

## RIB prosjektering ved implementering av parametrisk verktøy

Prosjektering av reviderbare brudekker, med en arbeidsflyt mellom Grasshopper, Revit og FEM-design.

AMALIE AAMODT



### VEILEDERE

Robert Stenersen McShane, Dagfin Skaar AS  
Dmitry Vysochinskiy, UiA

**Universitetet i Agder, 2021**

Fakultet for teknologi og realfag  
Institutt for ingeniørvitenskap



## Obligatorisk egenerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Jeg erklærer herved at min besvarelse er mitt eget arbeid, og at jeg ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	<input checked="" type="checkbox"/>
2.	Jeg erklærer videre at denne besvarelsen: <ul style="list-style-type: none"> <li>- ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.</li> <li>- ikke refererer til andres arbeid uten at det er oppgitt.</li> <li>- ikke refererer til eget tidligere arbeid uten at det er oppgitt.</li> <li>- har alle referansene oppgitt i litteraturlisten.</li> <li>- ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.</li> </ul>	<input checked="" type="checkbox"/>
3.	Jeg er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	<input checked="" type="checkbox"/>
4.	Jeg er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	<input checked="" type="checkbox"/>
5.	Jeg er kjent med at Universitetet i Agder vil behandle alle saker hvor det foreligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	<input checked="" type="checkbox"/>
6.	Jeg har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	<input checked="" type="checkbox"/>

## Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).

Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage Aura og på UiA sine nettsider med forfatter(ne)s godkjenning.

Oppgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Jeg gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:

JA  NEI

Er oppgaven båndlagt (konfidensiell)?  
(Båndleggingsavtale må fylles ut)

JA  NEI

- Hvis ja:

Kan oppgaven publiseres når båndleggingsperioden er over?

JA  NEI

Er oppgaven unntatt offentlighet?

JA  NEI

(inneholder taushetsbelagt informasjon. Jfr. Offl. §13/Fvl. §13)

## Forord

Denne masteroppgaven er utarbeidet ved instituttet for ingeniørvitenskap, som en del av masterprogrammet for Byggkonstruksjon ved Universitetet i Agder. Masteroppgaven er den avsluttende oppgaven i emnet BYG508, og ble utarbeidet i det fjerde og siste semesteret.

Jeg vil benytte anledningen til å takke intern veileder ved Universitet i Agder, Dmitry Vysochinsky, for god oppfølging av fremdrift og tilbakemeldinger. Jeg vil også rette en takk til ekstern veileder ved Dagfin Skaar AS, Robert Stenersen McShane, for all veiledning og hjelp til å finne gode kontaktpersoner og faglig innsikt. I tillegg vil jeg takke Jonas Høgli Major og Jon Lunden Hustad for at dere har bistått med egne erfaringer innenfor bruk av egne dataprogrammer, som arkitekt og mekatronikkingeniør. Til sist vil jeg vil jeg rekke en stor takk til Renata Carolina Ribeiro Portelada, som har vært så generøs og satt av tid til å besvare spørsmål relatert til parametrisk design, samt bruk av programvaren Grasshopper.

Oppgaven er utarbeidet i samarbeid med bedriften Dagfin Skaar AS, og har som mål å benytte parametrisk design i prosjekteringsprosessen til brukonstruksjoner. Her er også målet å benytte en arbeidsflyt som samler arbeidet på en plass, uten å produsere separate CAD- og FEM-modeller. Der er valgt ut at brudekker er hovedelementet som skal prosjekteres. Til prosjekteringen tas det i bruk det parametriske verktøyet Grasshopper, samt to de tradisjonelle modellerings- og beregningsverktøy, hos DS: Revit og FEM-Design. Oppgaven bygger videre på forprosjektet som ble gjennomført høsten 2020.

## Summary

The process of development for computer programs, is modified and updated continuously. Just in the last twenty years, both modelling- and calculation tools have developed to be able to handle and produce 3D models. Today, it is also possible to use parametric design for these work assignments. Parametric design produces algorithms to define the work tasks, that are to be done in the modelling- and calculation programs.

This master thesis is therefore investigating how parametric modeling can generate a more efficient revision process, of the geometric- and static model, for design of bridge decks. To accomplish this, a study of literature, gaining experience from mechatronics engineering, and parametric designing was carried out. According to this, it has been concluded that parametric design can achieve a workflow that avoids the import and export of CAD- and FEM models to achieve interaction. This workflow provides the opportunity to generate an efficient revision process of both models, by implementing parametric design in the early phase of the project. The reason for this, is that not all of the design elements were possible to produce parametrically, with Grasshopper. At the same time as the revised content, in most cases, indicate that the models must be produced again. This outcome destroys the modeled products of reinforcement, dimensions, and Revit-sections, because they have to be produced in either Revit or FEM-Design. They depend on the products, produced with Grasshopper.

## Innholdsfortegnelse

Obligatorisk egenerklæring .....	i
Publiseringsavtale.....	ii
Forord .....	iii
Summary .....	iv
Innholdsfortegnelse .....	v
Figurliste .....	vii
Tabelliste .....	ix
Begrepsliste .....	x
Forkortelser .....	x
1 Innledning.....	1
2 Samfunnsperspektiv.....	2
3 Teoribakgrunn .....	4
3.1 Dataassistert utforming.....	4
3.1.1 Utvikling.....	5
3.1.2 BIM .....	5
3.1.3 Autodesk Revit.....	6
3.2 Finite Element Method.....	9
3.2.1 Linjeelement .....	10
3.2.2 StruSoft FEM-Design.....	10
3.3 Samhandling av CAD og FEM.....	13
3.3.1 StruXML Revit Add-In .....	13
3.4 Parametrisk design /modellering .....	14
3.4.1 Visuell programmering .....	15
3.4.2 Grasshopper3D.....	16
4 Forskerspørsmål .....	20
4.1 Avgrensninger.....	20
5 Case .....	21
5.1 Programvarene.....	21
5.1.1 Revit.....	21
5.1.2 FEM-Design-3D structure .....	24
5.1.3 Grasshopper .....	29
5.1.4 Rhino.inside.revit.....	33
5.2 DS prosjektering .....	35
5.2.1 Prosjekteringsprosessen.....	35
5.2.2 Samhandling mellom Revit og FEM-Design.....	36

5.2.3	Utfordringer med tradisjonell modellering .....	37
5.3	Brudekket .....	38
5.3.1	Sagåne bru .....	38
5.3.2	Idealisert modell /Statisk modell.....	40
5.3.3	Inndata til lastdimensjonering.....	41
6	Metode .....	43
6.1	Litteraturstudie.....	43
6.2	Innhentet erfaringer fra mekatronikkingeniøren.....	45
6.3	Anvendte programvarer og opplæringsprosessen.....	46
6.3.1	Opplæring.....	47
6.4	Parametrisk prosjektering .....	49
6.4.1	Valg av variable parametere .....	52
6.4.2	Valg av prosjekteringselementer.....	53
6.4.3	Universelle grupper .....	55
6.4.4	Revit modeller .....	59
6.4.5	FEM-Design modeller .....	64
7	Resultat.....	71
7.1	Grenseindustrien mekatronikk.....	71
7.2	Litteraturstudie formål to.....	72
7.3	Parametrisk prosjektering .....	75
7.3.1	Grasshopper skript .....	75
7.3.2	Arbeidsflyten for produksjon .....	76
7.3.3	Grasshopper produktene .....	79
7.3.4	Parametriske revisjonsprosess .....	92
8	Diskusjon .....	98
8.1	Litteraturstudiet .....	98
8.1.1	Svakheter ved valgt metode.....	98
8.2	Parametrisk prosjektering .....	98
8.2.1	Beregningsproduktene .....	99
8.2.2	Revisjonsprosessen .....	103
8.2.3	Arbeidsflyt .....	105
8.2.4	Svakheter ved valgt metode/case .....	106
9	Konklusjon .....	108
10	Anbefalinger .....	110
	Referanser .....	111
	Vedlegg.....	117



## Figurliste

Figur 2.1: Forutsetningene i det digitale veikartet [8, s.6] .....	3
Figur 3.1: Brukergrensesnittet til Rvt [3, Figur 3.6] .....	7
Figur 3.2: Rvt hierarkiet, med vegg som eksempel [22] .....	7
Figur 3.3: Tre parameter-typer i Rvt [22] .....	9
Figur 3.4: Eksempler på linjeelementer i FEM, [24, bilde 3.1] .....	10
Figur 3.5: Modulene til Fd .....	11
Figur 3.6: Brukergrensesnitt Fd – 3D Structure [28, s.5] .....	12
Figur 3.7: Alternative arbeidsmetoder for å produsere en FEM-modell [24, s. 231] .....	13
Figur 3.8: Verktøyene i fanen StruSoft i Rvt .....	14
Figur 3.9: Prosessen av å tegne sinuskurve i Python og med komponenter i GH [1, s.6] .....	15
Figur 3.10: Få frem Grasshopperviduet fra Rhino .....	16
Figur 3.11: Brukergrensesnittet til GH [44], laget i henhold til [1, s.15] .....	17
Figur 3.12: Kategoriseringen av komponent-paletter i GH .....	17
Figur 3.13: Ekstra komponenter i GH, ved nedlastning av RiR [44] .....	18
Figur 3.14: Forhåndsvisning av GH-data i Rvt [51] .....	19
Figur 3.15: Farge og merknad i Grasshopper på Revit-ikoner [2] .....	19
Figur 5.1: Familie-maler.....	22
Figur 5.2: Sette «Label» på målsetting i en familie .....	22
Figur 5.3: Kontroll av «Label» opprettede parametere, til egendefinert familie i Rvt.....	23
Figur 5.4: Armert bjelke, snittvisning .....	23
Figur 5.5: "Free Form" armering i Rvt .....	24
Figur 5.6: Tverrsnitt som strukturelle objekter i Fd .....	25
Figur 5.7: Beregninger i fanen "Analysis" og "RC design", figur utarbeidet fra [62] .....	26
Figur 5.8: Manuell armering i Fd .....	27
Figur 5.9: Utformingsparametere til armeringsanalysen i Fd .....	28
Figur 5.10: Auto design parametere til et brudekke i Fd .....	28
Figur 5.11: En av metodene for beregning og modellering av armering i Fd.....	29
Figur 5.12: Oppbygningen til komponenter, med CNR komponenten som eksempel [1, s.41] .....	29
Figur 5.13: Eksempel på «Zoomable User Interface» funksjonen på en komponenter i GH [1, s.46] ..	30
Figur 5.14: Eksempler på parametere i Grasshopper.....	30
Figur 5.15: Tre-grener [1, kap.1.5.2] .....	31
Figur 5.16: Data-tre med "Panel" komponenten [1, kap.1.5.2] .....	31
Figur 5.17: Plasseringen av Python komponenten i GH, som standardisert komponent.[66] .....	32
Figur 5.18: Måter å gruppere komponenter[44].....	32
Figur 5.19: Hvordan endre enheter i GH i Rhino «Document Properties» menyen [1, s.32] .....	33
Figur 5.20: Finne elementers ID og bruke dem i GH .....	33
Figur 5.21: Bruk av komponenten «Element Geometry» .....	34
Figur 5.22: Bruk av komponenten «Set Element Parameter» .....	34
Figur 5.23: Hente ut kategorier og typer fra Rvt til GH .....	34
Figur 5.24: Fasene som prosjekteringsprosessen inneholder [3, Figur7.5] .....	35
Figur 5.25: Tegningsprodukter i skisse- og forprosjekt, [3, Figur 7.6] .....	36
Figur 5.26: Tegningsprodukter i detalj og sluttfasen, [3, Figur 7.7-7.8] .....	36
Figur 5.27: Analytisk modell versus visuell fremvisning, figur 7.12 og 7.14 hentet fra [3, s.44-45] .....	37
Figur 5.28: Hva som inkluderes fra Sagåne bru, bilde hentet fra IFC-fil til Sagåne bru.....	39
Figur 5.29: Oppriss og snitt av Sagåne bru .....	39
Figur 5.30: FE-modell av Sagåne bru .....	40
Figur 5.31: Tverrbærer-forankringen til brudekket av Sagåne bru .....	40
Figur 6.1: Søkemotorer benyttet i litteraturstudiet [75, 76] .....	44
Figur 6.2: Informasjonsflyten/arbeidsflyten mellom GH, Rvt og Fd .....	50
Figur 6.3: Fremvisning av komponentene i GH .....	50

