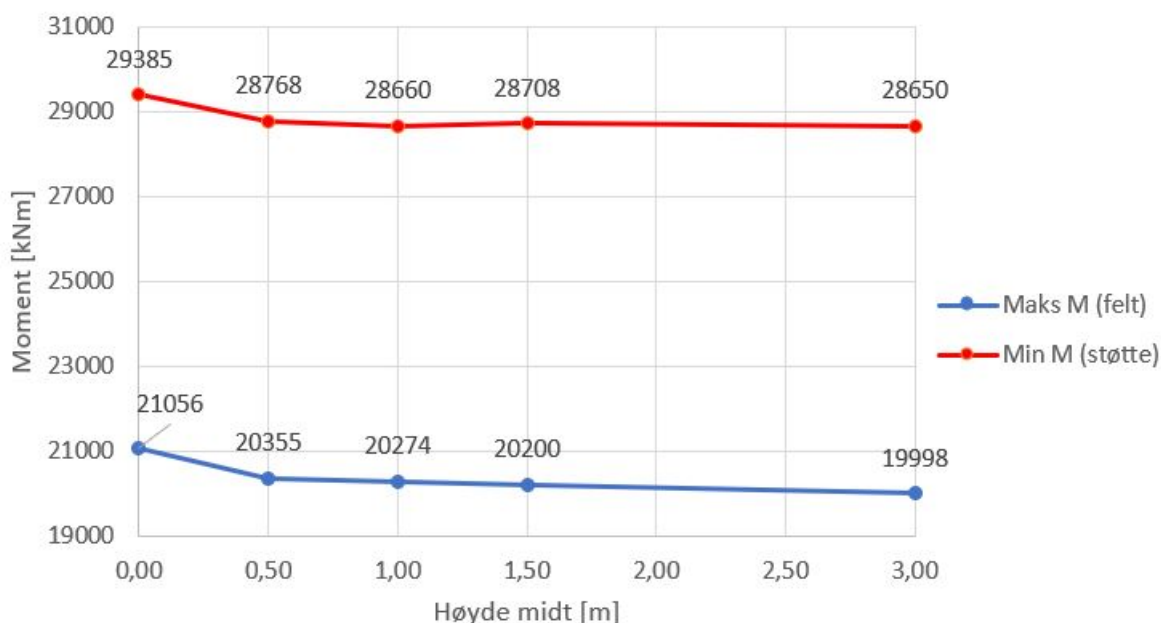
Arbeidsomfanget for disse fem (inkludert 0 meter høyde) er ikke stort. I første omgang kjøres geometriscript med en høydeendring. Videre genereres lastscriptet slik at laster følger riktige

7. PARAMETRISK STUDIE

linjer og elementer. Deretter overføres modellen til Sofistik, hvor beregning kjøres på nytt. Ny oppdatert akse fra Dynamo legges også inn i Sofistik. Beregningene i Sofistik tar ca. 80 sekund og arbeid i Revit/Dynamo tar ca. 40 sekund. En helt ny modell er dermed laget på to minutt, hvor beregning av modellen står for over halvparten av tidsbruken.

Tabell 7.5: Resultat for høydeendring av bru

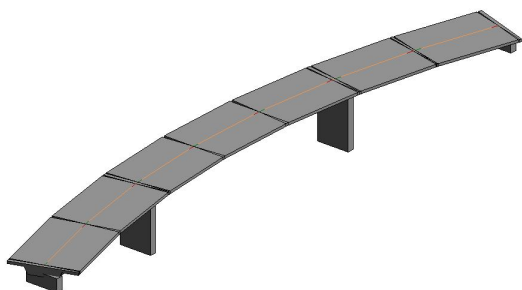
Høyde i midten av bru [m]	Maks M [kNm]	Min M [kNm]	Maks N [kN]	Min N [kN]	Maks Uz [mm]	Min Uz [mm]
0	21056	-29385	965	-1880	1,771	-34,562
0.5	20355	-28768	927	-1919	1,650	-33,405
1.0	20274	-28660	907	-1985	1,609	-33,228
1.5	20200	-28708	880	-2067	1,573	-33,084
3.0	19998	-28650	814	-2315	1,473	-32,741



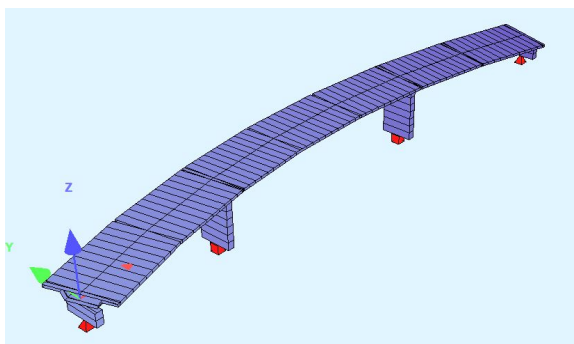
Figur 7.12: Maks/min moment for høydeendring, absoluttverdi

Av resultatene kommer det frem at med en buet bru blir momentene mindre, både felt- og støttemoment. Dette er forventet oppførsel, men man ville antatt at resultatene skulle vært enda bedre desto høyere og mer buet bru blir. Dette kan komme av at bruene ikke er fastholdt i endene i x-retning og dermed blir ikke trykkreftene i bruene vesentlig større (Min N i tabell 7.5). Dette fører igjen til at momentene ikke blir særlig mye mindre. Likevel er oppførselen til bruene på vei i riktig retning med tanke på resultatene for en buet bru, illustrert i figur 7.12. Det er nesten ingen forskjell i nedbøyning for noen av høydene.

7.6 Studie 6 – Endring av veilinje

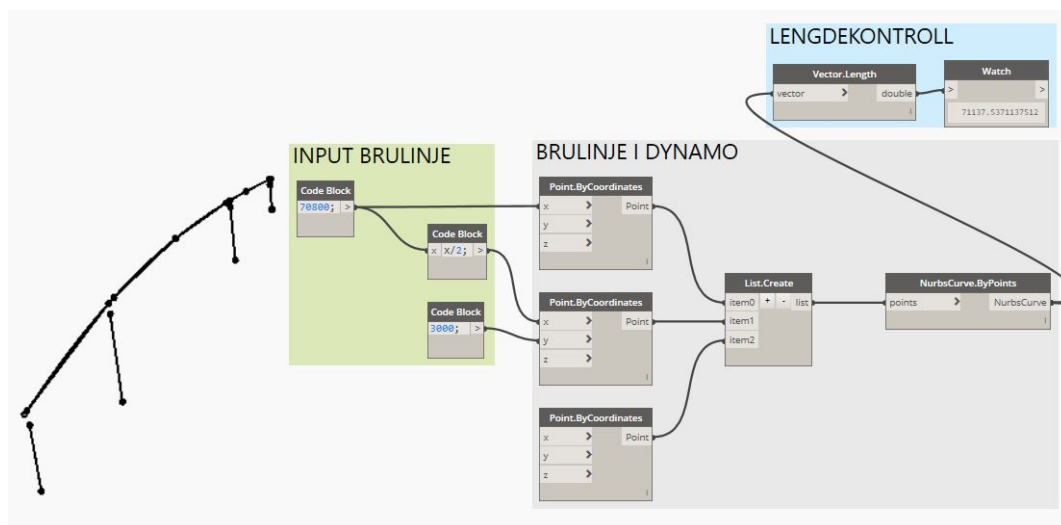


Figur 7.13: Kurvet bru i Revit



Figur 7.14: Kurvet bru i Sofistik

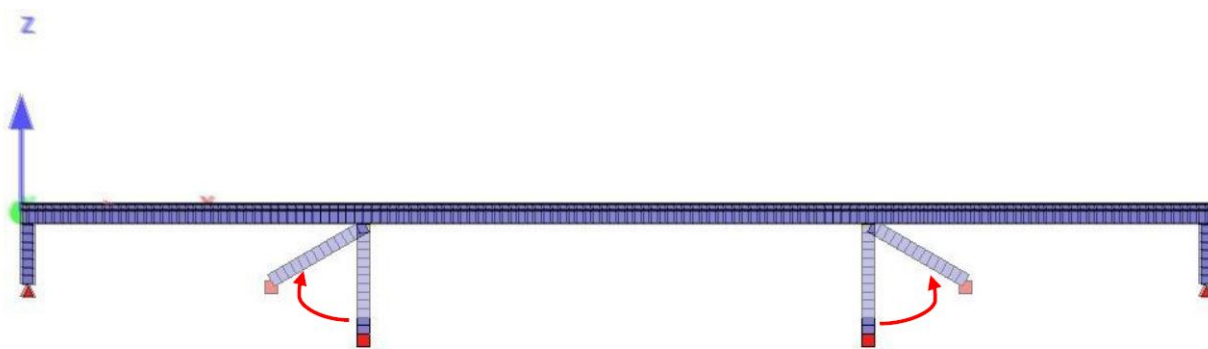
Endring av veilinjen i planet gjøres på samme måte som for høydeendringer. Uten en gitt veilinje (med f.eks. excelpunkter) gjøres dette ved å legge inn en forskyvning i planet slik at bruene blir kurvet når punktene settes sammen til en *nurbs curve*. Dette er illustrert i figur 7.15 med en horisontal forskyvning av midtpunkt på 6 meter. Arbeidsflyt forklart i studie 4 følger heretter og arbeidsomfanget er følgelig likt.



Figur 7.15: Kurvet bru i Dynamo

Tegningsmodellen illustrert i figur 7.13 ser man åpenbare store åpninger mellom bruprofilene. Årsaken er at brumodellen ikke er oppdelt i nok elementer. En tett oppdeling av elementer vil redusere åpninger, men til gjengjeld gjøre modellen mer tidkrevende å etablere. Poenget med studiet er å vise hvordan en kurvet veilinje er mulig å lage. Studie fremstiller samtidig svakheten i tegningsmodellene ved kurvet veilinje.

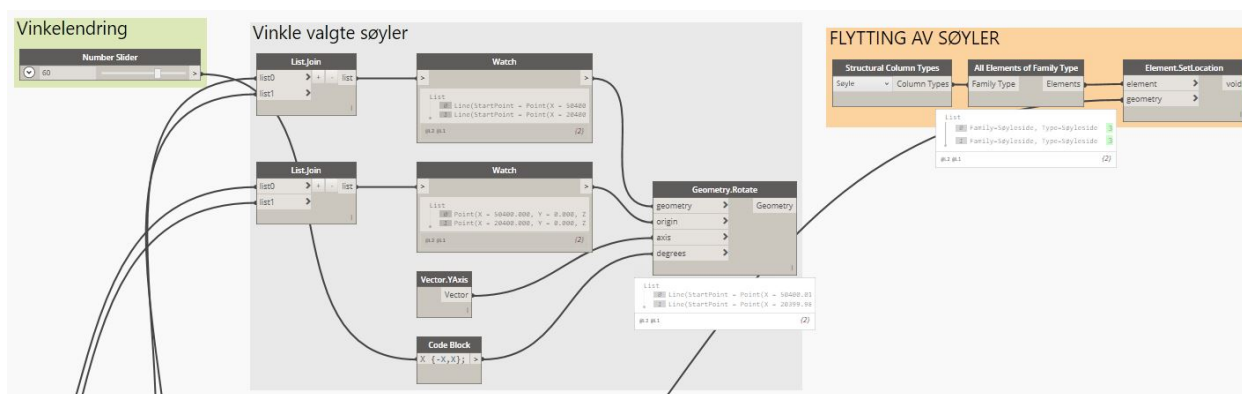
7.7 Studie 7 – Endring av søylevinkel



Figur 7.16: Endring av søylevinkel

Vinkler på søyler kan forekomme på bakgrunn av konstruksjons- eller utformingskrav for bruene, eller av hensyn til utforming av terrenget rundt. Et studie utføres dermed for vinkelendringer om søylenes y-akse. Under studiet gjøres det ikke endringer av original lengden til søylene. Dersom søylenes lengde skal forlenges på grunnlag av vinkelendring må Dynamoscriptet modifiseres, da scriptet er primært oppbygd rundt vinkelrette søyler. Likevel foreligger det, som nevnt tidligere, parameter hvor søylelengde endres.

Det er valgt å vinkle de midtre søylene om y-aksen. Rett søyle er gitt av 0° , største vinkel er 60° fra rett søyle. Dynamoskriptet der en slik endring kan gjennomføres er gitt i figur 7.17.

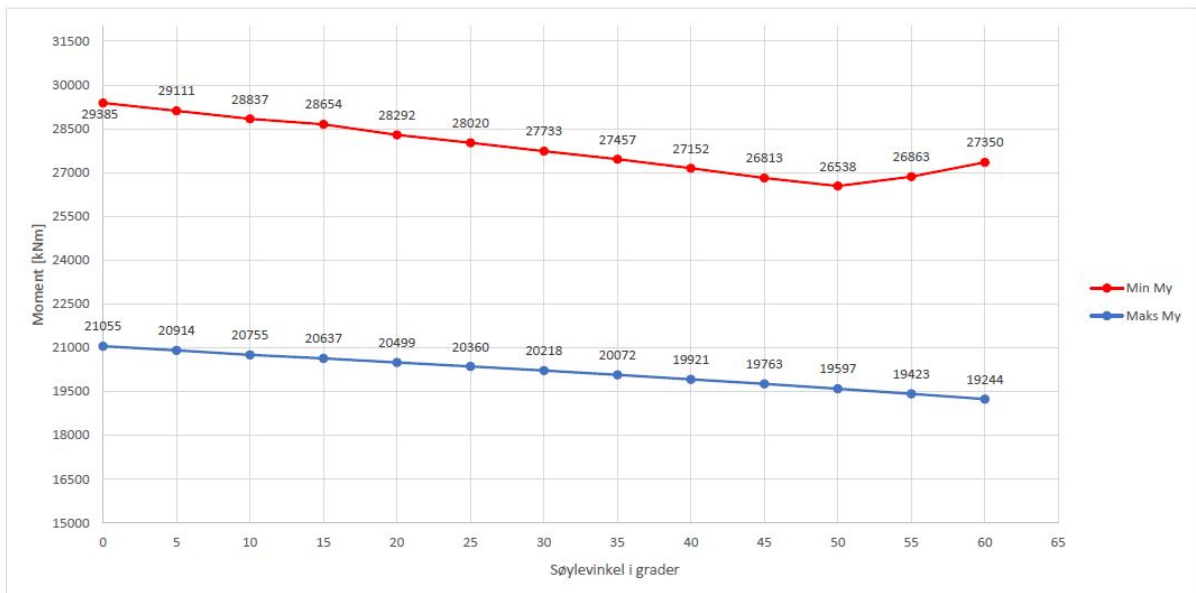


Figur 7.17: Vinkelendring i Dynamo

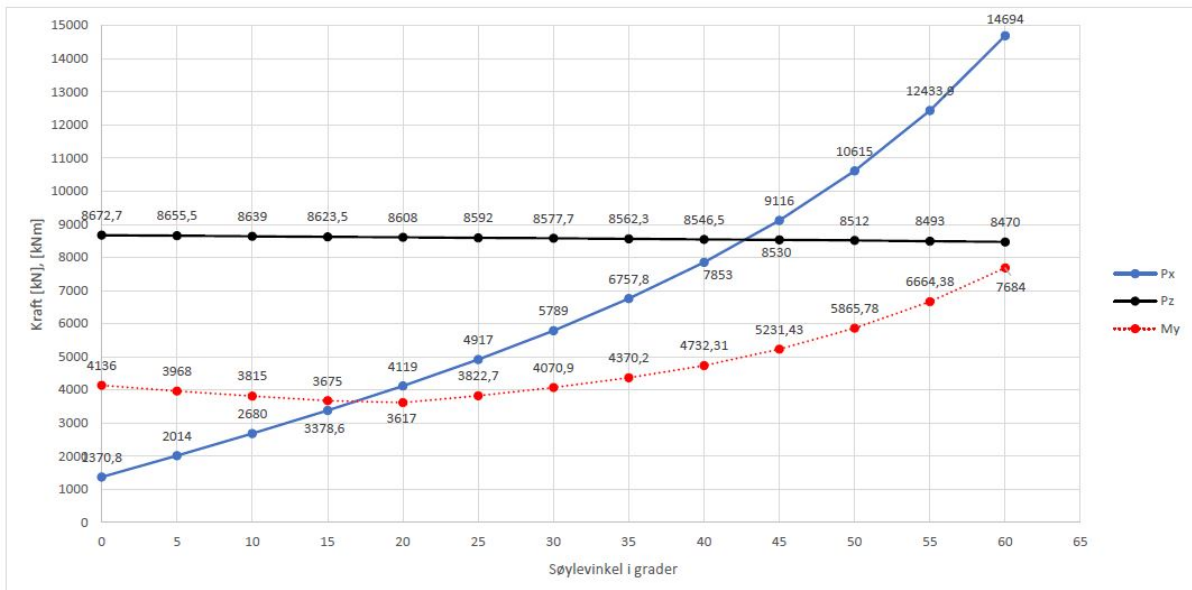
Arbeidsomfanget for endring av vinkler krever få operasjoner. Det starter med endring av vinkelgrad i skriptet etterfulgt av nye generering. Deretter følges lik prosedyre som nevnt tidligere, med overføring og analyse i Sofistik.

Fokuseres det på moment over midtre søyle, vil minste verdi være med en søylevinkel på 50° , se figur 7.18. Endring i feltmoment vil være tilnærmet lineær fra rett søyle, til 60° vinklet søyle. Verdier tilknyttet deformasjon utgjør tilnærmet ingen forskjell ved vinkelendring av midtre søyler.