Figur 6.4: rekkefølgen på hvordan programvarene må åpnes.....	51
Figur 6.5: Grupperingen av gruppene i GH .....	52
Figur 6.6: Visuell fremvisning av utformingsmulighetene til brudekkets veilinje .....	55
Figur 6.7: Arbeidsflyt kurvene som definerer brudekkets føringslinjen .....	56
Figur 6.8: Logikken bak produksjon for gjeldene kurve .....	57
Figur 6.9: Transformere enhet fra prosent til grader .....	57
Figur 6.10: Eksempel på når "Gjeldende kurve" får naturlig tverfall .....	58
Figur 6.11: Komponenter for produksjon av nivåer i Rvt og Fd .....	58
Figur 6.12: Komponenter for produksjon av akser i Rvt og Fd.....	58
Figur 6.13: Komponenter for produksjon av bjelke-element i Rvt.....	59
Figur 6.14: Hente inn tverrsnittsform til brudekket fra Rvt .....	60
Figur 6.15: Hente ut parameterne og overstyre dem i Rvt .....	60
Figur 6.16: Armering med komponenten "AddDirectShape.ByGeometry" .....	61
Figur 6.17: Referansekurvene til tverrarmeringen.....	62
Figur 6.18: Bruk av Python-komponent for 3D modellering av armering i Rvt.....	63
Figur 6.19: Komponenter for produksjon av sekundærelementene i Rvt .....	63
Figur 6.20: Kodene som ble endret i komponenten til prosjektinnstillingene.....	64
Figur 6.21: Gruppen benyttet for produksjon av Fd-modell.....	65
Figur 6.22: Komponenter til produksjon av brudekke og tverrbærer i Fd .....	65
Figur 6.23: Komponenter for produksjon av opplegg i Fd .....	66
Figur 6.24: Komponentene for produksjon av lasttilfeller og lastkombinasjoner .....	67
Figur 6.25: Skriptoppsett for produksjon av lastkombinasjoner .....	68
Figur 6.26: Komponenter benyttet til produksjon av laster.....	69
Figur 6.27: Komponenter benyttet for å gjennomføre RC beregninger .....	70
Figur 7.1: Eksempel på preppet sammenstilling, modellert i Solidwork.....	71
Figur 7.2: Resulterende GH-skript .....	75
Figur 7.3: Hoveddelene til hvert av GH-skriptene, med tilhørende fargegrupper.....	76
Figur 7.4: Arbeidsflyt til den parametriske prosjekteringen .....	77
Figur 7.5: GH-skript for produksjon av Rvt-modellene .....	78
Figur 7.6: GH-skript for produksjon av Fd-modellene.....	79
Figur 7.7: Forsøk på å produsere Rvt-målsetting i GH, bildet er tatt fra 3.0 skriptet .....	79
Figur 7.8: Prosjektinnstillingene spesifisert i GH .....	80
Figur 7.9: Produktet brudekke og armering i RVT-modell 1.1 .....	80
Figur 7.10: Produktene av prosjekteringselementene fra GH, RVT-modell 1.1, 3D visning .....	81
Figur 7.11: Armeringsdetalj i start- og endepunkt til brudekket.....	82
Figur 7.12: «Egenskap paletten» til bjelke-elementet .....	82
Figur 7.13: Snitt 1 i RVT-modell 1.1, med detaljnivå Fine og Visuell stil «Consistent Colors» .....	82
Figur 7.14: Aktivere nivåene som plan-visninger i Rvt .....	83
Figur 7.15: Planvisning av Nivå 1 og 2, RVT-modell 1.1, detaljnivå «Fine» og Visuell stil «Consistent Colors».....	83
Figur 7.16: Akse 1 og 4 i henhold til brudekket.....	83
Figur 7.17: Egendefinert familiekomponent i GH, til produksjon av bjelke .....	84
Figur 7.18: Produktene av prosjekteringselementene fra GH, RVT-modell 3.1, 3D visning .....	85
Figur 7.19: Produktene brudekke og armering i RVT-modell 3.1.....	85
Figur 7.20: Snitt 1 i RVT-modell 3.1, med detaljnivå Fine og Visuell stil «Consistent Colors» .....	86
Figur 7.21: Unøyaktighet mellom akse 1 og 4 i henhold til avslutningen på brudekkene .....	86
Figur 7.22: Fd-modell 1.2, 3D visning og fysisk vest-visning .....	88
Figur 7.23: Fd-modell 2.2, 3D visning og fysisk vest-visning .....	89
Figur 7.24: Fd-modell 3.2, 3D visning og fysisk vest-visning .....	89
Figur 7.25: Resulterende lasttilfeller i Fd .....	90
Figur 7.26: Resulterende beregninger og resultater for designvalgene «Check» og «AutoDesign» ....	90

Figur 7.27: Manuell armering av tverrbærer og brudekke .....	92
Figur 7.28: Universelle variable parametere .....	92
Figur 7.29: Rvt-spesifikke variable parametere.....	93
Figur 7.30: FEM-spesifikke variable parametere.....	94
Figur 7.31: Hvordan revisjonene til RVT-modell 1.1 og 1.3 skal gjennomføres .....	96

## Tabelliste

Tabell 3.1: Fanemenyenes underkategorier i Fd .....	12
Tabell 5.1: To objektstyper i Fd, med definisjon og posisjon .....	24
Tabell 5.2: Noen av lasttypene i Fd 3D-Structure [61] .....	25
Tabell 5.3: Lastkombinasjoner og lastfaktorer [72] .....	42
Tabell 6.1: Synonymer til temaene i denne oppgaven.....	43
Tabell 6.2: Søk, søkeord, antall treff, potensielle treff og inkluderte masteroppgaver.....	45
Tabell 6.3: Programvarer benyttet, samt versjonstype, nedlastingsdato og lisenstype.....	46
Tabell 6.4: Kapitlene i "Guide" veilederen som ble gjennomgått .....	47
Tabell 6.5: Videoene gjennomgått på Youtube-kanalen til Aussi BIM Guru .....	48
Tabell 6.6: Relevant kunnskap hentet fra forumet til RiR for opplæring av produksjon av armering ..	49
Tabell 6.7: Norske definisjoner på ordinære Rvt og Fd definisjoner.....	53
Tabell 6.8: Rvt-elementene for parametrisert prosjektering med tilhørende objektstype beskrivelse ..	54
Tabell 6.9: Definisjonene av de ulike typene av lasttilfeller i Fd versus i GH .....	68
Tabell 7.1: Mål, programmer og arbeidsflyt til inkluderte masteroppgaver .....	72
Tabell 7.2: Metodene og konklusjonene til de inkluderte masteroppgavene .....	74
Tabell 7.3: Arbeidsflyt for produksjon av brudekker i hvert GH-skript .....	77
Tabell 7.4: Egenskapene til hvert av brudekkene i RVT-modell 3.1, i form av parametere og deres innhold.....	86
Tabell 7.5 Beregningstid til analysene, med «Check» og «AutoDesign», for de 4 tilfellene.....	91
Tabell 7.6: Utforminger på brudekket i henhold til mulighet for armering .....	91
Tabell 7.7: Revisjonsmuligheter i GH-skript for RVT-modellene .....	94
Tabell 7.8: Revisjonsmuligheter i GH-skript for Fd-modellene .....	96

## Begrepsliste

Family	Familie er en samling av objekter i Autodesk Revit, med samme grunnleggende egenskaper. Eksempel på familier er vegger, søyler og bjelker. [4, s. IV]
Family editor	Dette er et redigeringsprogram for å opprette eller gjøre endringer i eksisterende objekter/familie i Autodesk Revit Building programmet [4, s. IV]
Solid modellering	innebærer at objektet presenteres som solide kropper, i motsetning til overflatemodellering, der objektene ser ut som uendelig tynne skall. [4, s. IV]
Parameter	komponenter som inneholder redigerbare størrelser
Skript	det visuelle programmet som utarbeides i et parametrisk verktøy, ved å inneholde komponenter sammenkoblet med vaiere.
Prosjektering	Prosessen som gjøres for å ta beviste valg for modelleringen
Modellering	Gjenskape noe som er av ønske å skape.
Ø-størrelse	Diameteren på armeringsjernene
Boolsk operasjon	Et uttrykk som evaluerer en verdi; som sann eller usann. Uttrykket brukes i programmeringsspråk. [5]
Open source	Programmet må inneholde kilde koder og tillate distribusjon
Plugin	En programvareutvidelse (plugin eller addin) som en tilleggsmodul, utviklet for å tilby ekstra funksjonalitet til et eller flere programvarer.

## Forkortelser

AEC	Arkitekt, ingeniør og konstruksjon næringen
BIM	Building Information Modeling
CAD	Computer-Aided Design
DAK	Datassistert konstruksjon
VP	Visual Programming
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Method
ULS	Ultimate Limit State
SLS	Serviceability Limit State
API	Application Programming Interface
COM-teknologi	Component Object Model-teknologi

Rvt	Revit
Fd	FEM-Design
RiR	Rhino.Inside.Revit
GH	Grasshopper
Rhino	Rhinoceros

## 1 Innledning

Bruk av CAD-programvarer og FEM-beregningsverktøy er allerede godt etablert hos prosjekterende byggingeniører. Dette er for flere bedrifter, fremdeles en todelt prosess, ved at det produseres separate 3D modeller. En del av utviklingen innebærer å etablere integrerte modeller, som kan styres av et sett nøkkelparametere for å samle arbeidet på et sted. Parametrisk design er et verktøy som kan benyttes til dette formålet. Det benyttes til å minske avstanden mellom produksjonen i de ulike programvarene. Det kommer av at de generelt har mindre begrensninger, og benyttes i samsvar med tradisjonelle prosjekteringsprogram. Dette er et verktøy som allerede er tatt i bruk ved flere større rådgivende bedrifter i Norge, og blir foreløpig ansett som et konkurransefortrinn. Multiconsult er et eksempel på en slik bedrift.

For å kunne holde følge med utviklingen som skjer innenfor dataprogrammer, har bedriften Dagfin Skaar AS et ønske om å implementere parametrisk design i prosjekteringsprosessen. Interessen kommer også som følge av å selv kunne bruke dette verktøyet som et konkurransefortrinn, i håp om redusert tidsbruk i prosjektene, og bedre produkter/løsninger. Det har allerede blitt kartlagt hvilke prosjekter og hvor i prosjekteringsprosessen til Dagfin Skaar AS, som kan ha størst nytteverdi av et slikt verktøy. Implementering av parametrisk design har størst nytteverdi på konstruksjoner som består av krumninger i mer enn to plan, og som tas i bruk i hele prosjekteringsprosessen. Dette ble konklusjonen i forprosjektet, som ble gjennomført høsten 2020 [3, s.iv].

Problemstillingen i denne oppgaven, er utarbeidet i samarbeid med bedriften Dagfin Skaar AS. Den bygger videre på arbeidet som ble gjort i forprosjektet, og lyder som følger: Hvordan kan prosjekteringsprosessen til rådgivende byggingeniører effektivisere revisjonsprosessen til bruprosjekter, ved å implementere ett parametrisk verktøy? Denne problemstillingen skal studeres i henhold til prosjekteringsprosessen, hos bedriften Dagfin Skaar AS.

## 2 Samfunnsperspektiv

Digitalisering er en faktor som er med på å omforme samfunnet og næringer, både i privat- og offentlig sektor. Dette begrepet er blant annet knyttet til innovasjon innenfor bygg-, anlegg- og eiendomsnæringen (BAE næringen). I en rapport til Byggenæringens landsforening (BNL) fra 2017 [6, s.29], deles digitaliseringen inn i tre nivå. Det første nivået omhandler automatisering og effektivisering. Her blir teknologiske verktøy tatt i bruk, for å effektivisere arbeidet som allerede utføres i dag. Det andre nivået innebærer å gjennomføre eksisterende prosesser på en bedre og mer effektiv måte. En arbeidsprosess med ny rekkefølge, av andre aktører. Det tredje og siste nivået endrer gjennomføringen helt, hvor eksisterende grensesnitt og rammer nullstilles. Et eksempel på en slik digitalisering er taxiformen Uber, som betegnes som en radikal endring av forretningsmodellen til taxi-næringen. [6, s.29]

En av de utøvende aktørene i BAE-næringen, som har utviklet seg stort innenfor digitale verktøy, er arkitekter og rådgivere [7]. De har eksempelvis implementert BIM, som inngår i nivå én av digitaliseringen. Her er modellene digitale i prosjektsamarbeid. Men en utfordring med digitaliseringen i BAE-næringen, er at aktørene digitaliserer hver for seg. Dette fører til en fragmentert næring, hvor det er problemer med å få ut ønskede gevinster. Behovet for koordinasjon av fellesaktiviteter og øke tempoet av en digitalisering sammen, er derfor stort. [8, s.2]

Et eksempel på hvorfor fragmenteringen i næringene oppstår, kommer av at hver sektor har egne interesser for å ivareta konkurransefortrinnet sitt. Derfor kan eksempelvis endring av forretningsmodeller få store konsekvenser for de ulike aktørene i BAE-næringen. [6, s.29] Her har aktørene arkitekter og rådgivere, entreprenører og byggherren, ulike interesser. Arkitekter og rådgivere arbeider mot å få en kontrollert plan- og prosjekteringsprosess. Dette for å få størst mulig del av ressursene. Mens entreprenørene setter søkelys på mer involvering i tidligfasen, hvilket da går på bekostning av prosjekteringsteamet. Og for byggherre/byggeier kan det være mest hensiktsmessig å ha en utvikling av standardiserte konsepter, utenfor prosjektene. En slik utvikling vil kunne redusere prosjekteringsarbeidet i de enkelte prosjekter. Det vil også overføre større mengder arbeid til standardisering og industrialisering. Standardiserte konsepter forsterkes ytterligere, når overgangen går mot leverandørprosjektering. Dette vil utfordre prosjekterende aktører på timeverksforbruk. [6, s.29]

Selv om en slik endring av forretningsmodeller både utfordrer entreprenører, arkitekter og rådgivere, vil heldigitalisering i dagens arbeidsprosesser gi en ny grad av standardisering og industrialisering av løsninger. Det er derfor viktig at foreningene i bransjen starter en diskusjon om fremtidens verdikjende tjenester og forretningsmodeller allerede nå, for å forenkle møte med en usikker digital fremtid. [6, s.29]

En forutsetning for å få gevinster av digitalisering, er altså at næringene digitaliseres sammen. For å få til dette har BAE-næringen nå tatt initiativ til et felles digitalt veikart for å få i gang denne omstillingen sammen. [8, s.3] Det digitale veikartet beskriver fire forutsetninger som må ligge til grunn for at næringen skal kunne bli heldigitalisert. Disse er presentert i Figur 2.1, og omhandler etablering av en felles digital plattform, og tilrettelagte standarder og lovverk for digital samhandling. Deretter fremkommer det at bedriftene og utdanningsprogram skal ha kompetanseutvikling. Den

siste forutsetningen som skal til for å realisere gevinster, er å spre den beste praksisen av digitale arbeidsprosesser og forretningsmodeller. Dette skal gjøres for å kunne måle effekten av heldigitalisering. [8, s.6]



Figur 2.1: Forutsetningene i det digitale veikartet [8, s.6]

Veikartet er ikke bare et overordnet rammeverk for hvordan næringen må arbeide for å bli heldigital. Det omhandler også hvordan næringen må arbeide for å bli konkurransedyktig og bærekraftig i 2025. De spesifikke målene for veikartet er at det skal føre til kostnadsreduksjon, samt en halvering av byggetiden og klimautslippene. Dette er mål som er hentet fra strategien til Storbritannia «Construction 2025», hvilket er ambisiøse mål for digitaliseringen. [6, s.4] [8, s.3]

Behovet for å få en mer bærekraftig BAE-næring, kommer av at denne næringen ofte kalles 40% næringen. Dette begrepet betyr at de bruker 40% av all energi og alle ressurser, i tillegg til at de produserer omkring 40% av alt avfall årlig. Det er derfor viktig å redusere ressurser og forbruk, for å møte klimamålene til FN. Dette må skje i tillegg til at produksjonen må økes, grunnet befolkningsvekst og urbanisering. [6, s.11]

FNs bærekraftsmål har en felles arbeidsplan, som blant annet setter søkelys på å stoppe klimaendringene i verden innen 2030 [9]. Miljøaspektet i det digitale veikartet er med på å arbeide mot å nå spesifikt et av formålene til FNs bærekraftsmål. Dette er det niende målet, som innebærer et fokusområde om å løse de miljømessige utfordringene, med å ta i bruk innovasjon og teknologisk fremgang. [10]

Det som gjelder målet angående kostnadsreduksjon i det digitale veikartet, har en egen økonomisk motivasjon for BAE næringen. Da det er beregnet at den norske gevinsten er opp imot 100 milliarder kroner årlig, dersom næringene blir heldigital. Dette er spesifikt relatert til kostnadseffektivisering og verdiøkning. Dette estimatet er gjennomført i Storbritannia, med utgangspunkt i at det ble investert 422 milliarder kroner i BAE næringen, i 2016. En gevinst på 100 millioner kroner vil ikke bare komme BAE-næringen til gode, men hele felleskapet i Norge. [8, s.2]



### 3 Teoribakgrunn

I dette kapittelet presenteres teorigrunnlaget av utviklingsprosessen innfor datateknologien til AEC-næringene<sup>1</sup>. Her vil begrepene DAK, CAD, BIM og FEM beskrives, samt innebærende programvare for modellering-, beregning-, og parametriske gjennomføring inkluderes.

#### 3.1 Dataassistert utforming

Datamaskinen har en betydelig innvirkning på funksjoner til prosjektering og produksjon. I dag har mikroelektronikk gjort det mulig å utvikle datamaskiner som har mye datakraft, og som er pålitelige og økonomisk overkommelig. En digital datamaskin kan utføre matematiske og logiske beregninger i henhold til forhåndsbestemte instruksjoner. Og det er nettopp dette som beskriver et program i en datamaskin. [11, s.20] Her er programvarer for generelle formål oftest bygd opp med et veldefinert hierarki [11, s.47].

Mikroelektronikken som er utviklet i dag, er blant annet akseptable for mange CAD applikasjoner. CAD er her den engelske forkortelsen for Computer-Aided Design, og omhandler design ved bruk av en datamaskin. Her er datamaskinen den viktigste funksjonen til CAD-applikasjoner. Den norske betegnelsen som ble anvendt i starten av CAD-utviklingen, var den norske forkortelsen DAK. Dette er en forkortelse for Dataassistert konstruksjon, og var mest omtalt i litteratur før 2000 tallet. [12, s.8] [11, s.20] I denne rapporten vil begrepet CAD benyttes videre, da dette er den internasjonale definisjonen.

Det finnes et bredt utvalg av CAD-programmer, både enkle og komplekse. CAD programmer kan generere realistiske fremvisninger av konstruksjoner, samt klima-, belastnings- og lyssimuleringer. Dette avhenger av type program som benyttes. Produktene her produseres som to- og tredimensjonale tegninger. Derfor er dette et verktøy som er viktig for ingeniørfaget, både for bil-, konstruksjons-, anleggs- og arkitektteknikk [11, s.3-4]. [12, s.8]

For å håndtere CAD programmer, er det nødvendig å ha kunnskap om matematikk, datamaskin, beregningsteknikk, kommunikasjonsmetoder, samt tegningsferdigheter [11, s.xxv]. Programvarene baseres på kapasiteten til datamaskinen og programvareutviklingen [11, s.4]. Moderne CAD systemer er basert på interaktiv datagrafikk (ICG). ICG betegner et brukerorientert system hvor datamaskinen er utviklet til å skape, transformere, og fremvise data i form av bilder eller symboler. I mange systemer er bildene konstruert ut ifra elementer med basisgeometri, både i form av punkter, linjer og sirkler. Disse skal kunne modifiseres i henhold til kommandoer gjort av brukeren. Dette er modifiseringer som eksempelvis å forstørre- og redusere størrelse, flytte lokasjon og rotere. [11, s.3]

Det typiske ICG systemet består av en maskinvare og en programvare. Men i tillegg til dette kommer den menneskelige konstruktør, da ICG-system forstørrer mulighetene som kommer av brukeren. Dette betyr at konstruktøren utfører produksjonen i henhold til det menneskelige intellektuelle ferdigheter, mens datamaskinen produserer i henhold til kapasiteten den har. [11, s.4]

---

<sup>1</sup> AEC-næringene = Arkitekt, ingeniør og konstruksjon næringen

I AEC næringen innebærer begrepet CAD, utarbeidelse av designet som en automatisert prosess. I designformålet har denne teknologien erstattet manuell tegningsdokumentasjon. Her fungerer programvarene som et verktøy som igjen forenkler utarbeidelsen av konstruksjonsdokumenter, som muliggjør for å utforske designideer og visualisere konsepter. [13] Dette omfavner utførelse av prosjektering og produksjon [11, s.xxv]. Produksjonen av CAD tegninger øker produktiviteten og kvaliteten til konstruksjoner. I tillegg gir dokumentasjonen en bedre fremstilling, og dermed fremmer bedre kommunikasjon. [11, s.3-4] [11, s.17]

### 3.1.1 Utvikling

Parametriske ideer innen prosjekteringsmodellering var en vesentlig funksjon i det første CAD programmet. Dette var programvaren Sketchpad, som ble utviklet av Ivan Sutherland i 1962. Parametrisering ble også en del av de banebrytende CAD-systemene på begynnelsen av 1970 tallet. Visjonen til systemene som ble utviklet, var å konstruere en visuell 3D bygning ved å modellere alle bygningselementene. [14]

En ny bølge av 3D parametriske programvarer, kom på midten av 1980 tallet. Disse ble herav viktige programmer som ble en standard implementering, hos industriene: elektronikk, infrastruktur, romfart, marinteknikk og bilproduksjon. På dette tidspunktet foretrakk de fleste aktørene i AEC bransjen å kun implementere 2D CAD system på datamaskinen. Det tok oppimot 20 år for å få innført 3D. [15]

Fra 1993-2008 ble både datagrafikken og CAD-teknologien påvirket av prosjektering og produksjonsverktøyene, med en betydelig forbedring [11, s.xxv]. Det ble en slik forbedring, at populariteten av å ta i bruk 3D digitale CAD-modeller for ingeniørtilnærminger, har vært økene frem til 2016. Modellene ble et sentralt punkt i utviklingsprosessen, og den viktigste datakilden til bruk og utveksling mellom aktørene i AEC næringen. [15] På denne måten ble flere manuelle og arbeidskrevende oppgaver redusert. Dette i forbindelse med å registrere konstruksjonsdata, og konvertering til ulike samhandlende filformat, i tillegg til utveksling [16].