7. PARAMETRISK STUDIE

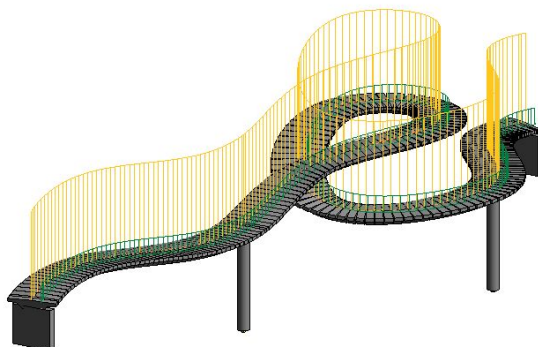


Figur 7.18: Hvordan vinkelendring i midtre søyler påvirker momentet, gitt som absoluttverdier



Figur 7.19: Maks opplagerbetingelser ved vinkelendring av søyle

7.8 Studie 8 – Kompleks geometri



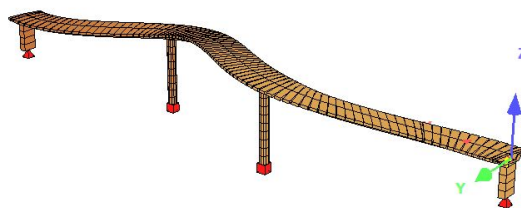
Figur 7.20: Kompleks bruutforming med laster

Studie 8 tar i større grad for seg mer kompleks brugeometri. Dette illustrerer hvor universal en parametrisk modell kan være. Selve senteraksen til bruene tilknyttet flere punkter for å skape en litt annerledes geometri. Søyleplasseringen endres og de midtre rektangulære søylene kan eksempelvis erstattes med sirkulære. Det parametriske tverrsnittet kan enkelt forandre geometri slik vist tidligere. Den da mer robuste enkle brukonstruksjonen kan alternativt få form av en mer komplisert gangbru. Det bør nevnes at det er ikke tatt konstruktive hensyn under dette studiet. Studiet gjøres for å vise kapasiteten til en parametrisk modell. Brumodellene i studiet er derfor ikke særlig egnet fra et konstruktivt ståsted.

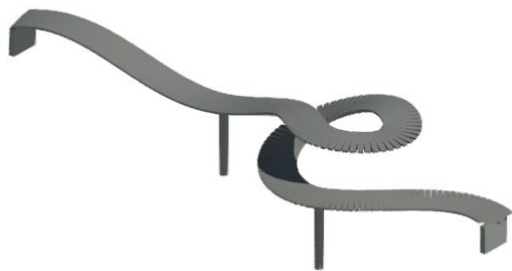
Geometriscriptet sorterer punkt gjennom kurvens x-verdier for å skape nødvendige rette bjelkeelementer. Dersom brudekket skal rotere tilbake må sortering av punkt endres. Endringen bør gjøres slik at punktene blir sortert etter en kurve. Noden anvendt til dette er *Points.SortPointsAlongCurve*. Bruen kan da i større grad følge en helt vilkårlig kurve, da denne kurven blir delt inn i små rette elementer.



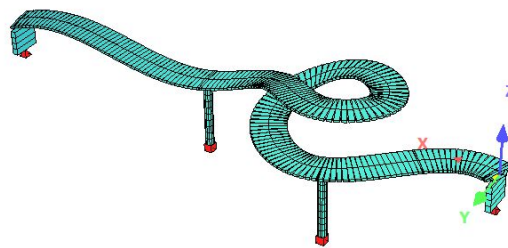
Figur 7.21: Svingende gangbru i Revit



Figur 7.22: Svingende gangbru i Sofistik



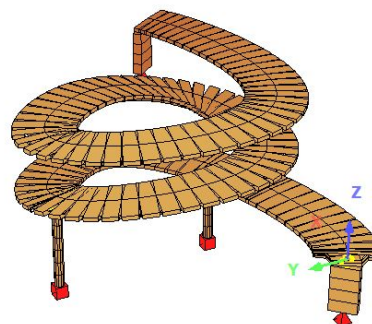
Figur 7.23: Kompleks gangbru i Revit



Figur 7.24: Kompleks gangbru i Sofistik

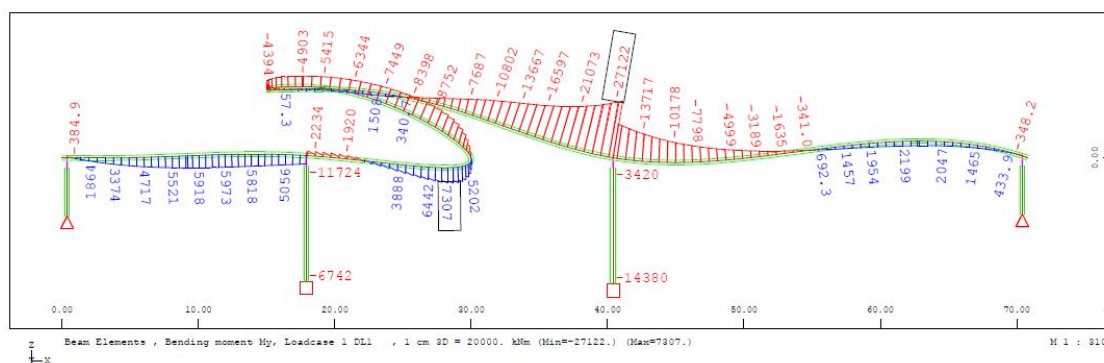


Figur 7.25: Kompleks gangbru i Revit 2



Figur 7.26: Kompleks gangbru i Revit 2

Å legge til laster er svært enkelt ved hjelp av skriptet for laster i Dynamo. Figur 7.20 illustrert hvor godt utfallet av lastscriptet er. Den analytiske delen i Revit kan da overføres til Sofistik for analyse. Geometriskriptet er primært bygd for fire søyler, dersom flere søyler skal legges til i geometrien må scriptet modifiseres en viss grad. Det er derfor kun benyttet fire søyler i dette studiet.



Figur 7.27: Eksempel på momentdiagram om y-aksen for sirkulær gangbru, kun egenvekt.

8 Diskusjon

Gjennom oppgaven har målsettingen vært å opprette en arbeidsflyt hvor parametrisk design blir tatt i bruk til å effektivisere bruprojektering. En optimal arbeidsflyt bidrar til etablering av en beregningsmodell, samt tegningsmodell med god kvalitet uten store utfordringer. Denne delen vil i større grad diskutere kvaliteten av arbeidsflyten utformet i oppgaven, bruken av parametrisk design og hvorvidt en slik metode er egnet for bruprojektering.

8.1 Arbeidsflyt

Opgaven har forsøkt å etablere en parametrisk modell med funksjonalitet både som tegnings- og beregningsmodell. Tegningsmodellen er ikke spesielt nøyaktig sammenlignet med ambisjonene på grunn av måten modellen lages. Årsaken til dette er at overgangen mellom Revit og Sofistik krever analytiske element.

Bjelkeelementene kan kun krumme i et plan av gangen og følgelig vil bruen ikke ha mulighet til å svinge samtidig med høydeforskjell langs veilinjen, uavhengig av størrelsesorden. En forenkling på dette problemet er å dele bruen opp i flere korte, rette analytiske element. Dersom inndelingen er tett nok vil elementene kunne lage både høydeforskjell og krumning langs veilinjen. Problemet ved å dele opp bjelkeelementene i små rette deler er at det vil oppstå åpninger mellom elementene i konstruksjonen, dersom veilinjen er kurvet. Tegningsmodellen mister av den grunn mye nøyaktighet. Samtidig er bjelkeelementene linjebasert, dette medfører at lokal rotasjon av elementets x-akse ikke lar seg gjøres på en akseptabel måte. En slik lokal rotasjon fører til «høydehakk» mellom elementene. Helningsvinkelen til tverrsnittet i tverretning vil derfor være konstant gjennom hele veilinjen. Forenklingen skaper med andre ord problemer i Revitmodellen, hvor mange rette element ikke representerer veilinjen korrekt, samt at Revitmodellen ikke er tilstrekkelig nøyaktig til uthenting av arbeidstegninger. En alternativ forenkling er å la bruen kun ha krumning i et plan av gangen. Det betyr at bruen enten svinger i xy -planet eller har høydeforskjell i xz -planet. Da er det mulig å unngå oppdelingen av analytisk elementer, men igjen vil ikke Revitmodellen gi et korrekt bilde av bruen basert på tverrsnittsrotasjon og realiteten.

Flyten vist i figur 1.1 innledningsvis fungerer dermed ikke ideelt, grunnet disse forenklingene. Den optimale måten bruen modelleres i Revit, der tegningsmodellens nøyaktighet og kvalitet antageligvis vil være tilfredsstillende, er ved anvendelse av adaptive familier. Tas denne type familien i bruk kan man knytte et tverrsnitt opp mot punkter, eksempelvis senterlinje, høyre gangfelt og venstre gangfelt, i Dynamo. Bruens geometri kan deretter etableres ved å «ekskludere» tverrsnittene sammen til en monolittisk brumodell uten «åpninger». Dette gjør at bruen kan svinge og ha høydeforskjeller samtidig med at tverrsnittet kan endres langs veilinjen. Problemet er denne familien ikke har en analytisk modell og dermed hindres overgang til Sofistik.

Begrensninger i Revit forårsaker problemer det er vanskelig å komme rundt. Det er av den grunn uunngåelig å ikke foreta forenklinger i arbeidsflyten. Revit representerer dermed mye av svakheten, mens på den andre siden så representerer Dynamo styrken i arbeidsflyten for parametrisk design. Gjennom Dynamo kan det meste styres, endres og kontrolleres i Revit,

og det på en effektiv måte. Geometri lages enkelt med linjer og punkter som overføres til elementer i Revit. Elementene i Revit kan samtidig lett endres eller automatiseres gjennom å bestemme parametere i Dynamo. Herunder tverrsnittsdimensjoner, materiale, fastholdinger for bruelement og søyler, laststørrelser osv.

På en annen side så er det ikke alltid Dynamo jobber slik man vil. Det kan oppstå utfordringer når Dynamascript åpnes på nytt. Dette innebærer at nodene som lager bruelement ikke alltid oppdaterer eksisterende element, men heller lager nye med samme plassering. Det eksisterer flere måter å løse denne utfordringen på og det er en av fordelene og styrkene med Dynamo. Evnen Dynamo har til å hente og operere automatisk på element fra Revit er god. I flytmetode IV endres søylene ved å hente de enkelt fra Revit til Dynamo, og deretter gi de ny plassering gjennom Dynamo. Lastgenereringen løser dette på en mer effektiv måte ved å slette eksisterende laster og automatisk lage nye.

Lastskriptet utforming gjør bruedesign og endring av lastbildet høyst effektivt. Det er erfart at endringer gjennom Sofiplus kan være tidkrevende, da man manuelt må endre lasten på hvert element. Gjennom Dynamoskriptet for laster krever lignende operasjon kun endring i laststørrelse, samt en ny «generering» av skriptet. På bakgrunn av at lastskriptet er bygd opp rundt Revit elementene, vil en endring utføres i alle elementdelen. Dette er utvilsomt effektiviserende for opprettelse eller endring av linjelaster. Skriptet for laster er likevel ikke perfekt. Endringer av geometri kan føre til problemer for linjelastene, dersom det legges inn en krumning for bruene. Linjelaster følger automatisk elementene i bruene, men det gjør ikke de eksentriske linjelastene. En ny oppdatering av lastskript må dermed utføres for å få korrekt lastbilde.

Et annet relevant punkt er at Dynamo ikke inneholder alle noder nødvendig for å gjøre det man vil. Løsningen på dette er imidlertid å programmere egendefinerte noder gjennom Python- eller Csharp-skript til bruk i Dynamo. Dette kan for eksempel være noder som lager last i Revit, eller akse til Sofistik. I tillegg finnes det nedlastbare pakker i Dynamo, hvor andre dynamobrukere har lagt ut sitt arbeid. Grunnlaget for egendefinerte noder er RevitAPI hvor man kan programmere Revit sitt brukergrensesnitt.

Dynamo har en god link til opprettelsen av gridlines i Revit, slik at denne metoden fort kunne vært mer brukervennlig enn den valgte metoden med egendefinert node. Aksene vil i tillegg bli synliggjort i Revit og med det enklere å utelukke feil.

Revitmodell

Fordeler

- Dynamo oppretter og endrer geometri på en effektiv måte
- Lastgenerering gjøres svært effektivt, sammenlignet med tradisjonell metode
- Plassering for linjelaster tilknyttet elementer oppdateres automatisk ved geometriendringer
- Enkelt å bestemme verdier for parametere i modellen gjennom Dynamo
- Geometrien synliggjøres i en 3D-modell
- Flere endringer kan automatiseres, eksempelvis fastholdning til elementer

Ulemper

- Element som følger en gitt veinje gir kun bjelkeelement i den analytiske modellen
- Bjelkeelement håndterer krumninger i to plan av gangen dårlig
- Oppdeling av bruene i korte rette element medfører mindre nøyaktighet for tegningsmodellen
- Dynamo oppdaterer ikke alltid eksisterende element i Revit
- Lasttilfeller, opplagerbetingelser og analytiske linker må legges til manuelt i Revit
- Plassering for linjelaster som ikke følger elementer oppdateres ikke automatisk ved geometriendringer

For å tilfredsstillere begrensninger i Revit sine analytiske bjelkeelementer blir nevnte dobbeltkrummet linjer erstattet med rette linjer. Denne forenklingen utgjør ingen nevneverdig forskjell i FEM-modellen etablert i Sofistik. Bestemmelse av hvilken type beregningssystem, størrelse på mesh og konstruksjonsklasse er blant ting det legges til uten problem i Revit-Sofistik overgangen. I tillegg vil overføringen automatisk beregne tyngdepunkt av tverrsnittet, slik Sofiplus gjør ved design av eget tverrsnitt. Overgangen fungerer samtidig svært bra for linjelaster og egenlast, hvor egenlasten blir automatisk beregnet gjennom overføringen. Generelt vil overføring av den analytiske modellen skje uten problem og er direkte klar for bearbeiding i Sofistik.

Likevel er det ikke en feilfri overgang mellom Revit og Sofistik. Mye av grunnen til dette er begrensninger i Revit. Nevnt tidligere er Revit designet for bygninger og dette tydeliggjøres enda bedre ved innsettelse av laster. Temperaturlast kan ikke etableres direkte i Revit og da heller ikke i Dynamo. Det ble derfor designet egen node for generering av tekstfil i Teddyformat. Temperaturlast ble deretter lagt inn i Sofistik ved hjelp av teksteditor. Selv om dette gjør flyten mindre bra, trengs den kun gjøres en gang så lenge verdier stemmer overens. Dette gjelder også for trafikklast. Gjennom oppgaven er det benyttet «traffic loader» i Sofistik for å legge inn og beregne trafikklast. Dette er gjort fordi Revit ikke har en slik funksjon. Traffic loader trenger en akse langs profilet for å beregne trafikklast, noe overgangen ikke automatisk lager. Sofiplus lager forøvrig heller ikke dette automatisk. En slik akse kan enten etableres ved hjelp av «gridlines» i Revit/Dynamo eller, slik anvendt i oppgave, med Teddyprogrammering.

Et annet punkt er måten Revit og Sofistik definerer hvor den analytiske bjelken skal føres i tverrsnittet. I Revit legges den analytiske bjelken i toppen av *structural framing* familien, mens i Sofistik vil tverrsnittet plasseres slik at den analytiske bjelken er i tyngdepunktet. Dette betyr at tegning- og beregningsmodellen ikke stemmer helt overens. Forskjellen kan føre til at lengdeforhold blir annerledes enn det tiltenkt, og dermed stivhetsforholdet i rammeanalysen.

En ambisjon i parametriseringen var å ha spennarmering gjennom hele prosessen, med en parametrisert visuell føring gjennom Dynamo og Revit. Da det ikke eksisterer en analytisk spennarmering i Revit er dette ikke gjennomførbart. Det ble derfor ikke mulig å danne en parametrisert kurve til spennarmeringen som ved overføring opptrer deretter. Med bruk av tilleggspakken SIX fant man ut at spennarmering får visualiseringen av vanlig armering, når modellen overføres fra Sofistik til Revit. Å oppnå lik visualisering er dermed mulig gjennom Dynamo, men derimot ikke oppnåelig å tilknytte armeringen til den analytiske modellen. Den valgte alternative metode for opprettelse av spennarmering i beregningsmodellen, ble med hjelp av Teddyfil, for å unngå design i to tegningsprogram (Revit og Sofiplus). Metoden fungerer bra, men er mindre brukervennlig enn en visuell tilnæringsmåte.

Materialene i Revit stemmer nødvendigvis ikke overens med materialer i Sofistik. Dette innebærer om eksempelvis B45 er valgt i Revit bør det kontrolleres i *materials*, under Sofistik analyse, at Revitmateriale blir representert av ønskelig material i Sofistik. Dette er ikke spesielt vanskelig og tar ikke mye tid, men det gjør overgangen mindre glatt enn ønskelig. Avslutningsvis er endring av tverrsnittsgeometrien ikke ideell. Dersom tverrsnittsgeometri endres i Dynamo/Revit så må disse elementene først slettes i Sofistik før de oppdaterte elementene overføres til Sofistik. Dette er en klar begrensning i overgangen fra Revit til Sofistik. Til sammenligning skjer denne endringen automatisk fra Dynamo til Revit og ingen profil må dermed slettes i Revit før en kan gjøre endringer i Dynamo. Overgangen mellom Dynamo og Revit foregår med andre ord i sanntid, mens mellom Revit og Sofistik er dette ikke tilfellet siden beregningsmodellen genereres på nytt hver gang.

Revit-Sofistikmodell

Fordeler

- Geomeri, forbindelser, opplagere og laster overføres enkelt.
- Egenlast beregnes automatisk ved overføring.
- Revit kan bestemme konstruksjonsklasse og mesh for beregningsmodell.

Ulemper

- Forskjell i plassering av analytisk bjelke fra tegningsmodell til beregningsmodell.
- Ikke mulig å opprette trafikk og temperaturlast i Revit.
- Profiler må slettes i Sofistik ved endring.
- Spennarmring ikke mulig i Revit.
- Materialer stemmer ikke direkte overens.

Arbeidsflyten krever at man går litt frem og tilbake mellom Dynamo og Revit. Dette innebærer blant annet at lastkombinasjoner må legges manuelt til i Revit, før en deretter kan gå inn i Dynamo og legge til laster. Videre eksempel er at geometri lages i Dynamo, før opplagerbetingelser må legges til manuelt i Revit. På lignende måte må flere oppgaver legges til i Sofistik etter at modellen overføres fra Revit til Sofistik. Dette innebærer nevnte trafikk- og temperaturlast, akse for veilinje og informasjon for hvordan beregninger utføres. De fleste av disse operasjonene må til gjengjeld kun gjøres én gang. Resultatet av at diverse input legges til i flere ledd underveis gjør likevel arbeidsflyten mindre optimal, da det er ønskelig at all input foregår i Dynamo. Implisitt betyr dette at arbeidsflyten ikke er spesielt glatt og funksjonell, sammenlignet med målsettingen.