I dag benyttes moderne parametriske CAD-programmer, til å forbedre tilpasning og fleksibilitet av geometriske endringer, så vel som gjenbrukbarhet. Dette kom med implementeringen av BIM i enkelte CAD-programmer. Her kan endringer bli foretatt på en effektiv og relativ enkel måte. Dette er dermed et funksjonsbaserte parametriske modelleringsparadime. BIM er en engelsk forkortelse for «Building Information Modeling», og er en digital plattform som samler teknologi, prosessforbedringen og digital informasjon. Det forbedrer prosjektresultater, som gjelder like mye for byggeprosjekter som for byggeteknikk og infrastrukturprosjekter. [15, 17, s.2,7]

### 3.1.2 BIM

Ved å ta i bruk BIM kan datafiler produserer innhold av fysiske komponenter, som igjen produsere sofistikerte 3D datastyrte modeller. Disse inneholder en grafisk presentasjon av objekter. En slik utformingsmetode startet en helt ny måte å tenke og arbeide på i AEC næringen [18, s.10]. Men generelt har det ført til en utvikling hvor ingeniørindustriene, ser for seg en mulig digital revolusjon. [17, s.2,7] [15]

Det mest grunnleggende nivået av BIM, er at det omhandler en prosess som brukes til å utforme, forstå og demonstrere fysiske og funksjonelle egenskaper. Det gir ett innsyn i en visuell konstruksjon i tre dimensjoner, med en datastyrt modellbasis. BIM programmer på sitt mest avanserte, brukes til å simulere bygging og drift av konstruksjoner eller fabrikasjoner. [17, s.2]

Utformingsprosessen ved bruk av BIM, innebærer at modellkomponentene ikke lenger utarbeides som enkeltlinjer, men modelleres. Det er nettopp dette som ligger i ordet informasjonsmodellering, som betyr å arbeide med en virtuell modell. [18, s.10]

Nåtidens BIM-programmer er implementert med parametrisk parametere. Parametrisk i denne sammenhengen, innebærer at størrelse, plassering og relasjonene til elementene, defineres og styres i henhold til parametere. En parameter er kjent fra matematikken, hvor størrelser i et uttrykk tildeles forskjellige verdier. [18, s.11]

### 3.1.3 Autodesk Revit

Revit (Rvt) er en modelleringsprogramvare innenfor sjangeren BIM. Det er utviklet at Autodesk, og baserer seg på Microsoft.NET Framework 4.5 og DirectX. [19] Programvaren utarbeider 3D modeller i målestokk. Disse utarbeides visuelt, men enkelte elementer kan også utarbeides analytisk. Modeller utarbeides av objekter og tar i bruk sammensatte elementer som består av relasjoner og egenskaper. Det er dette som gjør Rvt til en informasjonsmodell, som er forkortelsen IM i BIM. Mens den andre delen av BIM i Rvt forbedrer dataflyten mellom de andre prosjekterende aktørene i et prosjekt. Dette ved å importere og eksportere IFC-filer. [18, s.11]

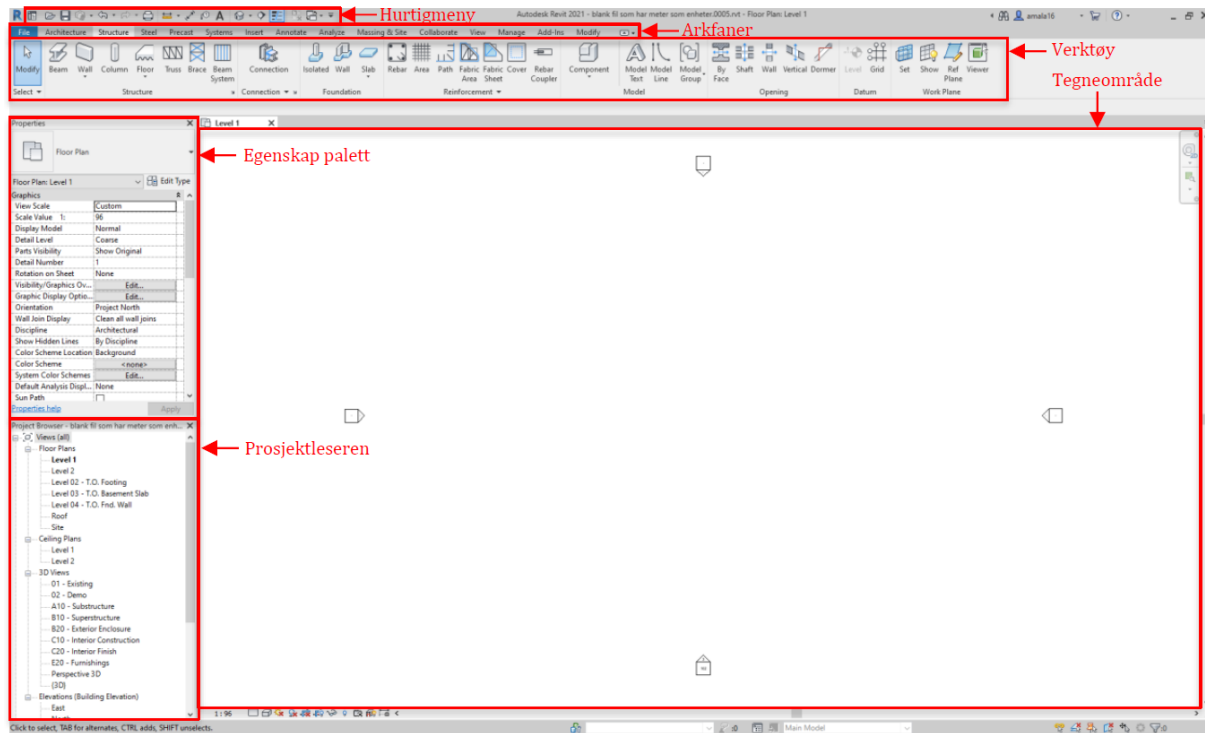
I tillegg til at Rvt kan dra nytte av IFC-filer, er programvaren utstyrt med et API<sup>2</sup> for å la eksterne applikasjonsutviklere integrere egne applikasjonene, i samsvar med Autodesk Rvt [20, kap.8.1]. Eksempler på applikasjoner som har tilgang på Rvt API-et, er verktøyene Dynamo og Grasshopper (GH), som utvikler informasjonsmodellering. Det kommer av at Rvt inneholder et grafisk programmeringsgrensesnitt med åpne kildekoder. Verktøyet er ment å utvide og optimalisere arbeidsflyten av BIM modelleringen i Rvt. [21]

#### **Bruker grensesnitt**

Rvt har et standardisert brukergrensesnitt, hvor noen av de grunnleggende elementene i organiseringen er påpekt i Figur 3.1. I tegneområdet presenteres utformet modell, samt ulike typer visninger (eksempelvis 3D- og snitt-visning). Øverst er hurtigmenyen plassert, og inneholder «ofte brukte» verktøy, og her er det tilpasningsmuligheter. Under hurtigmenyen ligger fanene, hvorav inneholder verktøylinjene til hver av dem. Verktøyene brukes for å produsere elementer i prosjektet. Alle elementer i Rvt består av et sett parametere som definerer egenskaper. Disse er synlige i en egenskap palett. Det som gjelder prosjektleseren, er at den har et innhold av blant annet alle visninger, mengder, ark og familier. Disse er listet opp med en hierarkisk logikk. [6, s.19]

---

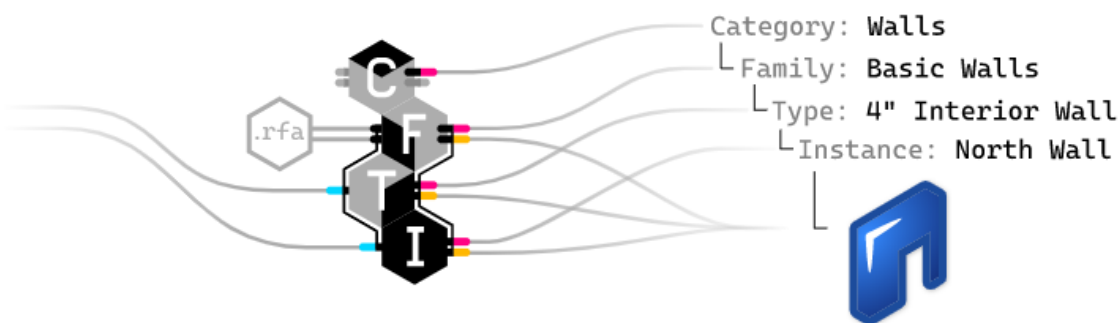
<sup>2</sup> Applikasjonsprogrammeringsgrensesnitt



Figur 3.1: Brukergrensesnittet til Rvt [3, Figur 3.6]

### Hierarki

Som nevnt over, modellerer Rvt frem en modell, fremfor å produsere enkeltlinjer [18, s.10]. Her består modellen av elementer, som den grunnleggende byggesteinen. Elementene er organisert i et strengt hierarkisk system. De er organisert ovenfra og ned, med følgende organiseringsnivåer: kategorier, familier, typer og forekomster [18, s.14, 29]. Et eksempel på denne organiseringen er presentert i Figur 3.2, som innebærer et veggelement. Denne organiseringen medfører at brukeren kan velge ett element og samtidig velge alle liknende elementer, basert på et spesifikt nivå i hierarkiet. Når elementet er utplassert i tegneområdet i Rvt, får det også tildelt en egen id. [22]



Figur 3.2: Rvt hierarkiet, med vegg som eksempel [22]

Øverst i hierarkiet er kategori. Kategoriene er forhåndsdefinert i programvaren. Det innebærer at kategorier ikke kan legges til, slettes eller tildeles nye navn. Rvt inneholder et utvalg av kategorier som inkluderer alle elementer som utgjør en visuell byggekropp. Dette kan eksempelvis være bjelke, søyle, tak, vegg og vindu. [23] [22] [18, s.14, 29]

Familie er det neste organiseringsnivået, under kategori [22]. Alle RVT-elementer tilhører en familie [23], og defineres som en gruppe elementer som har like parametere [18, s.14, 29]. Den er dermed lik en funksjon  $f(x)$ , som tar imot data fra type- og forekomstparametere (Instance parameters på engelsk) [23]. Familiene er både 2D og 3D, samt parametriske eller ikke-parametriske. Parametere benyttes for å ha mulighet til å endre eksempelvis høyde, material og/eller bredde i en familie. [18, s.319]

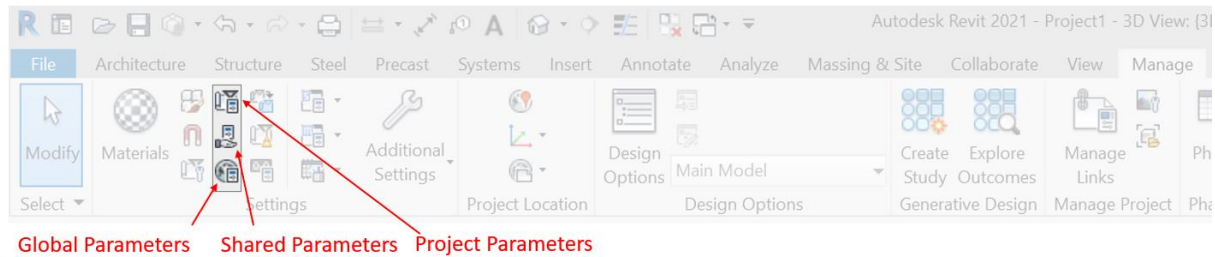
Logikken til en familie er definert i henhold til om den er System-, Loadable- eller In-Place-familie [18, s.15]. Kortfattet er disse definert som følger:

- En System-familie er forhåndsdefinert og lastes ned med programvaren. Dette er bygningsdeler som ofte utarbeides på byggeplass, slik som bjelker, søyler, tak og vegger. [18, s.15]
- En Loadable-familie kan importeres eller eksporteres. Denne familien utarbeides i Rvt før den lagres som en egen RFA-fil. Eksempler på denne type familie er bygningselementer som møbler og planter. [18, s.15]
- En In-Place-Familie er et unikt element, utformet i Rvt. Den utarbeides i henhold til geometri fra andre elementer. Dette innebærer å utforme en egendefinert familie, som ikke ellers er mulig å få frem i system-familier [23].