8.2 Parametrisk design

Parametrisk design gjennom Dynamo gir gode muligheter til å lage mer kompleks geometri i Revit, enn Revit har alene. Geometrien bestemmes ved å kode et sett med instruksjoner kan man fortelle programvaren hva den skal gjøre. Geometrien kan dermed etableres med logikk, og ikke ren tegningsmodellering. Instruksjonene er ofte enkle å forstå, men kompleksiteten av den geometriske utformingen kan være vanskelig å forutse. Å la datamaskinen løse utfallet av instruksjonene, fjerner begrensningene mennesker har til å visualisere nevnte logikk [31].

Standarder og håndbøker stiller mange krav til konstruksjoner og det er derfor nødvendig å utforske mange alternativer for å overkomme disse kravene. Den parametriske modellen gir mulighet til å generere og utforske et stort antall utforminger og design av bruene på en effektiv og produktiv måte. Dette er grunnen til at arbeidsflyten egner seg så godt for den konseptuelle fasen. Visuell programmering er et sterkt verktøy for nettopp dette. Parallelt med at den parametriske modellen kan generere et stort antall utforminger, så visualiseres disse direkte i Revit. Resultatene må fortsatt kalkuleres, men ingeniøren kan gjøre mange vurderinger på bakgrunn av hvordan modellen ser ut.

Ved å teste flere forskjellige utforminger kan man finne en mer ideell form på en effektiv måte. Den parametriske modellen gjør det enkelt å endre mange parametere i modellen uten at det krever spesielt mye arbeid. Største delen av tidsbruken går til rekalkulering av modellen og dette forekommer uavhengig av om man modellerer tradisjonelt eller med parametrisk design.

Laster og lastkombinasjoner kan skape problemer for parametrisk design. Når modellen endres, må nødvendigvis lastene og lastkombinasjonene også endres. Problemet her er at flere laster regnes på bakgrunn av hvordan geometrien på bruene er. Det er opp til ingeniøren å vurdere om disse må beregnes på nytt eller at en forenklet kan bruke lastene og kombinasjonene først beregnet. Dersom lastene beregnes på nytt er fordelen med Dynamo at de enkelt legges inn ved å kun endre et tall, som beskrevet i kapittel 6.1.1 om laster.

Det er også flere punkter ved brumodellering hvor parametrisering nesten ikke har latt seg gjøre. Dette innebærer temperatur og trafikklast, spennarmering og opplagerbetingelser.

Ulempen med parametrisk design er at mye av arbeidet blir dyttet over på startfasen av modelleringen. Den største tidsbruken går til utvikling av parametere og koder for å lage parametriske modeller. Det kan i tillegg være krevende å lære seg hvordan man arbeider med parametrisk design.

På en annen side er fordelen at mindre tid blir brukt på endringer av modeller, da dette nå blir gjort ved å endre på noen få parametere. Dette muliggjør opprettelse av nærmest ubegrenset utforminger på relativ kort tid, og dette gir mer frihet til å utforske, beregne og til slutt tilpasse designet av konstruksjonen. Dette er til forskjell fra den tradisjonelle metoden hvor store deler av arbeidstiden går til modellering.

Oppsummert Arbeidsflyt

Fordeler

- Egner seg best for den konseptuelle fase.
- Enkelt å gjøre endringer med parametriske modeller.
- Mange utforminger testes raskt.
- Gir mulighet til å tegne mer kompleks geometri.

Ulemper

- Ikke alle deler av modellering lar seg parametrisere.
- Laststørrelser som er avhengig av bruens utforming må egentlig regnes på nytt. Dette er en vurderingssak.
- Mye arbeid flyttes til startfasen for å lage parametere og koder.

8.3 Konklusjon

Arbeidsflyten i kapittel 6 viser at det er mulig å bruke parametrisk design til å etablere både beregnings- og tegningsmodell. Likevel bærer arbeidsflyten preg av tvunget forenklinger og alternative løsninger, mye på grunn av begrensninger i Revit. Det at bruene må oppdeles i rette elementer for å tilfredsstille en analytisk modell reduserer kvaliteten til tegningsmodellen. Følgelig egner ikke tegningsmodellen seg som grunnlag for arbeidstegninger. Selve arbeidsflyten fungerer godt, men ettersom Revit mangler essensielle elementer tilknyttet bruprojektering må diverse inputter gjøres i forskjellige steg. Den parametriske modellen uttrykker derav mindre fullstendighet og arbeidsflyten blir mer oppstykket.

Likevel egner arbeidsflyten seg godt for den konseptuelle fasen i et bruprojekt, hvor den parametriske modellen gir mulighet til å beregne mange alternative utforminger raskt og effektivt. Kapittel 7, om parametrisk studie, demonstrerer at den parametriske modellen både er fleksibel og funksjonell. Nettopp denne fleksibiliteten gjør arbeidsflyten så anvendelig for konseptfasen.

Arbeidsflyten oppnår en god analytisk modell, hvor det er rom for små programmessige utbedringer der flyten bedres. For å oppnå en tegningsmodell med tilstrekkelig kvalitet bør større utbedringer i programvaren forekomme, eller ny arbeidsflyt opprettes. Dermed konkluderes det med at programvarene ikke samarbeider godt nok til å realisere målsetningen om en parametrisk modell for effektivisering av bruprojektering, hvor det fremmes en komplett tegnings- og beregningsmodell.

9 Videre arbeid

Det er momenter man kan videreutvikle for å oppnå en mer ideell arbeidsflyt. Flere av begrensningene i arbeidsflyten stammer fra Revit, av den grunn kan nye forbedringer i denne fasen skape en god metode for brudesign. Primært ligger utbedringene hos utvikler, likevel beskrives muligheter for videre arbeid.

- Utvikle script i Dynamo som lager opplager og analytiske linker i Revit.
- Løse problemet med at Dynamo ikke alltid finner eksisterende elementer når Structural Framing og linjelaster legges inn. Metoden for linjelaster fungerer godt, men for strukturelle element er ikke metoden optimal.
- Legge til laster og lastkombinasjoner i Dynamo.
- Utforske mulighetene med å overføre adaptive familier til beregningsprogrammer. Alternativt er det å finne en annen metode å overføre tegningsmodellen til beregningsprogrammer.
- Utforske mulighetene for å benytte andre programmer for samme arbeidsflyt. Forslag til programmer er det visuelle programmeringsprogrammet Grasshopper og modelleringsprogrammet Rhino. Videre kan også andre beregningsprogrammer som FEM-design eller Autodesk Robot testes for arbeidsflyten.
- Utforske mulighetene til å bruke andre element enn bjelkeelement ved overføring fra tegningsmodell til beregningsmodell, herunder skallelement og solidelement. Det er fortsatt avgjørende at bruene kan følge veilinjer når den modelleres.
- Utforske muligheter til å bruke modellen for andre type bruer. Fagverksbru er en type hvor modellen sannsynligvis fungerer veldig bra, ettersom modellen så langt er begrenset til bjelkeelement.
- Det er også interessant å teste ut den parametriske modellen for bygninger, hvor endringene av skript vil hovedsakelig være å lage nye linjer til bjelker og søyler. Skript for dekker og vegger må også lages, men dette medfører ikke store vanskeligheter.
- Sofistik har et verktøy for armering i Revit, kalt Sofistik Reinforcement Dette verktøyet er det under oppgaven ikke lagt vekt på, men her foreligger det muligheter deriblant oppdele armering «lagvis». Det blir da tilrettelagt for at Sofistik kan beregne nødvendig armering.

Referanser

- [1] Regjeringen, «Politisk plattform for en regjering utgått av Høyre, Fremskrittspartiet og Venstre.», PDF, 14. januar 2018. side: <https://www.regjeringen.no/no/dokumenter/politisk-plattform/id2585544/#k15>.
- [2] D. Nagy, «Introduction to Computational Design», feb. 2017, [Online]. side: <https://medium.com/generative-design> (sjekket 12. apr. 2018).
- [3] D. Mackenzie, «Parametric + algorithmic design», feb. 2016, [Online]. side: <http://architecturenow.co.nz/articles/parametric-algorithmic-design/#img=0> (sjekket 12. mai 2018).
- [4] Warren og Mahoney, «Te Whittinga», u.å, [Online]. side: <https://warrenandmahoney.com/portfolio/te-whitinga> (sjekket 12. mai 2018).
- [5] Autodesk, «BIM and the Future of AEC», u.å, [Online]. side: <https://www.autodesk.com/solutions/bim> (sjekket 5. mar. 2018).
- [6] K. Bell, *Finite element analysis*. Fagbokforlaget, 2013.
- [7] Autodesk, «Revit Built for BIM», u.å, [Online]. side: <https://www.autodesk.eu/products/revit/overview> (sjekket 5. mar. 2018).
- [8] J. Morris, «Revit (Strucuture) Infracore 360», jun. 2016, [Online]. side: <https://www.autodesk.eu/products/revit/overview> (sjekket 3. jun. 2018).
- [9] Dynamo, «The Dynamo Primer», u.å, [Online]. side: http://primer.dynamobim.org/en/08_Dynamo-for-Revit/8-2_Selecting.html (sjekket 7. mar. 2018).
- [10] A. Hejnfelt, «From Revit to FEM», 2017, [Online]. side: <http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2017/dynamo-studio/bld125052#chapter=0> (sjekket 14. mar. 2018).
- [11] Dynamo, «The Dynamo Primer», u.å, [Online]. side: http://dynamoprimer.com/en/01_Introduction/1-1_what_is_visual_programming.html (sjekket 8. mar. 2018).
- [12] Autodesk.Help, «FAQ», Oktober 2017. side: <https://knowledge.autodesk.com/search-result/caas/CloudHelp/cloudhelp/2018/ENU/Revit-API/files/GUID-FB822689-505D-4DFE-8784-B24E3AFCC6F7-htm.html> (sjekket 27. mai 2018).
- [13] V. Ondejck, «Dynamo Graphic Standards at White arkitekter AB», Mai 2016, [Online]. side: <http://dynamobim.org/dynamo-graphic-standards-at-white-arkitekter-ab/> (sjekket 14. apr. 2018).
- [14] Sofistik, «Sofistik», [Online]. side: <https://www.sofistik.com/> (sjekket 1. feb. 2018).
- [15] —, *Tutorial SSD/SOFIPLUS - A Quick Rererence*, [Online], 2010. side: https://www.sofistik.com/uploads/tx_sofistik/Tutorial-SSD-SOFIPLUS-2010_1.pdf (sjekket 1. feb. 2018).
- [16] —, «CADINP - Text based input», u.å, [Online]. side: https://www.sofistik.eu/no_cache/solutions/structural-fea/cadinp/ (sjekket 1. feb. 2018).
- [17] —, «WINGRAF», 2017, [Online]. side: <http://www.sofistik.de/documentation/2018/en/tutorials/listoftutorials/general-workflows/wingraf.html> (sjekket 1. feb. 2018).
- [18] S. I. Sørensen, *Betongkonstruksjoner*, 2. utg. Fagbokforlaget, 2015.
- [19] Standard Norge, (2016), *Eurokode: Grunnlag for prosjektering av konstruksjoner*.
- [20] H. Johansen, «Beregningsveiledning for etteroppspente betongbruer», Statens Vegvesen, tekn. rapp., jan. 2017.

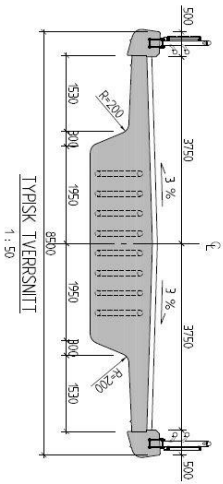
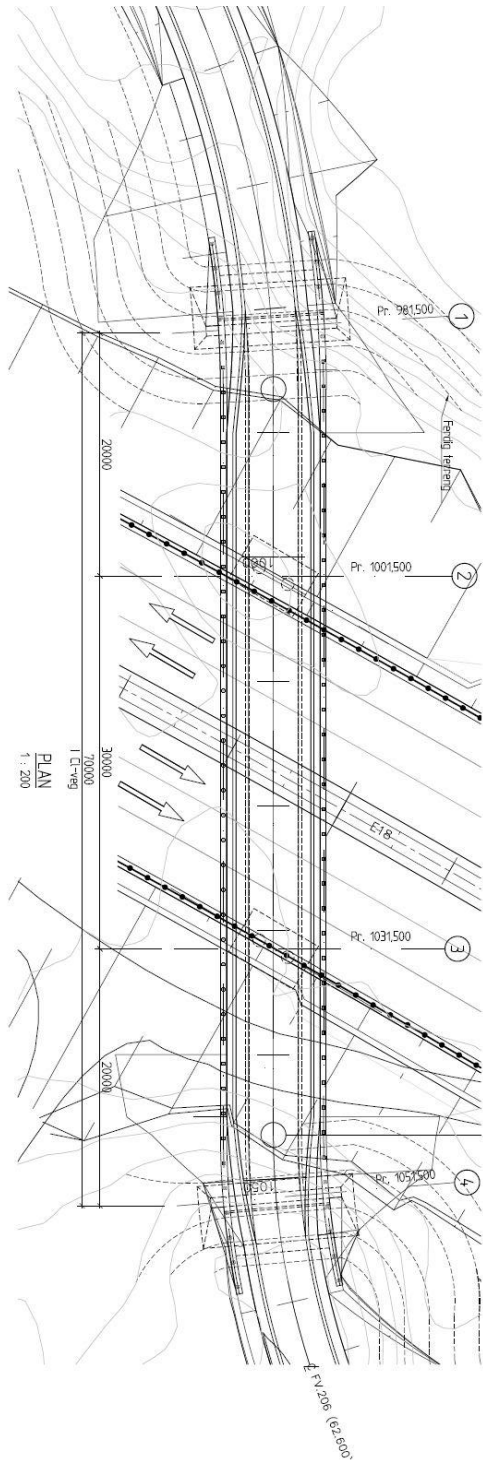
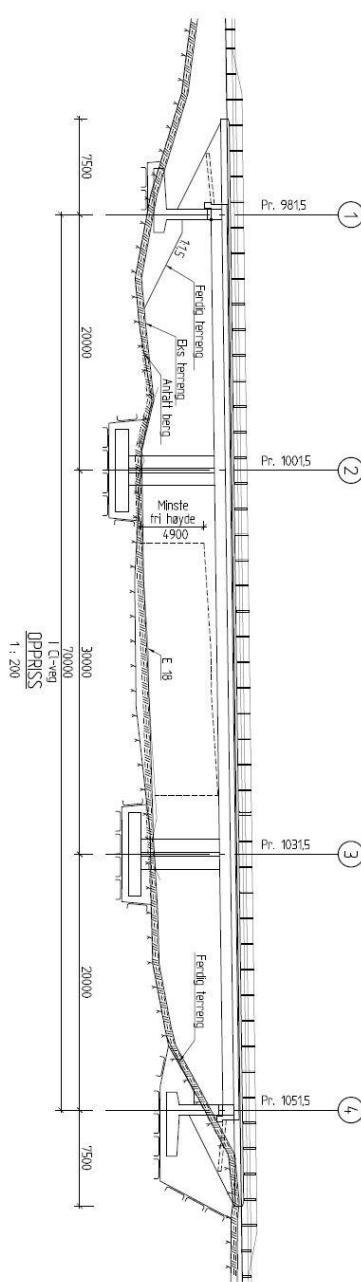
-
- [21] Standard Norge, (2008), *Eurokode 2: Prosjektering av betongkonstruksjoner - Del 1-1: Allmenne regler og regler for bygninger*.
- [22] «Håndbok N400 Bruprosjektering», Statens Vegvesen, tekn. rapp., jan. 2015.
- [23] Standard Norge, (2010), *Eurokode 1: Laster på konstruksjoner - Del 2: Trafikklast på bru*.
- [24] Standard Norge, (2009), *Eurokode 1: Laster på konstruksjoner - Del 1-4: Allmenne laster - Vindlaster*.
- [25] Standard Norge, (2008), *Eurokode 1: Laster på konstruksjoner - Del 1-5: Allmenne laster - Termiske påvirkninger*.
- [26] M. Engen, *Design of concrete shells, part I*, Forelesning, nov. 2018.
- [27] Spennteknikk, «TOBE FR4 potlager», BBR Netwrok, tekn. rapp., u.å. side: <http://www.spennteknikk.no/brosjyrer/TobeFR4-norsk-24022011.pdf>.
- [28] Sofistik, «Infrastructure Extension», u.å, [Online]. side: <https://www.sofistik.com/products/bim-cad/infrastructure-extension/> (sjekket 27. apr. 2018).
- [29] «Create Method», 2016, [Online]. side: <http://www.revitapidocs.com/2018.1/2e299798-26c8-9e42-47de-d496151d1e3d.htm> (sjekket 27. apr. 2018).
- [30] «Håndbok N100 Veg- og gateutforming», Statens Vegvesen, tekn. rapp., jun. 2014.
- [31] P. Jeffries, «What is Computational Design?», des. 2016, [Online]. side: <http://www.revitapidocs.com/2018.1/2e299798-26c8-9e42-47de-d496151d1e3d.htm> (sjekket 24. mai 2018).

Appendices

A Inspirasjonsbru

Grunnlag mottatt av COWI

PROFIL NR.	1000										1050		
HORI. KURVATUR	R=1												
BREDDETVIVDELSE	1:100m												
TIERNFALL	3,0%												
H.Kj.k.													
V.A.j.k.													
PROFIL H.	102,67	102,80	102,95	103,05	103,20	103,35	103,46	103,60	103,75	103,85	104,00		
TERRENG H.	99,92	97,24	96,36	96,05	96,21	96,67	96,96	97,41	100,68	103,54	103,50		



B Ekvivalente krefter

Beregning av ekvivalente krefter for å bestemme momentdiagram

$$f_{pk} := 1860 \frac{N}{mm^2} \qquad f_{p0.1k} := 1600 \frac{N}{mm^2}$$

$$k1 := 0.8 \qquad k3 := 0.95 \qquad k5 := 30 \qquad k7 := 0.75$$

$$k2 := 0.9 \qquad k4 := 50 \qquad k6 := 0.7 \qquad k8 := 0.85$$

$$\sigma_{p.m0} := \min(k7 \cdot f_{pk}, k8 \cdot f_{p0.1k}) = (1.36 \cdot 10^3) \frac{N}{mm^2}$$

$$A_p := 2100 \text{ mm}^2$$

$$P_{.m0} := A_p \cdot \sigma_{p.m0} = (2.856 \cdot 10^3) \text{ kN} \qquad \text{[EC1-1-1 (5.41)]}$$

$$P := P_{.m0} = (2.856 \cdot 10^3) \text{ kN}$$

$$e_m := 300 \text{ mm}$$

$$e_s := 0 \text{ mm}$$

$$L := 10 \text{ m}$$

$$\theta := 4 \cdot \frac{e_m}{L} = 0.12$$

$$q := \frac{(8 \cdot P \cdot e_m)}{L^2} = 68.544 \frac{\text{kN}}{\text{m}}$$

Parabel spennarmering

$$M_{ed} := q \cdot \frac{L^2}{8} = 856.8 \text{ kN} \cdot \text{m}$$

$$P_h := P = (2.856 \cdot 10^3) \text{ kN}$$

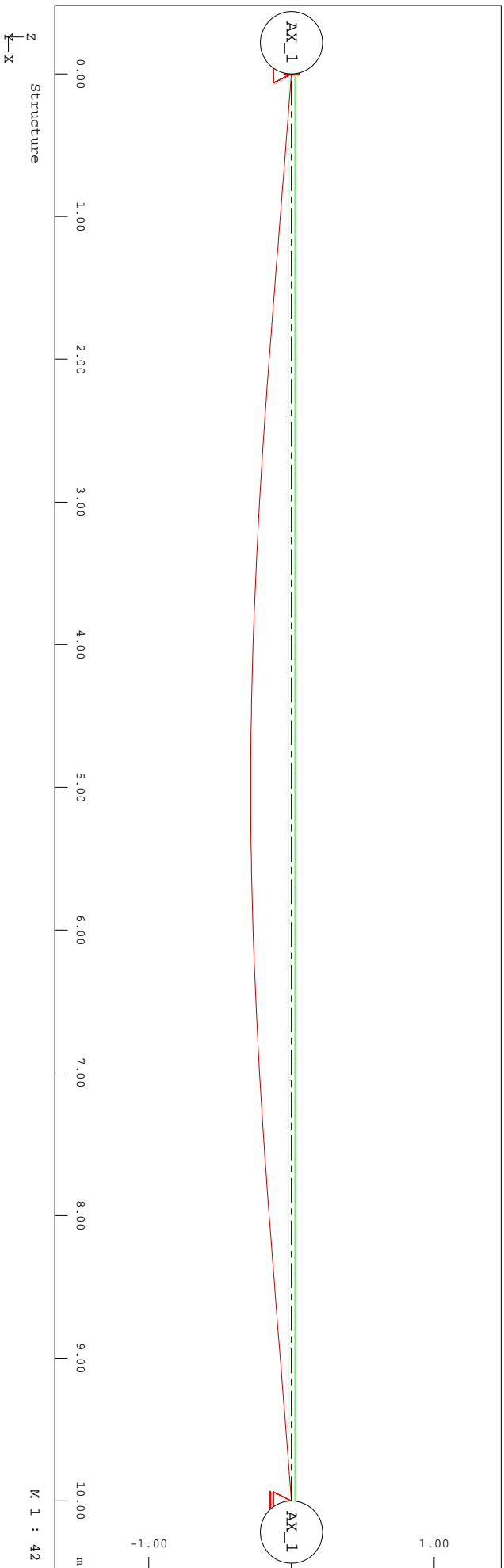
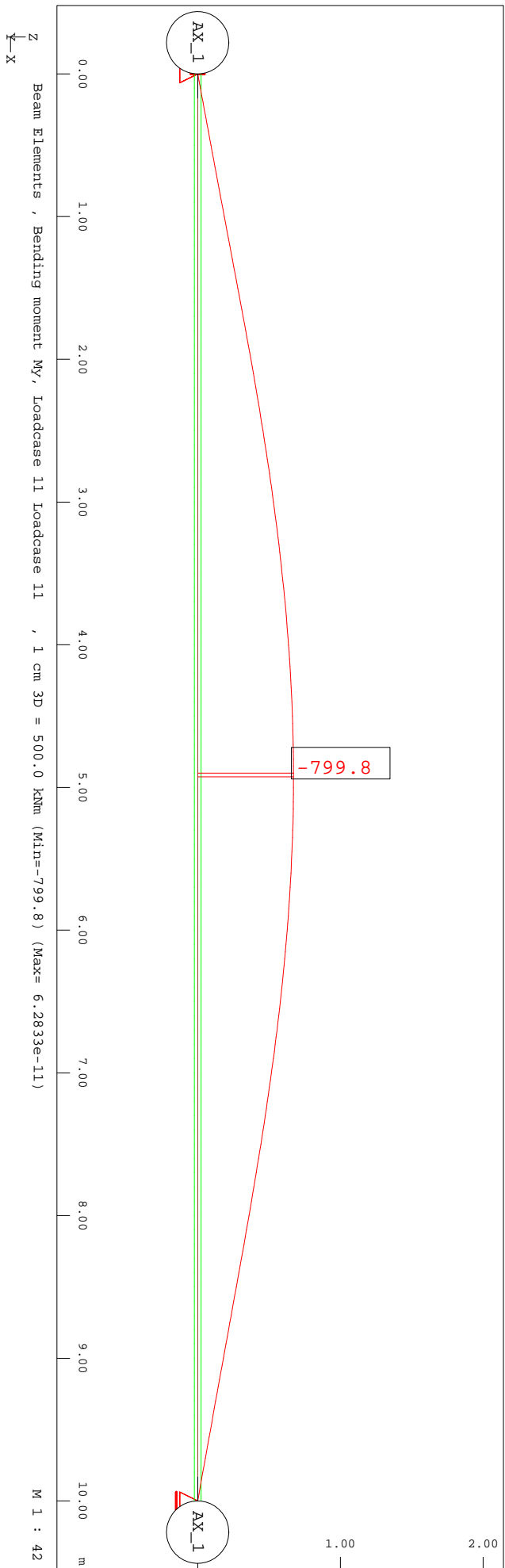
$$P_v := P \cdot \theta = 342.72 \text{ kN}$$

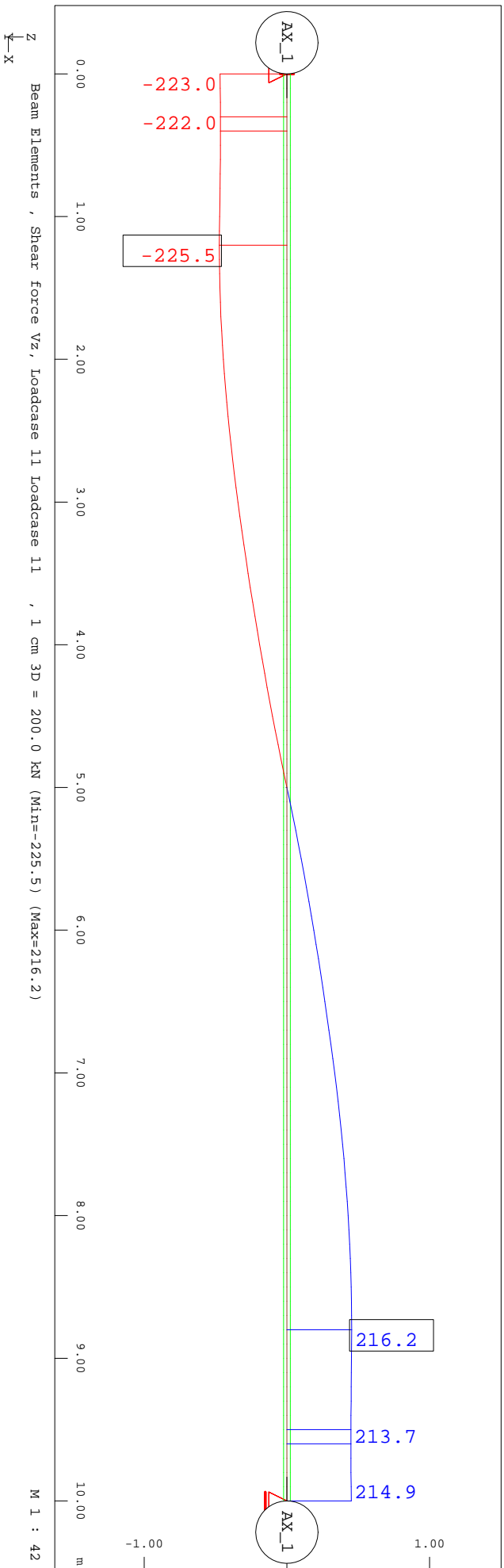
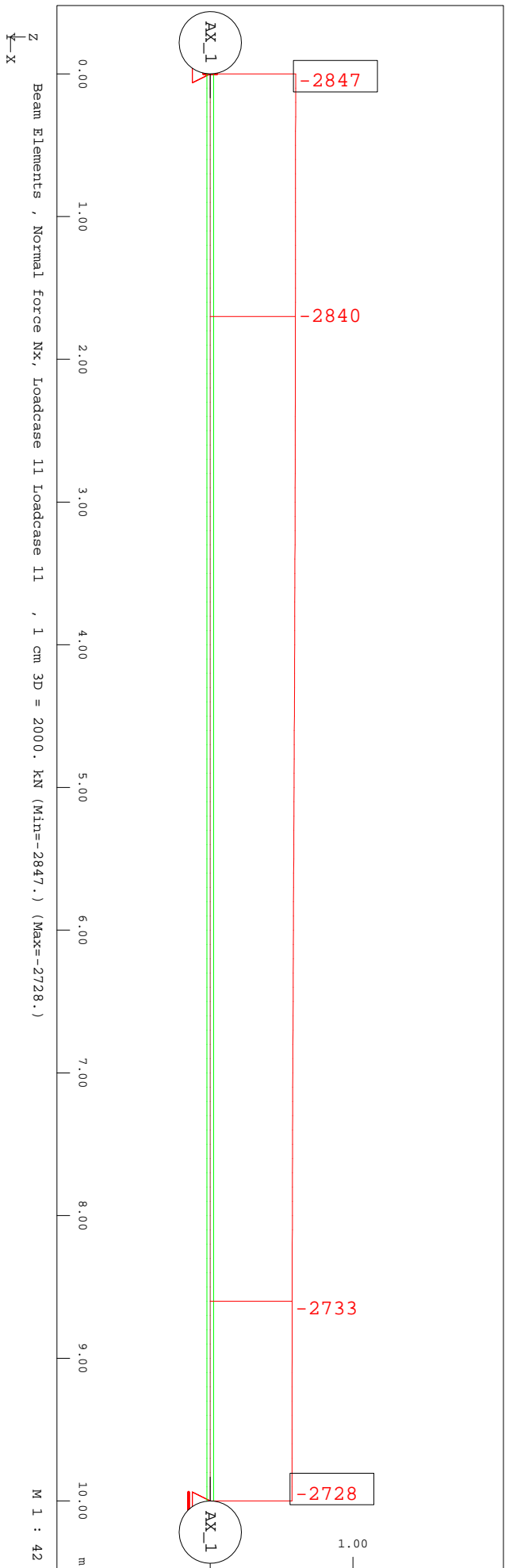
$$M := P \cdot e_s = 0 \text{ kN} \cdot \text{m}$$

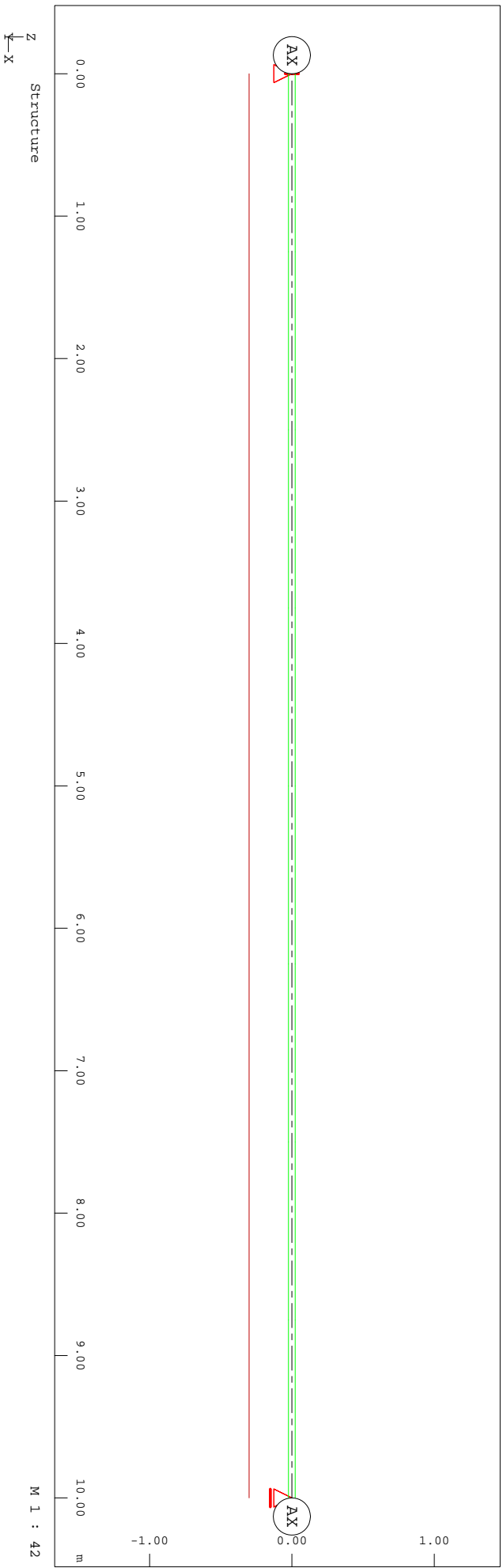
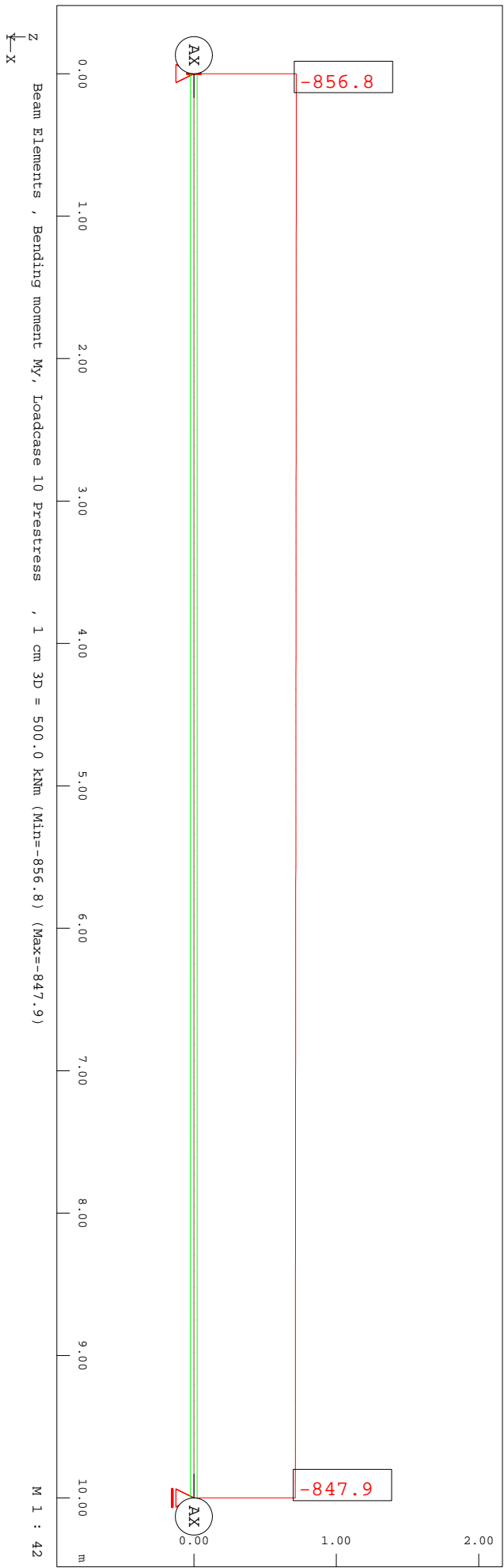
Rett armering