Typer er organiseringsnivå tre i hierarkiet, og som nevnt over kan en familie mates med én eller flere typer. Typer er en praktisk måte å uttrykke flere variabler samtidig, i én familie [23]. Det betyr at alle typer innenfor én familie, har like parametere [18, s.318]. Parameterne uttrykkes som type-parametere. Ved å tildele parameterne verdier, betyr at familiefunksjonen blir matet, og elementet blir skapt. Typeparametere endrer altså oppførsel eller andre aspekter av den spesifikke typen. Her er dimensjoner et eksempel på en type-parameter. [22]

Det siste nivået i hierarkiet er forekomst (instance på engelsk) [22]. En forekomst har ett sett med parametere som kun gjelder en spesifikk type familie. Den fremkommer ved å plassere en familie i arbeidsområdet til Rvt. Da tildeles elementet en unik ID, og defineres som et individuelt modellelement. [18, s.14, 318]

Elementene inneholder som nevnt parametere. Parametere kommer generelt som innebygde-, prosjekt-, delte-, og globale parametere. Innebygde parametere fremkommer i paletten med egenskaper, i henhold til valgt element. Mens prosjekt- og delte parametere kan tilpasses globalt, på utvalgt kategorier. De globale parameterne, kan benyttes på ulike forekomst- eller type- parametere i Rvt-kategorier. Hvor i Rvt-menyen, de tre sistnevnte parameterne kan produseres, er presentert i Figur 3.3. [22]



Figur 3.3: Tre parameter-typer i Rvt [22]

### 3.2 Finite Element Method

For å idealisere virkelige ingeniørproblemer i dag, benyttes FEM til å produsere en beregningsmodell [24, s. 25]. FEM er en forkortelse for Finite Element Method [24, s. 16], som på norsk betyr endelige elementmetode [25]. Rent matematisk er dette en metode som løser differensiallikninger numerisk. Disse ligningene anvender forskyvning som primærvariabel, som er årsaken til at metoden også er navngitt kraft- og forskyvningsmetoden. [24, s.16-17]

Publiseringen av den vitenskapelige publikasjonen «Stiffness and Deflection Analysis of Complex Structures,» av Aeronautical, i 1956 kan betraktes som starten til FEM [24, s. 12]. Etter denne publikasjonen, har FE-teknikken blitt behandlet gjennom titusener av vitenskapelige artikler og avhandlinger, samt omkring hundre publiserte lærebøker. Den praktiske fremveksten av FEM, har variert i henhold til to grunnleggende faktorer: teoriutvikling av FEM som teknikk, og den generelle datatekniske utviklingen. [24, s. 12-13]

Andre tilnærmede numeriske analysemetoder av FEM er FDM og BEM. Det er spesielt to egenskaper som gjør FEM til en bedre metode. For det første benytter FEM bundet symmetrisk systemmatriser. Disse er bedre egnet for implementering, fordi minnebehovet på datamaskinen blir moderat og raskt løselig. Den andre egenskapen, er at metoden behandler et bredere spekter av problemer. [24, s. 17] FEM-modellen utgjør en komplisert matematisk beskrivelse av strukturens egenskaper [24, s. 136]. Når modellen betrakter et konstruksjonsproblem, er et statisk lineært betraktning dominerende. Metoden behandler også ikke-lineære og dynamiske problemstillinger, men disse betraktningene er mer kompliserte å anvende. [24, s. 23]

FEM foretar FE-analyser av hele beregningsvolumet. Dette utføres ved at et antall elementer diskretiseres. Det betyr at elementene overføres til et løsbart problem, med et endelig antall variabler [26]. Utformingen av beregningsvolumet utføres i to deler: idealisering og modellering. Idealiseringen innebærer å skape en modell av en fysisk virkelighet. Modellen må forenkles, slik at det er praktisk mulig å opprettholde matematiske sammenhenger, for å ivareta koblinger. Her innebærer modelleringen å opprettholde disse koblingene. Det utføres ved å utvikle en matematisk beregningsmodell, som representerer det fysiske problemet. [24, s. 73]

Det skal alltid foretas forhåndsberegninger med tradisjonelle beregningsmetoder ved bruk av FEM. Disse verifiserer resultatene som fremkommer ved bruk av FEM. Resulterende beregninger presenterer et spenningsbilde, som benyttes til å vurdere konstruksjonselementenes bæreevne og

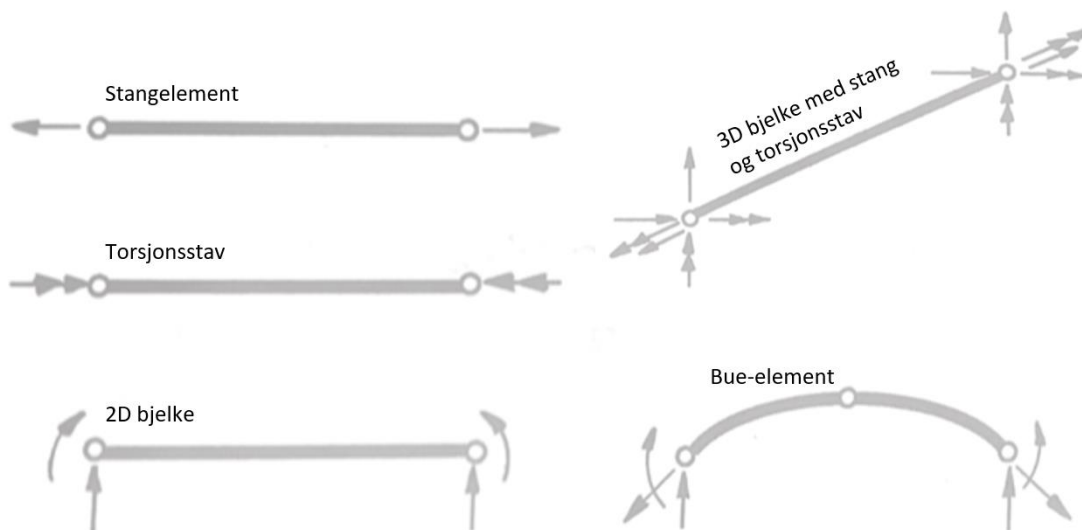
levetid. Det er derfor nødvendig at brukeren kombinerer dette med egne tekniske kunnskaper og erfaringer. [24, s. 25]

Bruken av FEM i dataprogrammer er i stadig utvikling. Dette kan bety at FEM-programmene som benyttes i dag, blir utdatert i fremtiden. [24, s. 14] Nå supplerer flere CAD-system for solid modellering, i form av FEM-moduler i konstruksjonsarbeid. Det betyr at programmer som bygger på CAD-system har innebygde FEM-løsninger. [24, s. 18,14,231] Filformatet som benyttes for dette kan være DWG-filer eksportert fra et CAD-program, eksempelvis AutoCAD [27] [24]. Å utnytte CAD-system i FEM program innebærer at arbeidet til ingeniørene rasjonaliseres [21, s. 231].

### 3.2.1 Linjeelement

For utarbeidelse av en matematisk beregningsmodell, må valg av elementtype defineres. Dette er én av faktorene som kan påvirker kvaliteten på resultatene. De fire mest vanlige endelige elementene i FEM er: linjeelement, skiveelement, plate- og skallelement og to og tredimensjonale solidelement. [24, s. 59-70]

Elementet med raskest beregningstid, er linjeelementer. Dette er et strukturelt element og idealiseres som endimensjonalt. Mange konstruksjoner kan tilnærmes som linjeelementer. Eksempler på geometri som kan representeres som linjeelementer er stang, torsjonsstav, 2D bjelke, kombinasjonen av bjelke og torsjonsstav, og bue. Disse er representert i figuren under.



Figur 3.4: Eksempler på linjeelementer i FEM, [24, bilde 3.1]

### 3.2.2 StruSoft FEM-Design

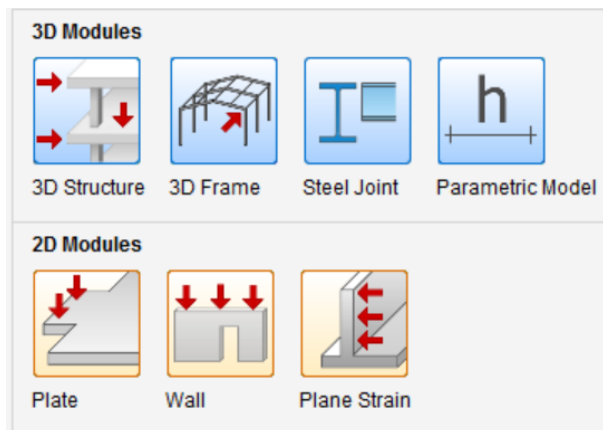
FEM-Design (Fd) er en programvare utviklet av StruSoft. Det er en programvare med et brukervennlig grensesnitt for strukturelle ingeniører. Fd gjennomfører eksakte analyser og konstruksjoner med bruk av Finite Element Method, som navnet på programvaren tilsier. Det er en programvare med full 64-bites, hvilket innebærer at alt minne på datamaskinen tas i bruk. Det har gjort det mulig å behandle store modeller, uten å få opp feilmeldinger når resultatene skal presenteres. [28, s.2 og 3]

Fd har et API som blant annet kan brukes av eksterne parametriske verktøy. API-et er basert på extensible markup language (XML), og muliggjør for kommunikasjon av Fd innhold via filformatet XML. XML brukes til å sortere og kommunisere data, og kan generere og/eller modifisere ulike programmeringsspråk. Eksempler på dette er Python, C#, VBA, C++. API-et til XML-filen er utarbeidet for to formål: [29]

1. StruXML, med filtypen .struxml. Denne brukes til å lagre objektdata som geometri, og masser av modellen. Dette brukes blant annet til å utveksle modeller med Rvt, og innebærer at filen representerer selve Fd-modellen.
2. FdScript, med filtypen .fdscript. Denne brukes til å utføre kommandoer som kjører analyser, i tillegg til å eksportere resultater. Dette innebærer at filen inneholder alle kommandoer for å kjøre Fd-modellen.

For å enklere produsere disse to filtypene og utføre kommandoer i Fd, tilbyr applikasjonen et sett med API-funksjonaliteter for Python, Rvt/Dynamo og Rhino/GH. [29]

Programvaren er systematisert i moduler, hvor tilgjengelige moduler avhenger av hvilken lisens brukeren har. For FEM-Design 20 er modulene presentert i Figur 3.5. Disse er utviklet for to- og tre-dimensjonale strukturelle problemer. En av modulene er «3D Structure», som vist i Figur 3.5. Denne modulen vil videre bli omtalt som Fd i denne rapporten. Modulen benytter kombinasjoner av plater, bjelker, søyler, avstivere med vilkårlig belastning. Her er .str det standardiserte filformatet som produseres. [28, s.3]



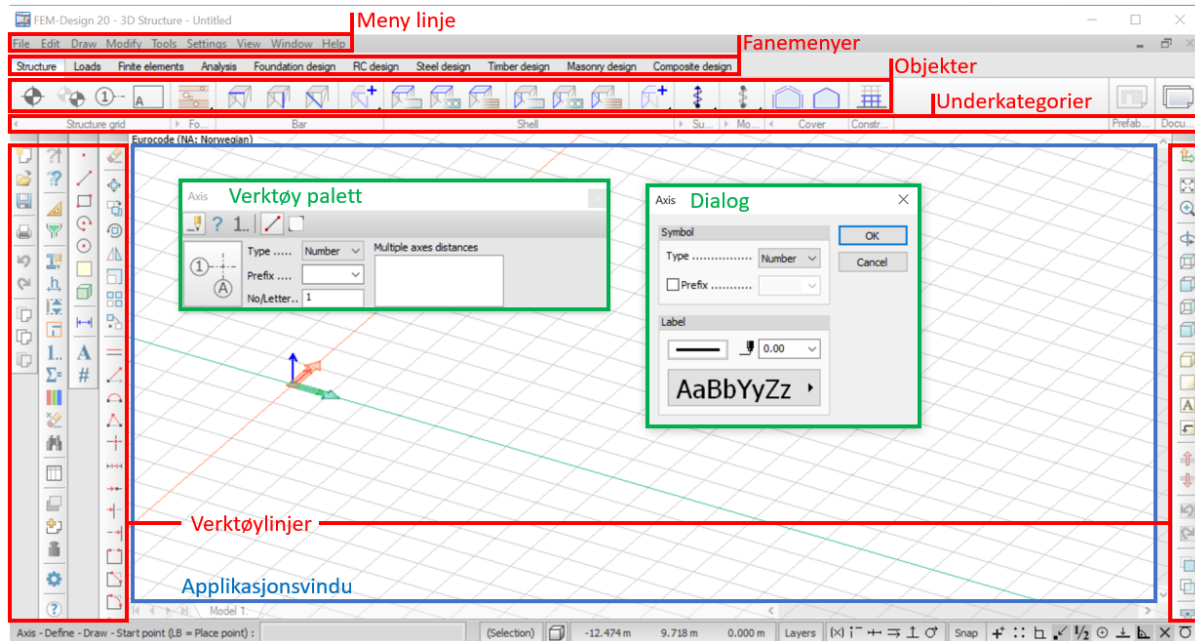
Figur 3.5: Modulene til Fd

Ved å velge en modul, åpnes den opp som et eget filformat. Her velges eurokode i forhold til hvilket land konstruksjonen skal beregnes for. Valget påvirker materialer og designparametere, og dermed resultatene. [28, s.4]

### Brukergrensesnitt

Brukergrensesnittet er likt for alle modulene, og innebærer oppsettet illustrert og presisert med merknader i Figur 3.6.