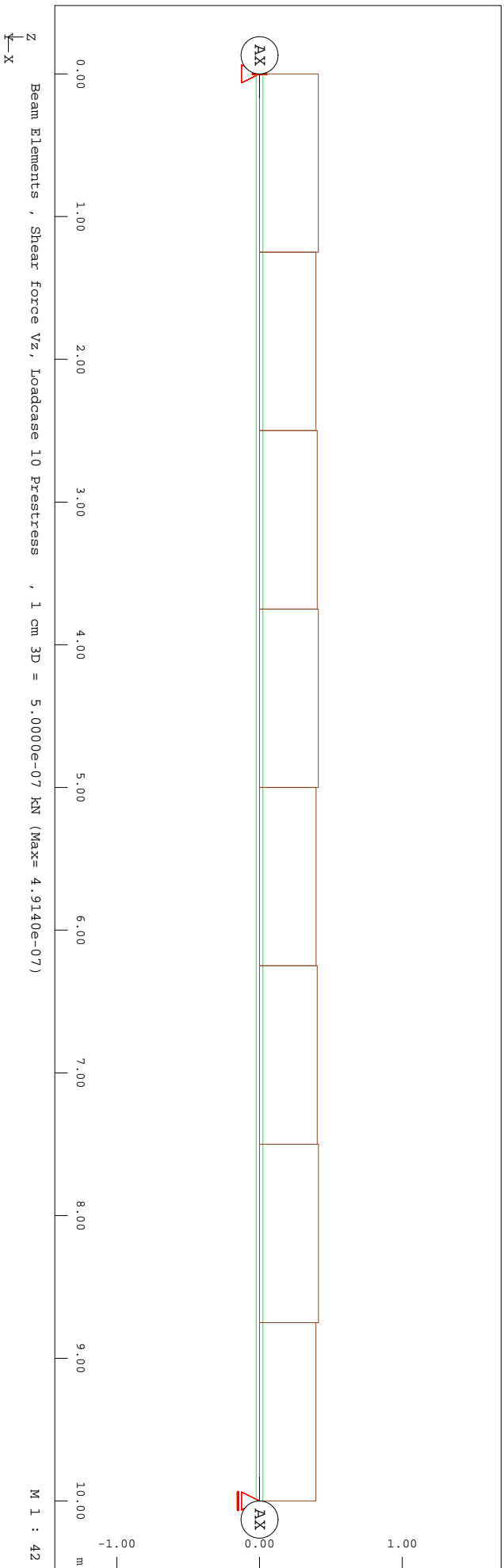
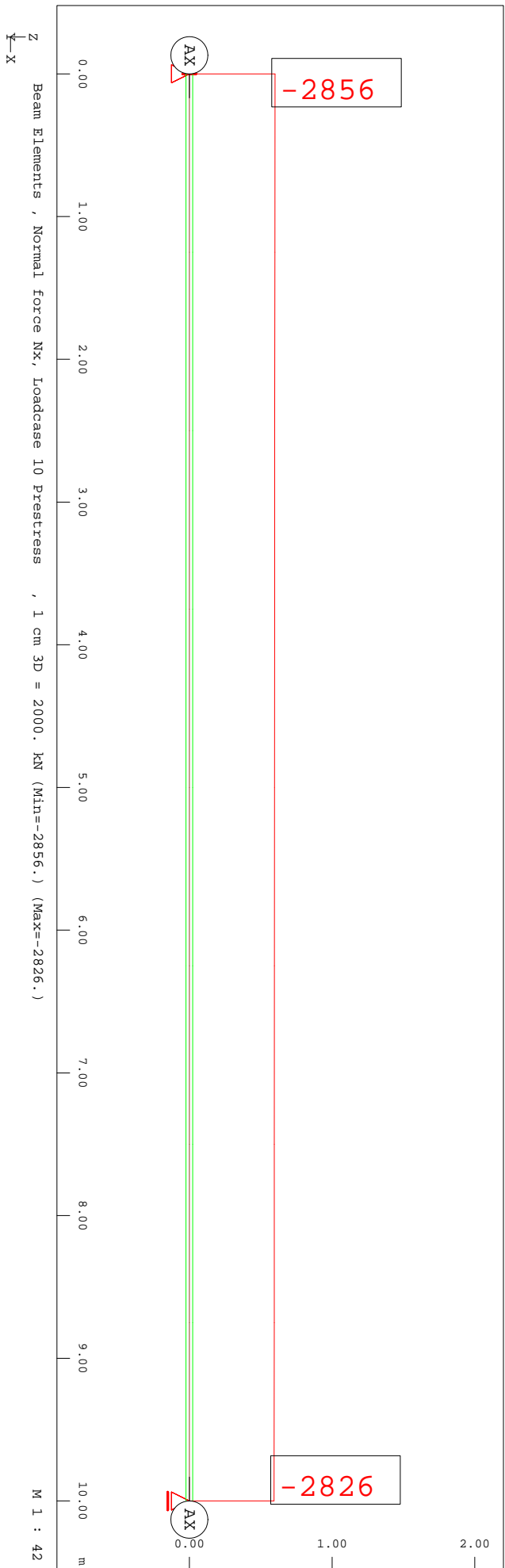
$$M := P \cdot e_m = 856.8 \text{ kN} \cdot \text{m}$$

$$N := P = (2.856 \cdot 10^3) \text{ kN}$$









C Egenlast laster

Egenvekt av kantdragere og belegning

Last fra kantdragere

Vekt armert betong

$$\gamma_b := 25 \frac{\text{kN}}{\text{m}^3}$$

Areal

$$A_{kant} := 297531 \text{ mm}^2$$

Last

$$g_{kant} := \gamma_b \cdot A_{kant} = 7.438 \frac{\text{kN}}{\text{m}}$$

Last gjelder hver enkelt kantdrager

Last fra belegning

Bredde av kjørebane hentet
fra Cowi-tegninger

$$b_{kjøreb} := 7.5 \text{ m}$$

Fra tabell i N400 om
belegningsvekt for største
spennvidde i bru

$$L_{spenn} := 30 \text{ m}$$

Største spennvidde $l < 50\text{m}$

$$g_{belegning} := 3.5 \frac{\text{kN}}{\text{m}^2}$$

Last fra belegning

$$g_{belegg} := g_{belegning} \cdot b_{kjøreb} = 26.25 \frac{\text{kN}}{\text{m}}$$

Total last:

$$g_{andre.laster} := g_{kant} + g_{belegg} = 33.688 \frac{\text{kN}}{\text{m}}$$

D Trafikklast

Trafikklast

NS-EN 1991-2:2003+NA:2010 Laster på konstruksjoner - Del 2: Trafikklast på bruer

4.2.1 Modeller for veitrafikklast

Tab 4.1

Bredde av kjørebane	$w := 7.5 \text{ m}$	$6 \text{ m} < w$
Antall "Notional" lanes	$n_1' := \frac{w}{3} = 2.5 \text{ m}$	$n_1 := 2$
Bredde av "Notional" lane	$w_l := 3 \text{ m}$	
Resterende bredde	$w - 3 \cdot n_1 \cdot m = 1.5 \text{ m}$	

4.3.2 Lastmodell 1

Korreksjonsfaktorer etter
NA.4.3.2

$$\alpha_{Q_i} := 1.0 \quad \text{for } i = 1, 2 \text{ og } 3$$

$$\alpha_{q_1} := 0.6$$

$$\alpha_{q_2} := 1.0 \quad \text{for } i > 1$$

$$\alpha_{qr} := 1.0$$

Laster fra tabell 4.2

Felt 1 akslinglast $Q_{1k} := 300 \text{ kN}$

Felt 2 akslinglast $Q_{2k} := 200 \text{ kN}$

Felt 1 jevnt fordelt $q_{1k} := 9.0 \frac{\text{kN}}{\text{m}^2}$

Felt 2/rest jevnt fordelt $q_{2k} := 2.5 \frac{\text{kN}}{\text{m}^2}$

Tandemsystem

Akslinglast felt 1 $\alpha_{Q_1} := 1.0$ $\alpha_{Q_1} \cdot Q_{1k} = 300 \text{ kN}$

Akslinglast felt 2 $\alpha_{Q_2} := 1.0$ $\alpha_{Q_2} \cdot Q_{2k} = 200 \text{ kN}$

Jevnt fordelt last

Jevnt fordelt last felt 1 $\alpha_{q_1} = 0.6$ $\alpha_{q_1} \cdot q_{1k} = 5.4 \frac{\text{kN}}{\text{m}^2}$

Jevnt fordelt last felt 2 $\alpha_{q_2} = 1$ $\alpha_{q_2} \cdot q_{2k} = 2.5 \frac{\text{kN}}{\text{m}^2}$

Jevnt fordelt last i rest $\alpha_{q_3} := \alpha_{q_2}$ $\alpha_{q_3} \cdot q_{2k} = 2.5 \frac{\text{kN}}{\text{m}^2}$

E Vindlast

Vindlaster

NS-EN 1991-1-4:2005+NA:2009 Laster på konstruksjoner. Del 1-4: Almenne laster - vindlaster

4.2 Vindhastighet og vindhastighetstrykk

NA.4.2 Grunnverdier

Tab. NA.4(901.1)

Referansevindhastighet vb.0
for Bamble kommune $v_{b,0} := 23 \frac{m}{s}$

Faktor for returperiode $c_{prob} := 1.0$

NA.4.2(2)P(901.1) Faktor for vindøkning med høyde over haver

Høyde byggeplass (antatt) $H := 50 \text{ m}$

Nivåfaktor
(under tregrensen) $c_{alt} := 1.0$

NA.4.2(2)P(901.2) Retningsfaktor

Tab NA.4.(901.4)

Velger mest ugunstig $c_{dir} := 1.0$

NA.4.2(2)P(901.3) Faktor for årtidsvariasjon

Tab NA.4(901.5)

Årtidsfaktor (september - april) $c_{season} := 1.0$

Basisvindhastighet (NA.4.1) $v_b := c_{dir} \cdot c_{season} \cdot c_{alt} \cdot c_{prob} \cdot v_{b,0} = 23 \frac{m}{s}$

4.3 Middelvind

Midlere høyde av
konstruksjon over bakken $z := 5 \text{ m}$

NA.4.3.2 Terrengruhet

Tab NA.4.1

Kategori II $k_r := 0.19$ $z_0 := 0.05 \text{ m}$ $z_{min} := 4 \text{ m}$

4.3.2 Terrengruhet

Ruhetsfaktor (4.1) for

$$z_{min} < z < z_{max} \quad 4 \text{ m} < 5 \text{ m} < 200 \text{ m}$$

$$(z_{max} = 200 \text{ m}) \quad (4.3.2)$$

$$c_r := k_r \cdot \ln\left(\frac{z}{z_0}\right) = 0.875$$

NA.4.3.3 Terrengform

Terrengformfaktor

$$c_0 := 1.0$$

Turbulensfaktor

$$k_I := 1.0$$

Begge settes lik 1.0 jfr SVV
668 Beregningsveiledning for
etteropspente betongbruer

Middelvind (4.3)

$$v_m := c_r \cdot c_0 \cdot v_b = 20.125 \frac{\text{m}}{\text{s}}$$

4.4 Vindturbulens

Turbulensavvik

$$\sigma_v := k_r \cdot v_b \cdot k_I = 4.37 \frac{\text{m}}{\text{s}}$$

Turbulens intensitet

$$I_v := \frac{\sigma_v}{v_m} = 0.217$$

4.5 Vindkasthastighet og vindkasthastighetstrykk

NA.4.4 Vindkasthastighetstrykket

$$v_s := v_m$$

$$k_p := 3.5$$

Vindkasthastighet

$$v_p := \left(1 + 2 \cdot k_p \cdot I_v\right)^{\frac{1}{2}} \cdot v_s = 31.947 \frac{\text{m}}{\text{s}}$$

Luftens densitet

$$\rho := 1.25 \frac{\text{kg}}{\text{m}^3}$$

Stedsvindhastighetstrykket

$$q_m := 0.5 \cdot \rho \cdot v_m^2 = 0.253 \frac{\text{kN}}{\text{m}^2}$$

Vindkasthastighetstrykk

$$q_p := \left(1 + 2 \cdot k_p \cdot I_v\right) \cdot q_m = 0.638 \frac{\text{kN}}{\text{m}^2}$$

Vindlast på overbygning

8.3 Kraftkoeffisienter

Horisontalkraft i tverretning

8.3.1 Kraftkoeffisient i x-retning

Høyde bru (+vogntog)

$$d_{tot} := 2 \text{ m} + 1.25 \text{ m} = 3.25 \text{ m}$$

Bredde bru

$$b := 8500 \text{ mm}$$

$$\frac{b}{d_{tot}} = 2.615$$

Fig. 8.3

$$c_{fx.0} := 1.8$$

Kraftkoeffisient for vindlast på brudekke i x-retning
Reduserer ikke c_{fx} mht. helning på tverrsnitt til sikker side.

$$c_{f.x} := c_{fx.0}$$

Basis hastighetstrykk (4.10)

$$q_b := \frac{1}{2} \cdot \rho \cdot v_b^2 = 0.331 \frac{\text{kN}}{\text{m}^2}$$

Eksponeringsfaktor (4.9)

$$c_e := \frac{q_p}{q_b} = 1.929$$

Jevnt fordelt vindlast:

$$f_x := c_{f.x} \cdot c_e \cdot d_{tot} \cdot q_p = 7.199 \frac{\text{kN}}{\text{m}}$$

Vertikalkraft

8.3.3 Vindlaster på bru i z-retning

Kraftfaktor

$$c_{fz} := 0.9$$

Jevnt fordelt vindlast

$$f_z := c_{fz} \cdot b \cdot q_p = 4.88 \frac{\text{kN}}{\text{m}}$$

Vertikal kraft virker med en eksentrisitet på $e=b/4$ fra bruens senterlinje, ref 8.3.3

$$e := \frac{b}{4} = 2.125 \text{ m}$$

Vindlast på søyler

7.6 Konstruksjoner med rektangulært tverrsnitt

Tverrsnitt søyler	$d_{s1} := 3900 \text{ mm}$	$d_{s2} := 3900 \text{ mm}$
	$b_{s1} := 800 \text{ mm}$	$b_{s2} := 800 \text{ mm}$
Forhold søyler	$\frac{d_{s1}}{b_{s1}} = 4.875$	$\frac{d_{s2}}{b_{s2}} = 4.875$
Kraftfaktor for rektangulært tverrsnitt fig 7.23	$c_{f.0.s1} := 1.10$	$c_{f.0.s2} := 1.10$
Reduksjonsfaktor ψ_r fra fig. 7.24	$\psi_r := 1.0$	for $r = 0$

7.13 Effektiv slankhet og endeeffektfaktor

Massivitetsfaktor (ingen åpninger)	$\varphi := 1.0$	
Høyde søyler	$l_{s1} := 3500 \text{ mm}$	$l_{s2} := 6500 \text{ mm}$
Tab. 7.16 anbefalt verdi for slankhet for $l < 15\text{m}$	$\lambda := 70$ eller	$\lambda_{s1} := 2 \cdot \frac{l_{s1}}{b_{s1}} = 8.75$
		$\lambda_{s2} := 2 \cdot \frac{l_{s2}}{b_{s2}} = 16.25$
Endeeffekt faktor fra fig. 7.36	$\psi_{\lambda.s1} := 0.68$	$\psi_{\lambda.s2} := 0.73$
Kraftkoeffisient for rektangulære tverrsnitt	$c_{f.s1} := c_{f.0.s1} \cdot \psi_r \cdot \psi_{\lambda.s1} = 0.748$	
	$c_{f.s2} := c_{f.0.s2} \cdot \psi_r \cdot \psi_{\lambda.s2} = 0.803$	

6.2 Bestemmelse av csds

For bygninger under 15m settes	$c_s d_s := 1$
--------------------------------	----------------

5.3 Vindkrefter

Søyle 1

$$f_{x.s1} := c_s d_s \cdot c_{f.s1} \cdot q_p \cdot b_{s1} = 0.382 \frac{kN}{m}$$

Søyle 2

$$f_{x.s2} := c_s d_s \cdot c_{f.s2} \cdot q_p \cdot b_{s2} = 0.41 \frac{kN}{m}$$

Setter forenklet vindlast på søyle til høyeste opptredende vindlast

$$f_{x.s} := f_{x.s2} = 0.41 \frac{kN}{m}$$

F Temperaturlast

Temperaturlast

NS-EN 1991-1-5:2003 + NA:2008 Laster på konstruksjoner - Del 1-5: Allmenne laster - Termiske påvirkninger

6.1.3 Jevnt fordelt temperaturlastet

Type 3 betongbru

$$T_{maks} := 36 \quad [NA.A1]$$

$$T_{min} := -35 \quad [NA.A2]$$

$$Te_{maks} := T_{maks} - 3 = 33$$

$$Te_{min} := T_{min} + 8 = -27$$

$$To := 10 \quad [NA.A1(3)]$$

$$\Delta T_{n.exp} := Te_{maks} - To = 23 \quad [6.1.3.3(3)]$$

$$\Delta T_{n.con} := Te_{min} - To = -37$$

6.1.4 + NA 6.1.4 Vertikalt linært varierende temperaturandel (Metode 1)

$$\Delta T_{m.heat} := 15 \quad [Tab NA 6.1]$$

$$\Delta T_{m.cool} := -8$$

Antar 100mm beleg

$$k_{sur.o} := 0.7 \quad k_{sur.u} := 1.0 \quad [Tab NA 6.2]$$

$$\Delta T_{m.heat.new} := \Delta T_{m.heat} \cdot k_{sur.o} = 10.5$$

$$\Delta T_{m.cool.new} := \Delta T_{m.cool} \cdot k_{sur.u} = -8$$

6.1.5 + NA 6.1.5 Samtidigheten av temperaturlast

$$\omega_n := 0.35$$

$$\omega_m := 0.75$$

$$\Delta T_{M,heat} \text{ (eller } \Delta T_{M,cool}) + \omega_N \Delta T_{N,exp} \text{ (OF } \Delta T_{N,con}) \quad (6.3)$$

eller

$$\omega_M \Delta T_{M,heat} \text{ (OF } \Delta T_{M,cool}) + \Delta T_{N,exp} \text{ (OF } \Delta T_{N,con}) \quad (6.4)$$

$$\Delta T_{m,heat,new} = 10.5$$

$$\Delta T_{m,heat,new} \cdot \omega_m = 7.875$$

$$\Delta T_{m,cool,new} = -8$$

$$\Delta T_{m,cool,new} \cdot \omega_m = -6$$

$$\Delta T_{n,exp} = 23$$

$$\Delta T_{n,exp} \cdot \omega_n = 8.05$$

$$\Delta T_{n,con} = -37$$

$$\Delta T_{n,con} \cdot \omega_n = -12.95$$

Komb	$\Delta T_{M,heat}$	$\Delta T_{M,cool}$	$\Delta T_{N,exp}$	$\Delta T_{N,con}$
1	1,0		0,35	
2	0,75		1,0	
3	1,0			0,35
4	0,75			1,0
5		1,0	0,35	
6		0,75	1,0	
7		1,0		0,35
8		0,75		1,0