Figur 3.6: Brukergransnitt Fd – 3D Structure [28, s.5]

Fanemenyene inneholder hoved-funksjonalitetene i Fd. Rekkefølgen av plasseringen indikerer anbefalt rekkefølge på arbeidsflyten for strukturell prosjektering. Her deles arbeidsflyten inn i 10 fanemenyer. For fanemenyene «Structure», «Loads», «Analysis» og «RC design» er deres underkategorier presentert i Tabell 3.1.

Tabell 3.1: Fanemenyenes underkategorier i Fd

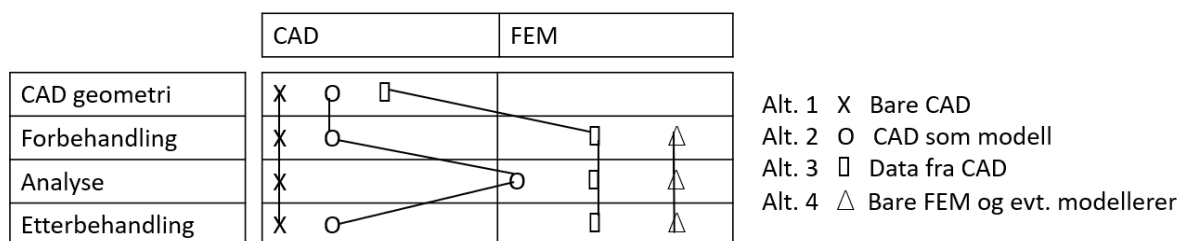
Fanemeny	Structure	Loads	Analysis	RC design
Underkategorier	Structure grid	Load case, combination, group & sunstruction stage	Calculate	Calculation
	Fondation	Force	Manage results	Manage results
	Bar	Temperature	Quick change	Quick change
	Shell	Stress	Equilibrium	Auxiliary results
	Suppport	Support motion	Auxiliary results	configure
	Modelling tool	Dynamic load	Deflection check	Bar reinforcement
	cover	Macro		Applied quantity
	Construction stages	Modify		
		Clipboard		

En modell i Fd består av strukturelle gjenstander, belastninger og endelige elementer. De strukturelle objektene i 3D elementer består av 1D stag og 2D plan, hvilket kommer av at Fd benytter FEM. Videre har disse objektene tverrsnitt med profiltipe og tykkelse, samt materialegenskaper. Det betyr at de kan defineres som linjer eller regioner.[30]

### 3.3 Samhandling av CAD og FEM

FEM modellen utformes oftest i forprosjektet, med utgangspunkt i tegningsunderlaget. I mange virksomheter der FEM-analyser utføres, utnyttes CAD-system for konstruksjonsarbeidet. Dette innebærer at FEM-arbeidet til ingeniørene rasjonaliseres, om CAD geometrien kan anvendes direkte for å skape en FEM-modell. For å behandle en FEM-modell i 3D, skal CAD-modellen også utformes i 3D. [24, s. 231]

Fire arbeidsmetoder som kan benyttes til å skape geometridata for FEM-modellen [24, s. 231], er presentert i Figur 3.7. I den venstre kolonne innebærer ordvalget CAD-geometri, tegningsgrunnlaget. Forbehandlingen innebærer å modellere opp geometrien, i henhold til riktige noder og elementer. Analysen innebærer å foreta beregninger, som analyseres. Etterbehandlingen innebærer at resultatene vurderes, før optimale løsninger i modellen presenteres.



Figur 3.7: Alternative arbeidsmetoder for å produsere en FEM-modell [24, s. 231]

Alt. 1 presenterer at CAD dataen ikke krever overføring, fordi det er mulig å arbeide i CAD-programmet for å utarbeide CAD-geometri, forbehandling, analyse og etterbehandling. Dette er den mest effektive og enkleste måten for ingeniører å arbeide på. Problematikken ved å benytte dette alternativet, er at CAD-systemene er mindre utviklet enn de mer robuste FEM-systemene. Fordelen med dette alternativet, er at eventuelle modifikasjoner av geometrien foretas direkte i programmet. Dette eliminerer risikoen for avvik mellom CAD- og FEM-geometri. [24, s. 231]

For Alt. 2 blir FEM-modellens geometri og forbehandling utformet som et CAD-system. Deretter utveksles noder og elementer til FEM-programmet for gjennomføring av analysen. Produktet av analysen returneres tilbake til CAD-systemet for å presentere resultatene. [24, s. 232]

I Alt. 3 overføres CAD-geometrien til et FEM-program. Her blir forbehandling, analyse og etterbehandling foretatt direkte i FEM-programmet. Denne prosedyren er fordelaktig med tanke på at FEM-system normalt består av kraftigere funksjoner til både for- og etterbehandling av løsninger, sammenliknet med FEM-modulene i CAD-systemene. [24, s. 232]

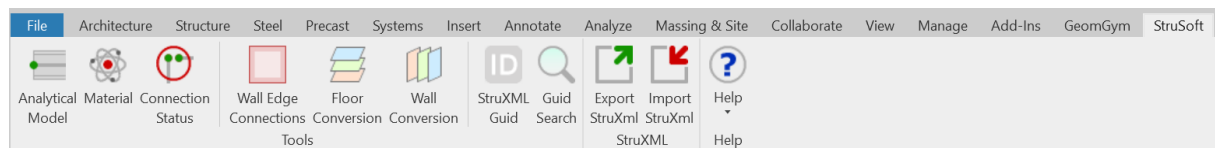
For det siste alternativet, Alt. 4, startes FEM-modelleringen uten CAD-geometri. Her blir FEM-modellen utarbeidet med noder og elementer, som benyttes til analysen og etterbehandlingen, direkte i FEM-programmet. Herav ingen utveksling mellom CAD- og FEM-programmer.

#### 3.3.1 StruXML Revit Add-In

«StruXML Revit Add-In» er verktøy som kan benyttes til gjennomføring av utveksling, mellom CAD- og FEM-modeller. Dette er et verktøy som er utviklet av StruSoft, og muliggjør direkte

kommunikasjon mellom Rvt Structure og Fd. Dette er gratis å laste ned. For å kunne gjennomføre overføringen av modeller mellom programmene, må modellen i Rvt bestå av analytiske elementer. Da kan modellen utveksles mellom de to programmene, og videreføre egenskaper tilknyttet elementene. [31]

Det eneste kravet for å kunne ta i bruk «StruXML Revit-Add-In» er at riktig versjon av StruXML er lastet ned til tilsvarende versjon av Rvt som skal benyttes. Etter nedlastningen får Rvt en ekstra fane med navnet StruSoft. Denne inneholder verktøy inndelt i tre grupper. Dette er gruppene Tools, StruXML og Help. Disse, samt verktøyene er illustrert i Figur 3.8. [32]



Figur 3.8: Verktøyene i fanen StruSoft i Rvt

For å åpne en StruXML-fil i Rvt, som er produsert i Fd, kan dette skje ved å benytte verktøyet «Import StruXML» i Figur 3.8. For import, må filen kartlegges for å kontrollere at egenskapene definert i Fd, får tilsvarende egenskaper i Rvt. Dette er en manuell arbeidsprosess for hver import som gjennomføres. [33] For å åpne struXML-fil i Fd, gjøres dette i eget program slik som for Rvt. Her er prosedyren for import også en manuell arbeidsoppgave. [34]

Armering for søyler og bjelker inngår som et analytisk element. Disse følger derfor med i overføringen av modellene mellom Rvt og Fd. Her tilsvarer navnet på armerings-stengene i Rvt; armerings-diameterne og kvaliteten i Fd. Disse er modellert som bøyler og langsgående stenger i Fd.[35]

### 3.4 Parametrisk design /modellering

Parametrisk design i et prosjekt, innebærer blant annet å spesifisere hovedparameterne som igjen resulterer i interaktive(samspill) endringer. En parametrisk modell oppdateres dermed automatisk. Dette er et hjelpemiddel som blant annet brukes av arkitekter og ingeniører. [36] For arkitekter hevdes det at det er krevende å gå fra kreativiteten og inn i programmeringen av konstruksjonene. Mens for ingeniørene er det mindre krevende, da de har mindre parametere å forholde seg til, sammenliknet med arkitektene. [37]

Parametrisk design utføres med egne verktøy. Disse er utformet for å få en mer effektiv prosjektering hos ingeniørene, og å oppnå et mer utfordrende utseende hos arkitektene. Hos begge aktørene åpnes det for å utforske flere alternativer, og eventuelt optimalisere konstruksjoner. Her er målet å finne de riktige inngangsparameterne, for så å få oversikt over den geometriske logikken. [36] Dette er nødvendig for å legge til regelbaserte forhold og logikk, istedenfor å gjøre manuell utforming. Et slikt arbeid, blir tilgjengelig for flere brukere, og ikke bare på de store kontorene. [38]

Forskjellen på parametrisk BIM og parametrisk design, er i hovedsak at BIM setter ut parametriske objekter på en manuell måte. Når det nå snakkes om parametrisk design, innebærer det noe mer enn en modellering (slik som det er praktisert de siste 12-13 årene). [39] BIM har manglet muligheten til å

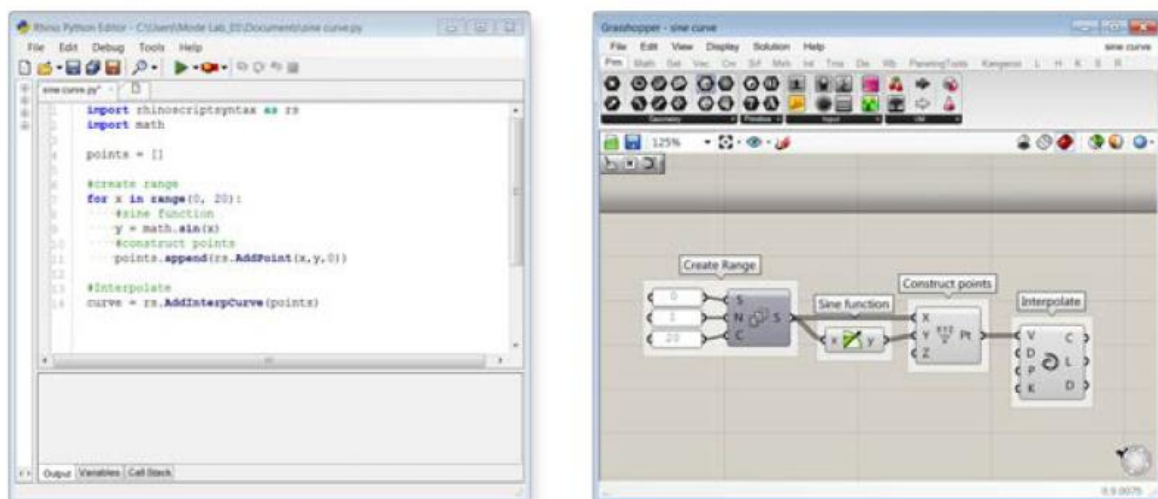
gjøre “smart modellering” i form av å beskrive algoritmene, og dermed programmere. Dette er mulig med parametrisk design. [39]

### 3.4.1 Visuell programmering

En måte å gjennomføre parametrisk design, er ved bruk av visuell programmering. Dette er et paradigme innenfor dataprogrammering. Her manipuleres logiske elementer grafisk av brukeren selv, fremfor å gjennomføre dette skriftlig slik som C#, Python, Rhinoscript. Programmer som krever slike koder, er bundet av syntaks-språk. Dette er ikke nødvendig i visuell programmering. Visuell programmering muliggjør å koble sammen funksjonelle blokker/komponenter til en sekvens av handlinger. Ulikheten mellom tekstlig koding og sekvenser av komponenter, er illustrert i Figur 3.9. Her er produksjonen av å tegne en sinuskurve i programvaren GH, utført både med Python-skript og med komponenter. [1, s.6]

Begrensningen ved å benytte komponenter, er at de må motta data fra riktig type utgang, fra oppstrøms-komponenter. En slik «skripting» gjør det mulig å unnsnippe barrieren av å ta i bruk tekstprogrammeringsprogram, som mange arkitekter og ingeniører ikke ønsker å bruke.