G Armeringskontroll

Fritt opplagt bjelke armeringskontroll

$$g := 25 \frac{kN}{m^3}$$

$$b := 200 \text{ mm}$$

$$h := 600 \text{ mm}$$

$$Egenlast := g \cdot b \cdot h = 3 \frac{kN}{m}$$

$$L := 10 \text{ m}$$

$$M_{ed} := 1.35 \cdot Egenlast \cdot \frac{L^2}{8} = 50.625 \text{ kN} \cdot \text{m}$$

$$c := 50 \text{ mm}$$

$$\alpha_{cc} := 0.85 \quad f_{ck} := 30 \frac{N}{mm^2} \quad f_{yk} := 500 \frac{N}{mm^2} \quad \gamma_c := 1.5 \quad \gamma_s := 1.15$$

$$f_{cd} := \alpha_{cc} \cdot \frac{f_{ck}}{\gamma_c} = (1.7 \cdot 10^7) \text{ Pa}$$

$$f_{yd} := \frac{f_{yk}}{\gamma_s} = (4.348 \cdot 10^8) \text{ Pa}$$

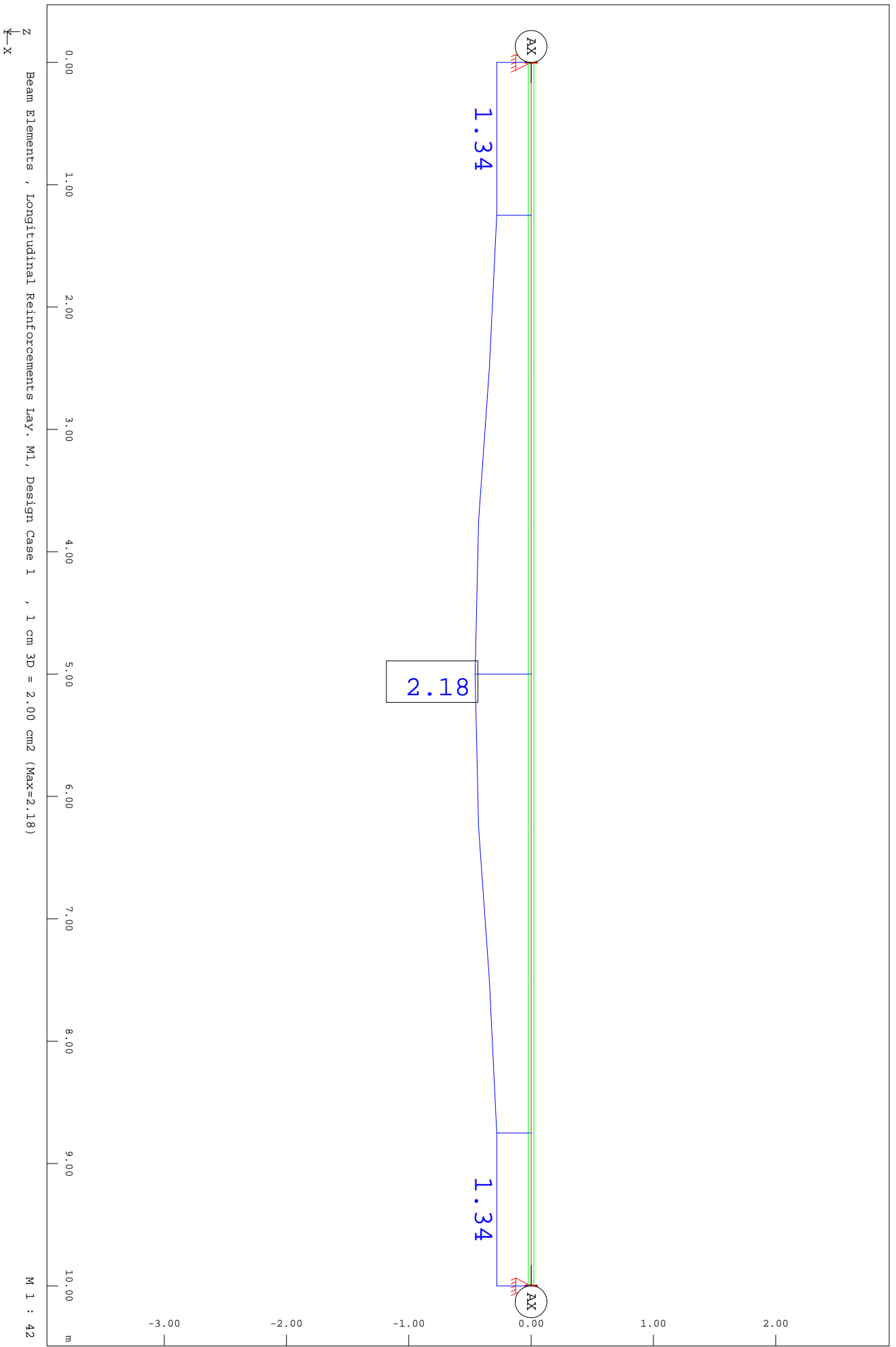
$$d := h - c = 0.55 \text{ m}$$

$$M_{rd} := 0.275 f_{cd} \cdot b \cdot d^2 = 282.838 \text{ kN} \cdot \text{m}$$

$$z := \left(1 - 0.17 \cdot \frac{M_{ed}}{M_{rd}} \right) = 0.97$$

$$z := 0.97 \cdot d = 0.534 \text{ m}$$

$$A_s := \frac{M_{ed}}{f_{yd} \cdot z} = 218.252 \text{ mm}^2$$



H Armeringsberegninger

Beregning av armeringsmengde

1.0 Effektiv Flensbredde EK 2 pkt.(5.3.2.1)

1.1: I første felt gir l_0

$$b := 7500 \text{ mm} \quad l_1 := 20000 \text{ mm}$$

$$b_w := 3900 \text{ mm} \quad l_0 := 0.85 \cdot l_1 = 17 \text{ m}$$

$$b_1 := 1800 \text{ mm}$$

$$b_2 := 1800 \text{ mm}$$

$$0.2 \cdot l_0 = 3.4 \text{ m}$$

$$b_{eff.1} := \min(0.2 \cdot b_1 + 0.1 \cdot l_0, b_1, 0.2 \cdot l_0) = 1.8 \text{ m}$$

$$b_{eff.2} := \min(0.2 \cdot b_2 + 0.1 \cdot l_0, b_2, 0.2 \cdot l_0) = 1.8 \text{ m}$$

$$b_{eff} := b_{eff.1} + b_{eff.2} + b_w = 7.5 \text{ m} = b$$

1.2: I mellomliggende felt l_0 .

$$l_2 := 30000 \text{ mm}$$

$$l_0 := 0.7 l_2 = 21 \text{ m}$$

$$b_{eff.1} := \min(0.2 \cdot b_1 + 0.1 \cdot l_0, b_1, 0.2 \cdot l_0) = 1.8 \text{ m}$$

$$b_{eff.2} := \min(0.2 \cdot b_2 + 0.1 \cdot l_0, b_2, 0.2 \cdot l_0) = 1.8 \text{ m}$$

$$b_{eff} := b_{eff.1} + b_{eff.2} + b_w = 7.5 \text{ m}$$

1.3: Over støtte

$$l_{0s} := 0.15 (l_1 + l_2) = 7.5 \text{ m}$$

$$b_{effs.1} := \min(0.2 \cdot b_1 + 0.1 \cdot l_{0s}, b_1, 0.2 \cdot l_{0s}) = 1.11 \text{ m}$$

$$b_{effs.2} := \min(0.2 \cdot b_2 + 0.1 \cdot l_{0s}, b_2, 0.2 \cdot l_{0s}) = 1.11 \text{ m}$$

$$b_{eff.s} := b_{effs.1} + b_{effs.2} + b_w = 6.12 \text{ m}$$

2.0 Bestemme slakkarmering for rent moment

- Betongkonstruksjoner, side 54. (4.33)

$$h_{tot} := 1250 \text{ mm}$$

$$t_{s\oyle} := 500 \text{ mm}$$

$$t := 300 \text{ mm}$$

$$d_g := 32 \text{ mm}$$

$$c_{min} := 60 \text{ mm}$$

$$h := 1137.5 \text{ mm}$$

-Redusert høyde mtp skrått profi,
konservativt

$$d := h - c_{min} - 25 \text{ mm} - \frac{37.5}{2} \text{ mm} = 1033.75 \text{ mm}$$

$$n := \frac{t}{d} = 0.29$$

-B20-B45 maks t/d= 0.33

$$\gamma_s := 1.15$$

$$\gamma_c := 1.5$$

$$\alpha_{cc} := 0.85$$

$$f_{yk} := 500 \frac{\text{N}}{\text{mm}^2}$$

$$f_{ck} := 45 \frac{\text{N}}{\text{mm}^2}$$

$$f_{yd} := \frac{f_{yk}}{\gamma_s} = 434.783 \frac{\text{N}}{\text{mm}^2}$$

$$f_{cd} := \alpha_{cc} \cdot \frac{f_{ck}}{\gamma_c} = 25.5 \frac{\text{N}}{\text{mm}^2}$$

$$F_{ed.sup} := 8672.7 \text{ kN}$$

$$\Delta M_{ed.ok} := \frac{F_{ed.sup} \cdot t_{s\oyle}}{8} = 542.044 \text{ kN} \cdot \text{m}$$

$$M_{ed.ok} := 29384 \text{ kN} \cdot \text{m}$$

$$M_{ed.ok.red} := M_{ed.ok} - \Delta M_{ed.ok} = 28841.956 \text{ kN} \cdot \text{m}$$

$$M_{ed.uk} := 21055 \text{ kN} \cdot \text{m}$$

$$M_{rd.støtte} := f_{cd} \cdot b_{eff.s} \cdot t \cdot (d - 0.5 \cdot t) = 41375.408 \text{ kN} \cdot \text{m}$$

$$M_{rd.felt} := f_{cd} \cdot b_{eff} \cdot t \cdot \left(d - \frac{t}{2}\right) = 50705.156 \text{ kN} \cdot \text{m}$$

$$z_1 := \left(1 - 0.17 \cdot \frac{M_{ed.ok}}{M_{rd.støtte}}\right) = 0.879$$

$$A_{s.1} := \frac{M_{ed.ok}}{f_{yd} \cdot z_1 \cdot d} = 74353.479 \text{ mm}^2$$

$$z_2 := \left(1 - 0.17 \cdot \frac{M_{ed.uk}}{M_{rd.felt}}\right) = 0.929$$

$$A_{s.2} := \frac{M_{ed.uk}}{f_{yd} \cdot z_2 \cdot d} = 50403.523 \text{ mm}^2$$

$$z_3 := \left(1 - 0.17 \cdot \frac{M_{ed.ok.red}}{M_{rd.støtte}}\right) = 0.881$$

$$A_{s.3} := \frac{M_{ed.ok.red}}{f_{yd} \cdot z_3 \cdot d} = 72797.499 \text{ mm}^2$$

$$R_{\phi 20} := 10 \text{ mm}$$

$$R_{\phi 25} := 12.5 \text{ mm}$$

$$A_{\phi 20} := \pi \cdot R_{\phi 20}^2 = 314.159 \text{ mm}^2$$

$$A_{\phi 25} := \pi \cdot R_{\phi 25}^2 = 490.874 \text{ mm}^2$$

Armeringsvalg:

$$u := \frac{A_{s.2}}{A_{\phi 20}} = 160.439$$

$$o := \frac{A_{s.1}}{A_{\phi 25}} = 151.472$$

Senteravstand:

$$s.1 := A_{\phi 25} \cdot \frac{(b_w - 2 \cdot c_{min})}{A_{s.1}} = 24.955 \text{ mm}$$

$$s.4 := A_{\phi 25} \cdot \frac{4 \cdot (b_w - 2 \cdot c_{min})}{A_{s.1}} = 99.821 \text{ mm}$$

Valg av armering, eksempelvis 4 lag Ø25 senter 100mm i overkant

$$s_{valgt} := 100 \text{ mm}$$

$$A_{\phi 25.c100} := A_{\phi 25} \cdot \frac{(4 \cdot b_w - 2 \cdot c_{min})}{s_{valgt}} = (7.599 \cdot 10^4) \text{ mm}^2$$

$$a_h := \max(4 \cdot R_{\phi 25}, d_g + 5 \text{ mm}, 20 \text{ mm}) = 50 \text{ mm}$$

$$a_v := \max(1.5 \cdot 2 \cdot R_{\phi 25}, d_g + 5 \text{ mm}, 20 \text{ mm}) = 37.5 \text{ mm}$$

Trykksonekapaistet:

$$f_{cd} = 25.5 \frac{\text{N}}{\text{mm}^2}$$

$$\sigma_{cd} := \frac{M_{ed.ok}}{t \cdot b_{eff} \cdot (d - 0.5 \cdot t)} = 10.589 \frac{\text{N}}{\text{mm}^2}$$

$$\sigma_{cd} := \frac{M_{ed.uk}}{t \cdot b_{eff.s} \cdot (d - 0.5 \cdot t)} = 18.11 \frac{\text{N}}{\text{mm}^2}$$

Midlere høyde

$$h.middel := \frac{(h + h.tot)}{2} = 1.194 \text{ m}$$

$$d := h.middel - c_{min} - 25 \text{ mm} - \frac{37.5}{2} \text{ mm} = 1090 \text{ mm}$$

$$F_{ed.sup} := 8672.7 \text{ kN}$$

$$\Delta M_{ed.ok} := \frac{F_{ed.sup} \cdot t_{søyle}}{8} = 542.044 \text{ kN} \cdot \text{m}$$

$$M_{ed.ok} := 29384 \text{ kN} \cdot \text{m}$$

$$M_{ed.ok.red} := M_{ed.ok} - \Delta M_{ed.ok} = 28841.956 \text{ kN} \cdot \text{m}$$

$$M_{ed.uk} := 21055 \text{ kN} \cdot \text{m}$$

$$M_{rd.støtte} := f_{cd} \cdot b_{eff.s} \cdot t \cdot (d - 0.5 \cdot t) = 44008.92 \text{ kN} \cdot \text{m}$$

$$M_{rd.felt} := f_{cd} \cdot b_{eff} \cdot t \cdot \left(d - \frac{t}{2}\right) = 53932.5 \text{ kN} \cdot \text{m}$$

$$z_1 := \left(1 - 0.17 \cdot \frac{M_{ed.ok}}{M_{rd.støtte}}\right) = 0.886$$

$$A_{s.1} := \frac{M_{ed.ok}}{f_{yd} \cdot z_1 \cdot d} = 69941.75 \text{ mm}^2$$

$$z_2 := \left(1 - 0.17 \cdot \frac{M_{ed.uk}}{M_{rd.felt}}\right) = 0.934$$

$$A_{s.2} := \frac{M_{ed.uk}}{f_{yd} \cdot z_2 \cdot d} = 47586.141 \text{ mm}^2$$

$$z_3 := \left(1 - 0.17 \cdot \frac{M_{ed.ok.red}}{M_{rd.støtte}}\right) = 0.889$$

$$A_{s.3} := \frac{M_{ed.ok.red}}{f_{yd} \cdot z_3 \cdot d} = 68489.773 \text{ mm}^2$$

3.0 Bestemme momentkapasitet til sammenligning retangulært tverrsnitt

$$\lambda := 0.8 \quad \eta := 1$$

$$\varepsilon_{cu} := 0.0035$$

$$\varepsilon_{yk} := 0.0025$$

$$\alpha := \frac{(\varepsilon_{cu})}{(\varepsilon_{cu} + 2 \cdot \varepsilon_{yk})} = 0.412$$

$$x := \alpha \cdot d = 448.824 \text{ mm} \quad \text{-Trykksone høyde, underarmert tverrsnitt, mer en tykkelse flens}$$

$$M_{rd,s} := 0.8 \cdot \alpha \cdot (1 - 0.4 \cdot \alpha) \cdot f_{cd} \cdot b_w \cdot d^2 = 32511.448 \text{ kN} \cdot \text{m}$$

4.0 Dimensjonere tverrsnitt med spennarmering [7.1.3]

$$\varepsilon_{cu} := 0.0035$$

$$f_{p0.1k} := 1600 \frac{N}{mm^2}$$

$$\gamma_s := 1.15$$

$$f_{cd} = 25.5 \frac{N}{mm^2}$$

$$f_{pd} := \frac{f_{p0.1k}}{\gamma_s} = (1.391 \cdot 10^3) \frac{N}{mm^2}$$

$$E_p := 195000 \frac{N}{mm^2}$$

$$f_{pk} := 1860 \frac{N}{mm^2}$$

$$\sigma_{p0} := \min(0.8 f_{pk}, 0.9 \cdot f_{p0.1k}) = (1.44 \cdot 10^3) \frac{N}{mm^2}$$

$$\varepsilon_{po} := \frac{\sigma_{p0}}{E_p} = 0.007$$

$$\Delta\varepsilon_{tap} := 0.96 \cdot \varepsilon_{po} = 0.007$$

- Antagelse av tap

$$\varepsilon'_{po} := \varepsilon_{po} - \Delta\varepsilon_{tap} = 2.954 \cdot 10^{-4}$$

$$\Delta\varepsilon_p := \frac{f_{pd}}{E_p} - \varepsilon'_{po} = 0.007$$

$$\alpha_b := \frac{(\varepsilon_{cu})}{(\Delta\varepsilon_p + \varepsilon_{cu})} = 0.339$$

-Trykksonehøyden for balansert

$$d = (1.09 \cdot 10^3) \text{ mm}$$

$$A_{pb} := 0.8 \cdot \frac{f_{cd}}{f_{pd}} \cdot \alpha_b \cdot b_{eff.s} \cdot d = 33109.609 \text{ mm}^2$$

-Nødvendig armering for balansert tverrsnitt

Velger spennarmering

$$A_{\emptyset p} := 2250 \text{ mm}^2 \quad A_n := 11$$

$$A_p := A_{\emptyset p} \cdot A_n = 24750 \text{ mm}^2$$

< A_{pb} , armeringen flyter

$$\alpha := \frac{(f_{pd} \cdot A_p)}{0.8 \cdot f_{cd} \cdot b_{eff.s} \cdot d} = 0.253$$

$$\alpha \cdot d = 275.814 \text{ mm}$$

$$z := (1 - \alpha \cdot 0.4) \cdot d = 979.675 \text{ mm}$$

-Momentarmen fra Tc til S

$$M_{ed} := \max(M_{ed.uk}, M_{ed.ok}) = 29384 \text{ kN} \cdot \text{m}$$

$$M_{rd} := 0.8 \cdot (1 - \alpha \cdot 0.4) \cdot \alpha \cdot b_{eff.s} \cdot d^2 \cdot f_{cd} = 33734.88 \text{ kN} \cdot \text{m}$$

*Dette er en konservativ metode siden slakkarmering ikke er tatt hensyn til, men $M_{ed} < M_{rd}$ tilstrekkelig kapasitet

Vurderer anderledes metode (7.2)

$$d := \left(\frac{M_{ed}}{0.269 \cdot f_{cd} \cdot b_{eff.s}} \right)^{\frac{1}{2}} = 836.63 \text{ mm}$$

$$z := (1 - \alpha \cdot 0.4) \cdot d = 751.95 \text{ mm}$$

$$A_p := \frac{M_{ed}}{f_{pd} \cdot 0.84 \cdot d} = 30052.179 \text{ mm}^2$$

$$n := \frac{A_p}{A_{\emptyset p}} = 13.357$$

$$A_p := A_{\emptyset p} \cdot 14 = 31500 \text{ mm}^2$$

5. 0 Kontroll av oppspenningstilstand, M- og N diagram for gitt tverrsnitt.