Grensesnittet for å benytte visuell programmering er derav mindre ukjent. [1, s.6]



Figur 3.9: Prosessen av å tegne sinuskurve i Python og med komponenter i GH [1, s.6]

To utbredte visuelle parametriske programvarer er GH og Dynamo. Disse er utviklet i henhold til hvert sitt CAD-program: Dynamo for Rvt og GH for Rhinoceros (Rhino). Rvt er her i større grad standarden for leveranser av både tegningsark og modeller, og de fleste ingeniørkontorer har tilgang på Rvt. Her er det verdt å nevne at prisforskjellen på lisensen til Rvt og Rhino er relativt stor. Rvt lisensen er dyr, sammenliknet med Rhino. [40]

En av filmene på Youtube-kanalen «Structured Parametrics», sammenlikner nettopp disse to programvarene. Kanalen har 1,84 tusen abonnenter, og relaterer arbeidet sitt til strukturelle ingeniører, BIM, og arkitektur [41]. Det fremkommer her at GH som programvare, har flere positive aspekter, sammenliknet med Dynamo. Det kommer blant annet av at GH har eksistert i lengre tid, og har derav flere samfunnsstøtter. Dette betyr at det er stor sannsynlighet for at spørsmål brukeren av programvaren har, kan besvares. I tillegg er det mer brukervennlig, og er lettere å ta i bruk i begynnelsen. GH kjører skriptene raskere, og har flere gjenstander i hukommelsen, i tillegg til at det

er stabilt. Det har også flere Plug-Ins som er utviklet av felleskapet. Eksempler på Plug-Ins er: spesiell geometri, strukturelle analyser og database-redskaper. [40]

Dynamo er en nyere programvare, og er vanskeligere å håndtere. Det kommer av at det mer teknisk, sammenliknet med GH. I tillegg kjøres programmeringen saktere og kan frysas, og kan kreve omstart. Det som gjelder Plug-Ins hos Dynamo, er vanskeligere å få tak i, og er mindre utviklet. Men hvordan GH og Dynamo fortsetter å utvikle seg i fremtiden, kan endre dette standpunktet. [40]

### 3.4.2 Grasshopper3D

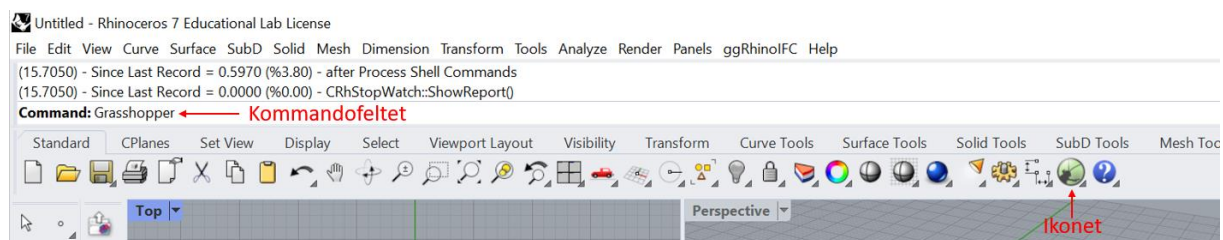
GH er et grafisk algoritmeverktøy, utviklet av David Ruttern. Dette verktøyet er innebygd i CAD-programmet Rhino[1, s.8], som en Plug-In. Dette innebærer at GH i seg selv er gratis [40]. Det produseres ingen geometriske objekter i GH-vinduet, men representerer instruksjoner og regler, i form av komponenter. Disse gir instruksjoner på hvordan geometrien skal modelleres i visningsområdet til Rhino. [1, s.28] Endringer som gjennomføres i GH oppdaterer utfallet fortløpende[42], da det har et dynamisk miljø til Rhino[1, s.29].

At programmet tar i bruk grafiske algoritmer som modelleringsverktøy, betyr at brukerne ikke trenger å ha kunnskap om skripping eller tekstprogrammering. Algoritmer er en Step-By-Step prosedyre, og produserer sekvenser. I tillegg gir det mulighet til å definere presis parametriske kontroll til en modell, muligheten til å undersøke generative arbeidsflyt, samtidig som det er en plattform som utvikler et høyt nivå av programmeringslogikk. Dette programverktøyet benyttes av flere aktører, eksempelvis hos arkitekter, ingeniører og produktutviklere. [1, s.2,6,8]

GH ble opprinnelig opprettet som en funksjonalitet innad i Rhino 4, i 2008. Her var den integrert i knappen «Record History», som muliggjorde å ha kontroll på en eksplisitt historikk. Dette omtales som forløperen av GH, for i 2014 hadde programvaren utviklet seg til å bli et kraftig visuelt programmeringsverktøy. [1, s.6] Siden Rhino 4, har GH blitt videreutviklet sammen med Rhino 5, 6 og 7. [43]

### Brukergrensesnitt

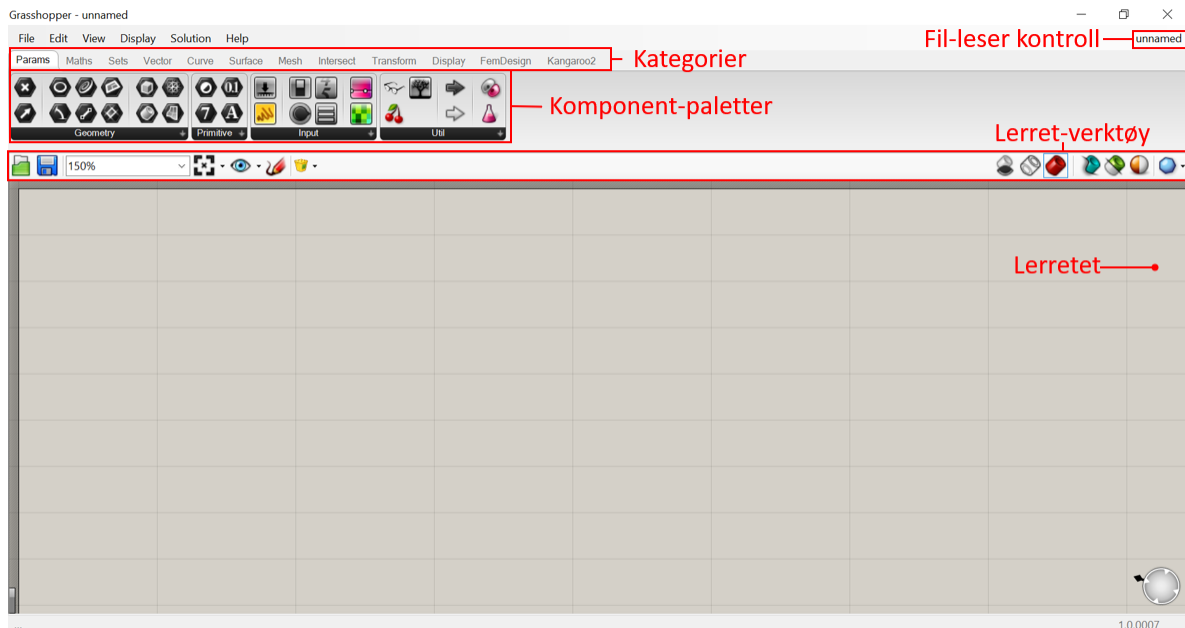
GH kan tas i bruk i Rhino 7 på to ulike måter. Disse er presentert i Figur 3.10. Enten skrives det inn Grasshopper i kommandolinjer til Rhino, eller ved å klikke på GH-ikonet. Ved å gjennomføre dette, åpnes et eget vindu, slik som vist i Figur 3.11.



Figur 3.10: Få frem Grasshoppervinduet fra Rhino

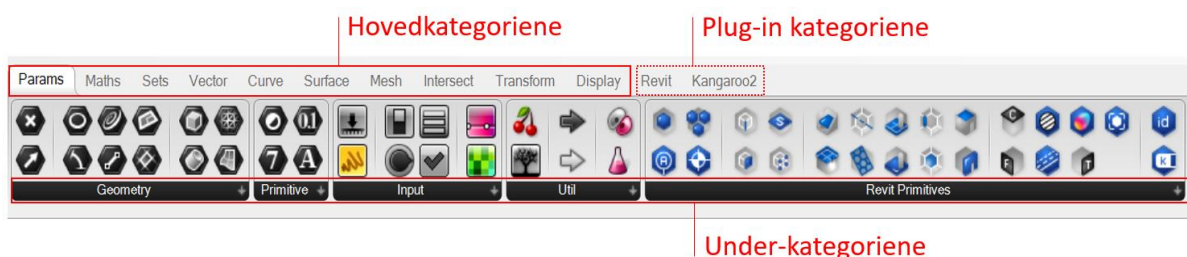
Grensesnittet til GH inneholder elementer med funksjonene illustrert i Figur 3.11. Her er fem av funksjoner utpekt. Den første funksjonen er lerretet, og er det primære arbeidsområdet for å produsere data med komponenter og vaiere. [1, s.19-20] Den andre funksjonen er fil-leser kontroll.

Den tillater å raskt arbeide i ulike filer som er lastet inn. Det gjøres ved å velge dem gjennom en nedtrekksmeny. Her kan komponenter og innhold enkelt kopieres og limes på tvers av GH-fil. [1, s.17]



Figur 3.11: Brukergrensesnittet til GH [44], laget i henhold til [1, s.15]

Den tredje funksjonen, er kategorier. Hver kategori inneholder komponent-paletter. Disse komponentpalettene inneholde komponenter. I Figur 3.12 er komponentpalettene igjen organisert i hver sin underkategori, i form av nedtrekksmenyer. [1, s.18] Kategoriene innhold dermed alle standardiserte algoritmene som er ferdigutviklet [45]. I figuren er kategoriene delt inn i to grupper, Plug-In kategoriene og hovedkategoriene. Hovedkategoriene er de som følger med GH, mens Plug-In kategoriene er nedlastede tilleggspakker.



Figur 3.12: Kategoriseringen av komponent-paletter i GH

For å opprette innhold i GH-vinduet, plasseres det som sagt komponenter på lerretet. Komponenter som kobles sammen med variere [1, s.7], etablerer en datastrøm fra venstre til høyre. Ved endt utarbeidelse av komponenter, lagres GH-filen med det standardiserte filformatet .gh. [1, s.56]

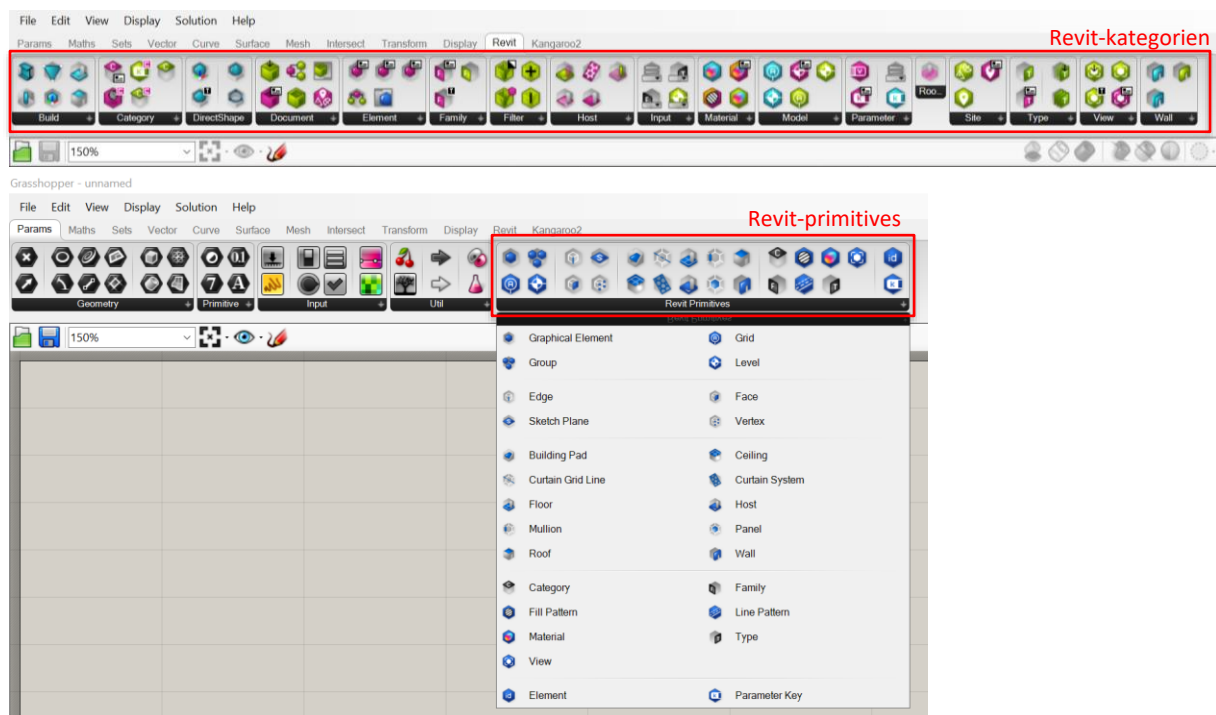
### Rhino.Inside.Revit

For å oppfylle behovet for AEC industrien, ble plattformen Rhino.Inside.Revit (RiR) etablert. Med RiR kan Rhino 7 og GH lastes inn i Rvt API-et [46]. Det er en relativt ny plattform, produsert av McNeel & Associates. [47]. I henhold til [48] ble RiR som nedlastingsvare tilgjengelig den 6. november i 2019.