$$\Delta_{dev} := 10 \text{ mm} \qquad h := 1135 \text{ mm} \quad - \text{Konservativt}$$

$$c_{nom} := 50 \text{ mm} \qquad e_{spenn} := 400 \text{ mm} \quad - \text{Eksentrisitet for kabel}$$

$$c_{min} := c_{nom} + \Delta_{dev} = 60 \text{ mm}$$

$$\gamma_s := 1.15$$

$$\gamma_c := 1.5$$

$$\alpha_{cc} := 0.85$$

$$E_s := 200000 \frac{\text{N}}{\text{mm}^2} \qquad E_p := 195000 \frac{\text{N}}{\text{mm}^2}$$

$$f_{ck} := 45 \frac{\text{N}}{\text{mm}^2} \qquad f_{yk} := 500 \frac{\text{N}}{\text{mm}^2}$$

$$f_{cd} := \alpha_{cc} \cdot \frac{f_{ck}}{\gamma_c} = 25.5 \frac{\text{N}}{\text{mm}^2} \quad f_{yd} := \frac{f_{yk}}{\gamma_s} = 434.783 \frac{\text{N}}{\text{mm}^2}$$

Valgt armering

$$A_{\phi 20} := \pi \cdot (25 \text{ mm})^2 = 1963.495 \text{ mm}^2$$

$$m := 5$$

$$A_{s,1} := m \cdot A_{\phi 20} = 9817.477 \text{ mm}^2$$

$$A_p = 31500 \text{ mm}^2$$

1. For rent trykk

$$\varepsilon_{c,1} := 1.75 \cdot 10^{-3}$$

$$\varepsilon_p := 1.75 \cdot 10^{-3}$$

$$d_1 := \frac{h}{2} - c_{min} = 507.5 \text{ mm}$$

$$b := b_w = 3900 \text{ mm}$$

Tykkresultant

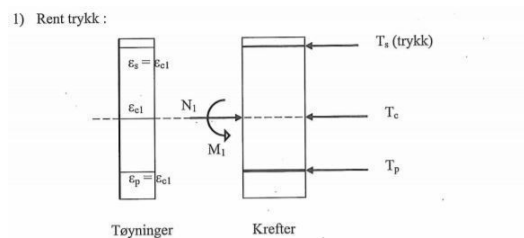
$$T_c := f_{cd} \cdot b \cdot h = (1.129 \cdot 10^5) \text{ kN}$$

$$T_s := E_s \cdot \varepsilon_{c,1} \cdot A_{s,1} = (3.436 \cdot 10^3) \text{ kN}$$

$$T_p := E_p \cdot \varepsilon_p \cdot A_p = (1.075 \cdot 10^4) \text{ kN}$$

$$N_1 := T_c + T_s + T_p = (1.271 \cdot 10^5) \text{ kN}$$

$$M_1 := (T_p \cdot e_{spenn}) - T_s \cdot d_1 = (2.556 \cdot 10^3) \text{ kN} \cdot \text{m}$$



2. Balansepunkt

$$\varepsilon_{cu} := 3.5 \cdot 10^{-3}$$

$$\varepsilon_s := 2.17 \cdot 10^{-3}$$

$$a := \frac{h}{2} - e_{spenn} = 167.5 \text{ mm}$$

*avstand fra bunn til tp.kabel

$$\alpha := \frac{\varepsilon_{cu}}{\varepsilon_{cu} + \varepsilon_s} = 0.617 \quad \alpha \cdot d = 516.438 \text{ mm}$$

$$\Delta\varepsilon_p := \frac{\varepsilon_{cu}}{\alpha \cdot d} \cdot (\alpha \cdot d - a) = 0.002$$

Trykkresultant

$$T_c := 0.8 \cdot \alpha \cdot d \cdot b \cdot f_{cd} = (4.109 \cdot 10^4) \text{ kN}$$

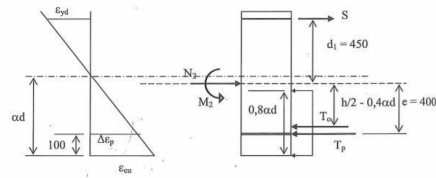
$$T_p := \Delta \varepsilon_p \cdot E_p \cdot A_p = (1.453 \cdot 10^4) \text{ kN}$$

$$S_s := E_s \cdot \varepsilon_s \cdot A_{s,1} = (4.261 \cdot 10^3) \text{ kN}$$

$$N_2 := T_c + T_p - S_s = (5.135 \cdot 10^4) \text{ kN}$$

$$M_2 := T_c \cdot (0.5 \cdot h - 0.4 \cdot \alpha \cdot d) + T_p \cdot e_{spenn} + S_s \cdot d_1 = (2.28 \cdot 10^4) \text{ kN} \cdot \text{m}$$

2) Balansepunkt, $\varepsilon_c = \varepsilon_{cu} = 3,5 \cdot 10^{-3}$ og $\varepsilon_s = \varepsilon_{yd} = 2,17 \cdot 10^{-3}$



3. Full flyt

$$\varepsilon_{cu} := 3.5 \cdot 10^{-3}$$

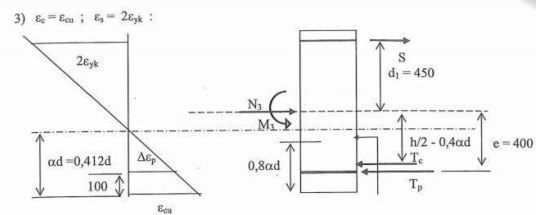
$$\varepsilon_s := 5 \cdot 10^{-3}$$

$$a := \frac{h}{2} - e_{spenn} = 167.5 \text{ mm}$$

$$\alpha := \frac{\varepsilon_{cu}}{\varepsilon_{cu} + \varepsilon_s} = 0.412 \quad \alpha \cdot d = 344.495 \text{ mm}$$

$$\Delta \varepsilon_p := \frac{\varepsilon_{cu}}{\alpha \cdot d} \cdot (\alpha \cdot d - a) = 0.002$$

avstand fra bunn til tp.kabel



Trykkresultant

$$T_c := 0.8 \cdot \alpha \cdot d \cdot b \cdot f_{cd} = (2.741 \cdot 10^4) \text{ kN}$$

$$T_p := \Delta \varepsilon_p \cdot E_p \cdot A_p = (1.105 \cdot 10^4) \text{ kN}$$

$$S_s := E_s \cdot \varepsilon_s \cdot A_{s,1} = (9.817 \cdot 10^3) \text{ kN}$$

$$N_3 := T_c + T_p - S_s = (2.864 \cdot 10^4) \text{ kN}$$

$$M_3 := T_c \cdot (0.5 \cdot h - 0.4 \alpha \cdot d) + T_p \cdot e_{spenn} + S_s \cdot d_1 = (2.118 \cdot 10^4) \text{ kN} \cdot \text{m}$$

4.Full flyt

$$\varepsilon_{cu} := 3.5 \cdot 10^{-3}$$

$$\varepsilon_s := 10 \cdot 10^{-3}$$

$$a := \frac{h}{2} - e_{spenn} = 167.5 \text{ mm}$$

avstand fra bunn til tp.kabel

$$\alpha := \frac{\varepsilon_{cu}}{\varepsilon_{cu} + \varepsilon_s} = 0.259 \quad \alpha \cdot d = 216.904 \text{ mm}$$

$$\Delta\varepsilon_p := \frac{\varepsilon_{cu}}{\alpha \cdot d} \cdot (\alpha \cdot d - a) = 7.972 \cdot 10^{-4}$$

Trykkresultant

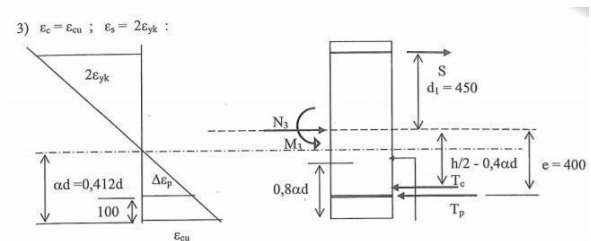
$$T_c := 0.8 \cdot \alpha \cdot d \cdot b \cdot f_{cd} = (1.726 \cdot 10^4) \text{ kN}$$

$$T_p := \Delta\varepsilon_p \cdot E_p \cdot A_p = (4.897 \cdot 10^3) \text{ kN}$$

$$S_s := E_s \cdot \varepsilon_s \cdot A_{s,1} = (1.963 \cdot 10^4) \text{ kN}$$

$$N_4 := T_c + T_p - S_s = (2.519 \cdot 10^3) \text{ kN}$$

$$M_4 := T_c \cdot (0.5 \cdot h - 0.4 \alpha \cdot d) + T_p \cdot e_{spenn} + S_s \cdot d_1 = 20219.492 \text{ kN} \cdot \text{m}$$



MAX/MIN (M)

$$N_{ed1} := 36446 \text{ kN}$$

$$M_{ed1} := 11008 \text{ J}$$

$$N_{ed2} := 36451 \text{ kN}$$

$$M_{ed2} := 18684 \text{ J}$$

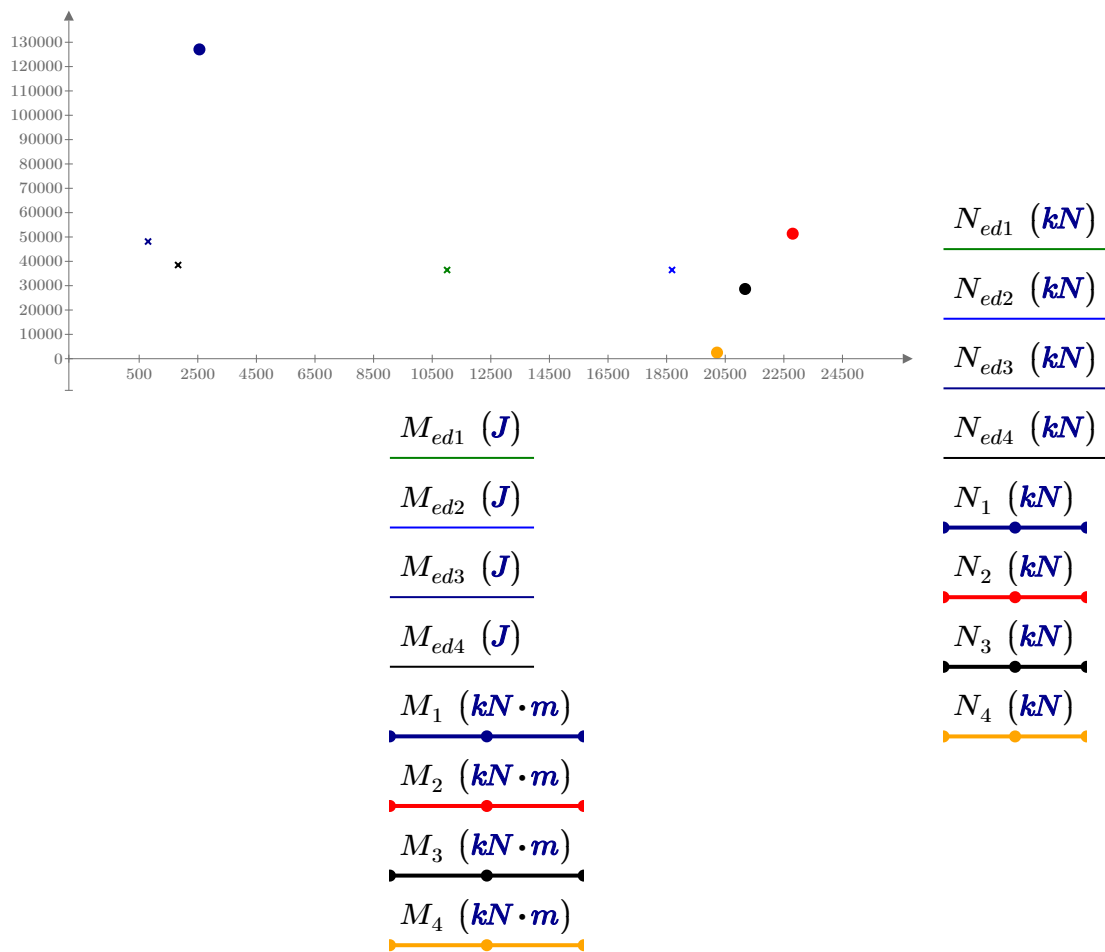
MAX/MIN (N)

$$N_{ed3} := 48144 \text{ kN}$$

$$M_{ed3} := 803 \text{ J}$$

$$N_{ed4} := 38481 \text{ kN}$$

$$M_{ed4} := 1828 \text{ J}$$



I Skript fra Visual studio

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Autodesk.DesignScript.Runtime;
7 using Autodesk.DesignScript.Interfaces;
8 using Autodesk.DesignScript.Geometry;
9
10 namespace Spennarmering_detalj
11 {
12     public class Spennarmering_del_en
13     {
14         public static List<string> Input(string Tittel, int Spennkabelnummer,
15             int Materialnummer, int Kabeltype, string Referanseakse, double
16             Startpunkt, double Sluttpunkt, List<double> Spenn)
17         {
18             // kabeltype er gitt fra Sofistik, BBV ETA NO er en standard type så
19             // kan man velge mellom ulike her. Standard ifølge SVV668
20             // Material bestemmes også fra sofistik slik at man etterpå velger
21             // nummeret referet til materialet i sofistik hvis man har importert
22             // geometri og lagt til material
23             //spennkabel nummer er for å kunne gi høyde for x antall kabler og
24             // nummerer sådan. alternativ ta spennkabelnummer som en variabel n og
25             // si antall kabler og da generere flere tekstfiler.
26
27             List<string> name = new List<string>();
28             name.Add("+PROG TENDON $ Spennarmering");
29             name.Add("HEAD " + Tittel);
30             name.Add("PAGE UNII 0");
31             name.Add("SYSP NOPS " + Spennkabelnummer + " COMP BBV ETA NO " +
32                 Kabeltype + " MAT " + Materialnummer);
33             name.Add("$ GEOMETRY Noh er nummeret til kabelen og etter auto skal
34                 man skrive inn nodenummeret man etablere referanseakse");
35
36             int n = Spenn.Count; //antall iterationer
37
38             name.Add("AXES NOH " + Spennkabelnummer + " TYPE REFX " +
39                 Referanseakse + "'");
40             name.Add("TOPP NOH " + Spennkabelnummer + " SP " + Startpunkt + " + 0 + " KIND
41                 REFX S " + Startpunkt); //første punkte
42             for (int i = 0; i < n; i++)
43             {
44                 name.Add("TOPP NOH " + Spennkabelnummer + " SP " + Spenn[i] + " + (i +
45                     1) + " KIND REFX S " + ((Spenn[i]))); //Lager n-antall
46                 spennoppdeling
47             }
48         }
49     }
50 }
```

```
37     }
38     name.Add("TOPP NOH " + Spennkabelnummer + " SP          " + (n + 1) + ↗
39     " KIND REFX S " + Sluttpunkt); //Siste punkt ↗
39     name.Add("TGEO NOG " + Spennkabelnummer + " NOH " + Spennkabelnummer ↗
40     + " NOPS " + Spennkabelnummer + " TITL " + "" + Tittel + "");
40     name.Add("PTUV S          U          V          DVS          RV          RL          ↗
41     DL          DR          TYPE=SPAN $ TYPE = SPAN = xi - spaninput via top ↗
41     points");
41     return name;
42
43 }
44 }
45
46 }
47
48
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace Spennarmering_detalj
6 {
7     public class Eksentrisitet
8     {
9
10         // Punkt velge et referansepunkt knyttet til spennoppdelingen.
11         //DVS vertikal inklinasjon dv/ds hvor raskt kabelføringen skal gå opp   ↗
12         // per strekning bortover, så vidt jeg tolker det,ofte null
13         // Tverrplassering U er hvor kabelen ligger i forhold til bjelkeaksens   ↗
14         // sentrum U=1 gir 1 meter i positiv y akse
15         // eksentrisitet V er kabelens plassering i z aksen i punktet fra       ↗
16         // toppunktet til bjelkeaksen. dvs V=0.2 betyr 200mm fra toppen av     ↗
17         // bjelken.
18         // RV, radius til vertikalen V, ofte kun verdier for "topppunktene"
19         // RL, Influenslengden til radiusen, ofte også kun verdi i toppunktene. ↗
20         // vanlig med 1, men sier hvor "buet" sirkelen er stort tall gir mer   ↗
21         // kompakt bue
22         // DL og DR sier om det skal være rett linje i punktet og hvor lang det ↗
23         // skal være i henholdvis negativ og positiv retning kan ikke være   ↗
24         // samtidig som RV og RL
25         //name.Add("PTUV S      U      V      DVS      RV      RL      DL ↗
26         //      DR      TYPE=SPAN $ TYPE = SPAN = xi - spaninput via top points"); ↗
27         // er gitt i første linjen
28
29         public static List<string> Input(double Punkt = -1, double           ↗
30             Tverrplassering = 0, double Eksentrisitet = 0, double DVS = 0, double ↗
31             Vertikal_radius = 0, double Radius_influenslengde = 0, double ↗
32             Rettlinje_i_negativ_retning = 0, double Rettlinje_i_positiv_retning = ↗
33             0) // gitt en default verdi 0
34         {
35             string text1 = Punkt.ToString(); //Endre Punkt til tekst
36             string text2 = Tverrplassering.ToString();
37             string text3 = Eksentrisitet.ToString();
38             string text4 = DVS.ToString();
39             string text5 = Vertikal_radius.ToString();
40             string text6 = Radius_influenslengde.ToString();
41             string text7 = Rettlinje_i_negativ_retning.ToString();
42             string text8 = Rettlinje_i_positiv_retning.ToString();
43
44             List<string> name = new List<string>();
45
46             if (text1 == "-1")
47             {
48                 name.Add(" ");
49             }
50         }
51     }
52 }
```

```
36     }
37     else
38     {
39         name.Add("      " + text1 + "      " + text2 + "      " + ?
                text3 + "      " + text4 + "      " + text5 + "      " ?
                + text6 + "      " + text7 + "      " + text8);
40
41     }
42
43
44     return name;
45
46     }
47 }
48 }
49
50
51
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace Spennarmering_detalj
6 {
7     public class Kabelegenskaper
8     {
9
10        public static List<string> Input(string Oppspenningside, string      ↗
            Oppspenningsprosedyre, double Friksjonssikkerhetsfaktor, int      ↗
            Antall_enkle_kabler)
11        //PSIG sier hvor man skal spenne opp fra RI=høyre, LE=venstre,RILE=først ↗
            høyre så venstre, LERI først venstre så høyre
12        // TRWS betyr Tesioning, releasing, restressing, slip.
13        //KAPA friksjon sikkerhetsfaktor
14        {
15            List<string> name = new List<string>();
16            name.Add("PSIG " + Oppspenningside + " " + Oppspenningsprosedyre + ↗
                "' KAPA - K3 - ");
17            name.Add("$ KAPA - set according the code - see manual");
18            name.Add("$ K3 - if sigma at t-infinite is too high: guess necessary ↗
                SIGM and input it, e.g. 1220 N / mm2");
19            name.Add("$ tendon definition");
20            name.Add("$ Construction stage date; ICS1 = placing + stressing, ↗
                ICS2 = grouting(bonding)");
21            name.Add("CS ICS1 11 ICS2 12 0");
22            name.Add("TEND NOT 1 NOG 1 NTEN "+Antall_enkle_kabler+" LC 11");
23            name.Add("END");
24            name.Add("END");
25            return name;
26        }
27    }
28 }
29
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Autodesk.DesignScript.Runtime;
7 using Autodesk.DesignScript.Interfaces;
8 using Autodesk.DesignScript.Geometry;
9
10 namespace Spennføringsforslag
11 {
12     public class Spennføringsforslag
13     {
14         public static List<string> Input(int Materialnummer, int Kabeltype,
15             string Referanseakse, double Startpunkt, double Sluttpunkt,
16             List<double> Spenn, double Profilhøyde, double Overdekking, int
17             Antall_kabler)
18         {
19             // kabeltype er gitt fra Sofistik, BBV ETA NO er en standard type så
20             // kan man velge mellom ulike her, Standard ifølge SVV668
21             // Material bestemmes også fra sofistik slik at man etterpå velger
22             // nummeret referet til materialet i sofistik hvis man har importert
23             // geometri og lagt til material
24             //spennkabel nummer er for å kunne gi høyde for x antall kabler og
25             // nummerer sådan. alternativ ta spennkabelnummer som en variabel n og
26             // si antall kabler og da generere flere tekstfiler.
27
28             List<string> name = new List<string>();
29             name.Add("+PROG TENDON $ Spennkabelforslag");
30             name.Add("HEAD Spennarmeringsforsalg");
31             name.Add("PAGE UNII 0");
32             name.Add("SYSP NOPS 1 COMP BBV ETA NO " + Kabeltype + " MAT " +
33                 Materialnummer);
34             name.Add("$ GEOMETRY Noh er nummeret til kabelen og etter auto skal
35                 man skrive inn nodenummeret man etablere referanseakse");
36
37             int n = Spenn.Count; //antall iterationer
38             double PF = Profilhøyde/2;
39             name.Add("AXES NOH 1 TYPE REFX '" + Referanseakse + "'");
40             name.Add("TOPP NOH 1 SP " + 0 + " KIND REFX S " +
41                 Startpunkt); //første punkte
42             for (int i = 0; i < n; i++)
43             {
44                 name.Add("TOPP NOH 1 SP " + (i + 1) + " KIND REFX S " +
45                     ((Spenn[i]))); //Lager n-antall spennoppdeling
46             }
47             name.Add("TOPP NOH 1 SP " + (n + 1) + " KIND REFX S " +
```

```

        Sluttpunkt); //Siste punkt
38     name.Add("T GEO NOG 1 NOH 1 NOPS 1 TITL 'Spennkabelforslag");
39     name.Add("PTUV S      U      V      DVS      RV      RL      ↗
        DL      DR      TYPE=SPAN $ TYPE = SPAN = xi - spaninput via top ↗
        points");
40     name.Add("      0      0      0");
41     name.Add("      0.04      0      0");
42
43     for (int i = 0; i <= n; i++)
44     {
45     name.Add("      " + (i + 0.5) + "      0      " + ((PF - ↗
        Overdekkning)/1000));
46     }
47     for (int i = 1; i <= n; i++)
48     {
49     name.Add("      " + (i) + "      0      " + "-" + ((PF - ↗
        Overdekkning)/1000));
50     }
51
52     name.Add("      " + (n + 0.8) + "      "+"0      0");
53     name.Add("      " + (n + 1.0) + "      "+"0      0");
54     name.Add("PSIG RI 'TRWS' KAPA - K3 -");
55     name.Add("$ KAPA - set according the code - see manual");
56     name.Add("$ K3 - if sigma at t-infinite is too high: guess necessary ↗
        SIGM and input it, e.g. 1220 N / mm2");
57     name.Add("$ tendon definition");
58     name.Add("$ Construction stage date; ICS1 = placing + stressing, ↗
        ICS2 = grouting(bonding)");
59     name.Add("CS ICS1 11 ICS2 12 0");
60     name.Add("TEND NOT 1 NOG 1 NTEN " + Antall_kabler + " LC 11");
61     name.Add("END");
62     name.Add("END");
63     return name;
64 }
65 }
66 }
67
68

```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Autodesk.DesignScript.Runtime;
7 using Autodesk.DesignScript.Interfaces;
8 using Autodesk.DesignScript.Geometry;
9
10 namespace Eksentrisitetsbasert
11 {
12     public class Brukervennlig_eksentristietbestemmelse
13     {
14         public static List<string> Input(int Materialnummer, int Kabeltype,
15             string Referanseakse, double Startpunkt, double Slutt punkt,
16             List<double> Spenn, List<double> Plassering, List<double>
17             Eksentrisitet, string Oppspennings side, int Antall_kabler)
18         {
19             List<string> name = new List<string>();
20             name.Add("+PROG TENDON $ Spennkabel forslag");
21             name.Add("HEAD Spennarmerings forsalg");
22             name.Add("PAGE UNII 0");
23             name.Add("SYSP NOPS 1 COMP BBV ETA NO " + Kabeltype + " MAT " +
24                 Materialnummer);
25             name.Add("$ GEOMETRY Noh er nummeret til kabelen og etter auto skal
26                 man skrive inn nodenummeret man etablere referanseakse");
27
28             int n = Spenn.Count; //antall iterationer
29             name.Add("AXES NOH 1 TYPE REF S " + Referanseakse + "");
30             name.Add("TOPP NOH 1 SP " + 0 + " KIND REF S " +
31                 Startpunkt); //første punkte
32             for (int i = 0; i < n; i++)
33             {
34                 name.Add("TOPP NOH 1 SP " + (i + 1) + " KIND REF S " +
35                     ((Spenn[i])); //Lager n-antall spennoppdeling
36             }
37             name.Add("TOPP NOH 1 SP " + (n + 1) + " KIND REF S " +
38                 Slutt punkt); //Siste punktet
39             name.Add("TGEO NOG 1 NOH 1 NOPS 1 TITL 'Spennkabel forslag");
40             name.Add("PTUV S U V DVS RV RL
41                 DL DR TYPE=SPAN $ TYPE = SPAN = xi - spaninput via top
42                 points");
43             int p = Plassering.Count;
44             for (int i = 0; i < p; i++)
45             {
46                 name.Add(" " + Plassering[i] + " 0 " +
47                     Eksentrisitet[i] + " 0 0 0");
48             }
49         }
50     }
51 }
```

```
                                0          0");
39         }
40         name.Add("PSIG " + Oppspenningside + " 'TRWS' KAPA - K3 -");
41         name.Add("$ KAPA - set according the code - see manual");
42         name.Add("$ K3 - if sigma at t-infinite is too high: guess necessary ↗
           SIGM and input it, e.g. 1220 N / mm2");
43         name.Add("$ tendon definition");
44         name.Add("$ Construction stage date; ICS1 = placing + stressing, ↗
           ICS2 = grouting(bonding)");
45         name.Add("CS ICS1 11 ICS2 12 0");
46         name.Add("TEND NOT 1 NOG 1 NTEN " + Antall_kabler + " LC 11");
47         name.Add("END");
48         name.Add("END");
49         return name;
50     }
51 }
52
53 }
54
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Autodesk.DesignScript.Runtime;
7 using Autodesk.DesignScript.Interfaces;
8 using Autodesk.DesignScript.Geometry;
9 namespace Temperaturlast
10 {
11     public class Temperaturlast
12     {
13         public static List<string> Input(string Gruppenummer, double T_ext,
14             double T_con, double T_heat, double T_cool)
15         {
16             List<string> name = new List<string>();
17             name.Add("+PROG SOFILOAD");
18             name.Add("HEAD Temperatur laster");
19             name.Add(" ");
20             name.Add("$legg inn temperaturlastene, hvor dt er jevnt fordelt dtz
21                 lineært varierende");
22             name.Add("LC 120 TYPE NONE TITL 'Temperatur sommer +dT'");
23             name.Add("BEAM GRP " + Gruppenummer + " TYPE DT " + T_ext); //legge
24                 inn dt for sommertemperatur
25             name.Add(" ");
26             name.Add("LC 121 TYPE NONE TITL 'Temperatur vinter -dT'");
27             name.Add("BEAM GRP " + Gruppenummer + " TYPE DT " + T_con); //legge
28                 til dt for vintertemperatur
29             name.Add(" ");
30             name.Add("LC 122 TYPE NONE TITL 'Temperaturgradient +dTz'");
31             name.Add("BEAM GRP " + Gruppenummer + " TYPE DTZ " + T_heat); //
32                 temperaturgradient sommertid +dtz
33             name.Add(" ");
34             name.Add("LC 123 TYPE NONE TITL 'Temperaturgradient -dTz'");
35             name.Add("BEAM GRP " + Gruppenummer + " TYPE DTZ " + T_cool); //
36                 temperaturgradient -dtz
37             name.Add("END");
38             name.Add(" ");
39             name.Add("+PROG SOFILOAD");
40             name.Add("HEAD Temperatur lastkombinasjon");
41             name.Add(" ");
42             name.Add("$Kombinere lastene med wn=0.35 og wm=0.75 inn i 8
43                 kombinasjoner."); //kombinasjonen wn og wm gjøres, se formel
44                 [6.1.5]
45             name.Add("LC 130 TYPE T_D TITL 'Kombinasjon 1'; COPY
46                 122 ;COPY 120 FACT 0.35"); //kombinasjon 1
47             name.Add("LC 131 TYPE T_D TITL 'Kombinasjon 2'; COPY 122 FACT
48                 0.75 ;COPY 120"); //Kombinasjon 2
```

```
...01.Bru\Temperatur\Temperaturlast\Temperaturlast\Class1.cs 2
40     name.Add(" ");
41     name.Add("LC 132 TYPE T_D TITL 'Kombinasjon 3'; COPY           ↗
        122           ;COPY 121 FACT 0.35"); //kombinasjon 3
42     name.Add("LC 133 TYPE T_D TITL 'Kombinasjon 4'; COPY 122 FACT ↗
        0.75 ;COPY 121"); //Kombinasjon 4
43     name.Add(" ");
44     name.Add("LC 134 TYPE T_D TITL 'Kombinasjon 5'; COPY           ↗
        123           ;COPY 120 FACT 0.35"); //kombinasjon 5
45     name.Add("LC 135 TYPE T_D TITL 'Kombinasjon 6'; COPY 123 FACT ↗
        0.75 ;COPY 120"); //Kombinasjon 6
46     name.Add(" ");
47     name.Add("LC 136 TYPE T_D TITL 'Kombinasjon 7'; COPY           ↗
        123           ;COPY 121 FACT 0.35"); //kombinasjon 7
48     name.Add("LC 137 TYPE T_D TITL 'Kombinasjon 8'; COPY 123 FACT ↗
        0.75 ;COPY 121"); //Kombinasjon 8
49     name.Add("END");
50     return name;
51     //muligens mulig å gjøre dette enda penere med loop.
52 }
53 }
54 }
55
56
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Autodesk.DesignScript.Runtime;
7 using Autodesk.DesignScript.Interfaces;
8 using Autodesk.DesignScript.Geometry;
9 namespace Aksevalg
10 {
11     public class Akse
12     {
13         public static List<string> Input(string Akseavn, List<Point> Punkter)
14         {
15             List<double> x = new List<double>();
16             {
17                 for (int i = 0; i < Punkter.Count; i++)
18                     x.Add(Punkter[i].X);
19             }
20             List<double> y = new List<double>();
21             {
22                 int n = Punkter.Count;
23                 for (int i = 0; i < n; i++)
24                     y.Add(Punkter[i].Y);
25             }
26             List<double> z = new List<double>();
27             {
28                 int n = Punkter.Count;
29                 for (int i = 0; i < n; i++)
30                     z.Add(Punkter[i].Z);
31             }
32         }
33
34
35         List<string> name = new List<string>();
36         name.Add("+PROG SOFIMSHC $ Akseoppsettelse");
37         name.Add("HEAD Definisjon av en akse for SOFILOAD load trains eller ↗
38             SSD Tendons");
39         name.Add("PAGE UNII 0");
40         name.Add("SYST REST");
41         name.Add("GAX " + "" + Akseavn + "" + " TYPC POLY"); //kan være ↗
42             ulike typer, poly gir rettlinjer.
43         for (int i = 0; i < Punkter.Count; i++)
44             name.Add("GAXC X " + x[i]/1000 + " Y " + y[i]/1000 + " Z " + z
45                 [i]/1000); //lager en akse fra flere punkter
46         name.Add("END");
47
48         return name;
49         //Punktene fra dynamo må reduseres med en tusenedel for å bli i lik ↗
```



```
        koordinater gitt i sofistik.
47     }
48 }
49 }
50 //Alternativ metode
51 /* public static List<string> Input(string Akseavn, Point p1,Point p2 )
52 {
53     double x1 = p1.X;
54     double y1 = p1.Y;
55     double z1 = p1.Z;
56     double x2 = p2.X;
57     double y2 = p2.Y;
58     double z2 = p2.Z;
59
60     List<string> name = new List<string>();
61     name.Add("+PROG SOFIMSHC $ Akseoppsettelse");
62     name.Add("HEAD Definisjon av en akse for SOFILOAD load trains eller SSD   ➤
        Tendons");
63     name.Add("PAGE UNII 0");
64     name.Add("SYST REST");
65     name.Add("GAX " + "" + Akseavn + "");
66     name.Add("GAXB X1 " + x1 / 1000 + " " + y1 / 1000 + " " + z1 / 1000 + " X2   ➤
        " + x2 / 1000 + " " + y2 / 1000 + " " + z2 / 1000); //lager en akse fra   ➤
        punkt 1 til punkt 2
67     name.Add("END");
68
69     return name;
70     //Punktene fra dynamo må reduseres med en tusenedel for å bli i lik   ➤
        koordinater gitt i sofistik.
71
72 }
73
74
75 }
76 }
77 */
78
```

J Pythonskript for linjelaster

Pythonskript for linjelaster. Hosted og mellom punkt.

```
1 import clr
2 clr.AddReference('ProtoGeometry')
3 from Autodesk.DesignScript.Geometry import *
4 clr.AddReference('RevitAPI')
5 import Autodesk
6 from Autodesk.Revit.DB import *
7 from Autodesk.Revit.DB.Structure import *
8 from System.Collections.Generic import *
9
10 clr.AddReference("RevitNodes")
11 import Revit
12 clr.ImportExtensions(Revit.Elements)
13 clr.ImportExtensions(Revit.GeometryConversion)
14
15 clr.AddReference("RevitServices")
16 import RevitServices
17 from RevitServices.Persistence import DocumentManager
18 from RevitServices.Transactions import TransactionManager
19 from System.Collections.Generic import *
20
21 lineLoads = []
22
23 host = IN[0]
24 force = IN[1].ToXYZ()
25 moment = IN[2].ToXYZ()
26 LLType = UnwrapElement(IN[3])
27
28 for i in host:
29     doc = DocumentManager.Instance.CurrentDBDocument
30     TransactionManager.Instance.EnsureInTransaction(doc)
31     l = LineLoad.Create(doc, UnwrapElement
32         (i), force, moment, LLType)
33     lineLoads.append(l)
34
35 TransactionManager.Instance.TransactionTaskDone()
36 OUT = lineLoads
```

```
1 import clr
2 clr.AddReference('ProtoGeometry')
3 from Autodesk.DesignScript.Geometry import *
4
5 clr.AddReference('RevitAPI')
6 import Autodesk
7 from Autodesk.Revit.DB import *
8 from Autodesk.Revit.DB.Structure import *
9 from System.Collections.Generic import *
10
11 clr.AddReference("RevitNodes")
12 import Revit
13 clr.ImportExtensions(Revit.Elements)
14 clr.ImportExtensions(Revit.GeometryConversion)
15
16 clr.AddReference("RevitServices")
17 import RevitServices
18 from RevitServices.Persistence import DocumentManager
19 from RevitServices.Transactions import TransactionManager
20 from System.Collections.Generic import *
21 doc = DocumentManager.Instance.CurrentDBDocument
22 lineLoads=[]
23
24 Pkt = UnwrapElement(IN[0])
25 force = IN[1].ToXYZ()
26 moment = IN[2].ToXYZ()
27 LLType = UnwrapElement(IN[3])
28 level = UnwrapElement(IN[4])
29
30 TransactionManager.Instance.EnsureInTransaction(doc)
31 for p in Pkt:
32     l = LineLoad.Create(doc,p[0].ToXYZ(),p[1].ToXYZ
33         (), force, moment, LLType, level)
34     lineLoads.append(l.ToDSType(False))
35 TransactionManager.Instance.TransactionTaskDone()
36 #Assign your output to the OUT variable.
37 OUT = lineLoads
```

K Tilleggsfiler

Dynamo_geometri - Geometrioppbygning av brumodell i Dynamo

Dynamo_laster - Lastskript for brumodell i Dynamo

Bru_parametrisk - Sofistik filen til parametrisk bru

Bru_parametrisk_ - Revit filen til parametrisk bru

Cowi_bru_nyeste - Sofiplus filen for tradisjonell modell (dwg)

Cowi_bru_nyeste - Sofistik filen for tradisjonell modell