Dette prosjektet har potensiale til å forbedre design og dokumentasjon i Rvt. Den viktigste funksjonen i et visuelt programmeringsmiljø, slik som GH, er interaktiviteten til komponentene. RiR viderefører dette over til Rvt, slik at brukeren har mulighet for å utforske designet mer effektivt, samt utforme nye løsninger. [2]

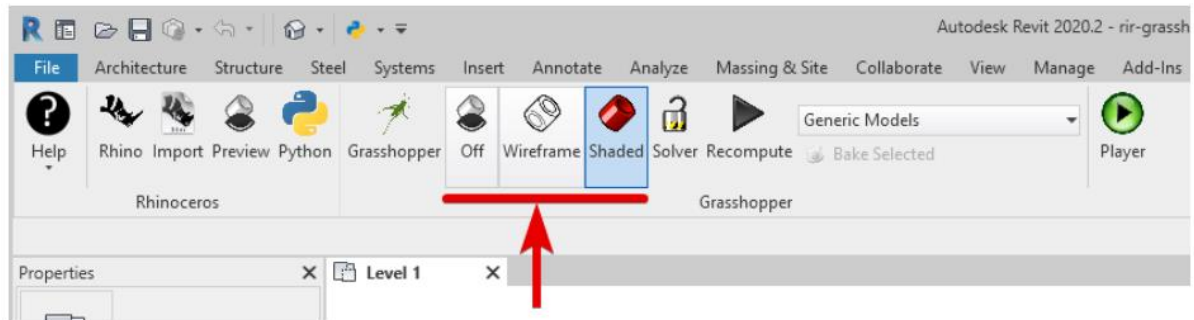
Siden 2019 har det kommet flere oppdateringen av verktøyet. Oppdateringene av RiR publiseres fortløpende, og gjøres tilgjengelig på nettsiden deres. I henhold til [49] er det publisert 26 oppdateringer, på tidsperioden fra 13.01.2020 til 04.03.2021. Den siste lanseringen av programvaren per 24.03.2021, er Rhino.Inside.Revit v0.1. Når nye versjoner av programvaren er tilgjengelig, kan disse lastes ned fra nettsiden [50].

Ved å ta i bruk RiR, får GH en ekstra kategori (navngitt Revit), samt Rvt primitives i eksisterende kategori «Params». Noen av komponentene som disse to kategoriene inneholder, er presentert i Figur 3.13. [2]

















Figur 3.13: Ekstra komponenter i GH, ved nedlastning av RiR [44]

Som vist i Figur 3.14, får også Rvt en ekstra fanemeny for bruk av Rhino og GH. Denne fanen er navngitt Rhino.Inside. Her kan både Rhino og GH åpnes via Rvt. Fremvisning av Rhino-geometri kan også bestemmes her, de tre alternativene er vist i Figur 3.14. Figuren illustrerer også «Solver» knappen, til høyre for «Shaded». Denne funksjonen er nyttig for å redusere ventetiden på større Rvt-modeller.[51]



Figur 3.14: Forhåndsvisning av GH-data i Rvt [51]

GH-komponenter som er Rvt-beviste, finnes i en rekke farger og merker. Disse er benyttet på Rvt-ikonene, for å indikere type, identitet og andre aspekter som dataen i tilsvarende komponent er utformet å fungere med. Dette er eksempelvis presentert i Figur 3.15. [2, 51] hvor det skilles mellom 2D/Data og 3D.

2D/DATA	3D		
		PARAMETERS	Parametere
		QUERY	Spørsmål
		DECONSTRUCT / ANALYZE	Dekonstruere/analyse
		MODIFY	Modifisere
		CONSTRUCT / CREATE	Konstruere/skape
		DELETE / PURGE	Slette/rens
		TYPE PARAM / COMPONENT	Type parameter/komponent

Figur 3.15: Farge og merknad i Grasshopper på Revit-ikoner [2]

### FEM-Design verktøykasse

Visuell programmering kan også benyttes for å produsere beregningsprodukter. I kapittel 3.2.2 ble det nevnt at Fd tilbyr et sett med API funksjonaliteter for GH. Disse funksjonalitetene er utarbeidet som en verktøykasse, som lastes ned som en Plug-In i GH. [52] Verktøykassen lastes ned fra nettsiden food4rhino.com[52, 53], som er en åpen ressurs. Som åpen ressurs, betyr dette blant annet at kodene i komponentene er tilgjengelige på github, i [54]. Denne verktøykassen gir GH tilgang på API-et til Fd. Verktøykassen får også her, en egen kategori med verktøy-paletter i GH, slik som ved nedlastningen av RiR.

Kodene som produserer data i komponentene her, genererer en dataflyt fra GH til Fd grensesnittet. [54] Her skjer kommunikasjonen mellom GH og Fd, via StruXML. Det skal da være mulig å gjennomføre utforming av parametrisert modell, kjøre analyse og lage automatisert arbeidsflyt i Fd. [52]



## 4 Forskerspørsmål

Formålet i denne masteroppgaven er å ta i bruk ett parametrisk verktøy, for produksjon av brukonstruksjoner med krumninger i flere plan. Dette skal gjennomføres i samsvar med DS sine tradisjonelle programvarer. Hensikten er å redusere tidsbruken som benyttes ved revidering i prosjekteringsprosessen til en RIB konsulent, samt utvide modelleringsmulighetene til tegningsmaterieell og beregningsgrunnlaget. Derfor er forskerspørsmålet som følger:

### **Hvordan kan parametrisk modellering generere en effektivisert revisjonsprosess, av geometrisk- og statisk modell, for prosjektering av brudekker?**

1. Hvordan kan det utvalgte parametriske verktøyet, forbedre samhandlingen mellom CAD- og FEM-modellutveksling?
2. Hvilke prosjekterings-elementer, i geometrisk og statisk modell, kan produseres med valgt parametrisk verktøy?
3. Hvilke revisjonsprosesser, i geometrisk- og statisk modell, kan utføres effektivt med valgt parametrisk verktøy?

#### 4.1 Avgrensninger

På grunn av tidsrammen på gjennomføringen av denne oppgaven, er det nødvendig å begrense oppgaveomfanget med følgende avgrensninger:

- For prosjektering av brudekker, er disse avgrenset til å hovedsakelig fokusere på bjelke-element i armert plastøst betong. Dette elementet kan også defineres som overbygningen til en bru.
- Prosjekteringen skal tilfredsstillere tradisjonelle modellerings- og beregningskrav for tegnings- og beregningsprodukter, som utarbeides hos Dagfin Skaar AS. Beregningsmodellen skal være statisk, hvilket betyr at dynamiske effekter idealiseres som statiske effekter. Det kommer av at dynamiske analyser ikke gjennomføres hos Dagfin Skaar AS.
- Det er produktene i programvarene og revisjonsmulighetene som er i hovedfokus, og det skal derfor ikke gjennomføres en ideell optimalisering og verifisering av beregningsmodellen. Det innebærer at vurdering av beregningsresultater ikke vil bli gjennomført.
- Det som gjelder revisjonene, er tiltenkt å skulle gjelde i alle fire fasene av en prosjekteringsprosess, altså et prosjekt på en utførelsesentreprise [3].
- I valgt parametrisk verktøy avgrenses oppgaven til å ha hovedfokus på å ta i bruk ferdigkomponerte komponenter. Dette kommer av at grensesnittet til tekstlig programmering er et helt annet fagfelt for en for bygg-ingeniører.
- Kostnadsperspektiv og CO<sub>2</sub>, er faktorer som ikke vil bli inkludert.

## 5 Case

Casen i dette prosjektet baserer seg på bruk av ett FEM- og ett CAD-program, samt ett parametrisk verktøy. Disse skal benyttes til prosjektering av reviderbare brudekker, i henhold til prosjekteringsprosessen hos bedriften Dagfin Skaar AS. Produktene produsert i de utvalgte programvarene, er tiltenkt å skulle brukes som en mal til liknende bruprosjekter. Programvarene er allerede nevnt i teorikapittelet, og er Rvt, Fd og GH. I dette kapittelet vil metoder og elementer til programvarene utdypes, i henhold til prosjekteringsrelevans i dette prosjektet. Casen innebærer derfor følgende tema:

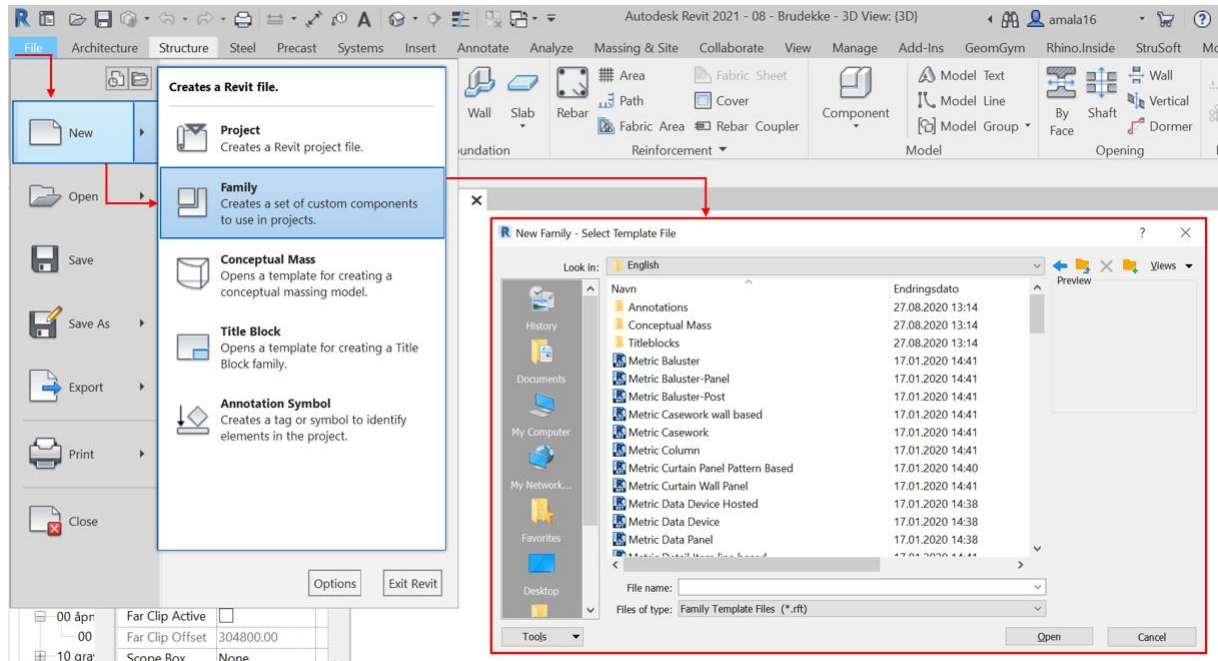
1. Programvarene
2. DS prosjektering
3. Brudekket

### 5.1 Programvarene

For utveksling av data mellom GH og Rvt, samt GH og Fd, er bindeleddet for dette, programvaren Rhino. Den siste versjonen av Rhino, er versjon 7, hvor Rhino selv hevder at dette er den viktigste oppgraderingen de har hatt. Programvaren ble lansert i desember 2020, og gjør det mulig å oppnå en helt ny arbeidsflyt innenfor modellering. De hevder videre at dette er den mest stabile og raskeste versjonen noensinne. En av de viktigste forandringene relatert til dette prosjektet, er at både Rhino og GH kan kjøres i Rvt, som et Rvt tillegg sammen med RiR. [55]

#### 5.1.1 Revit

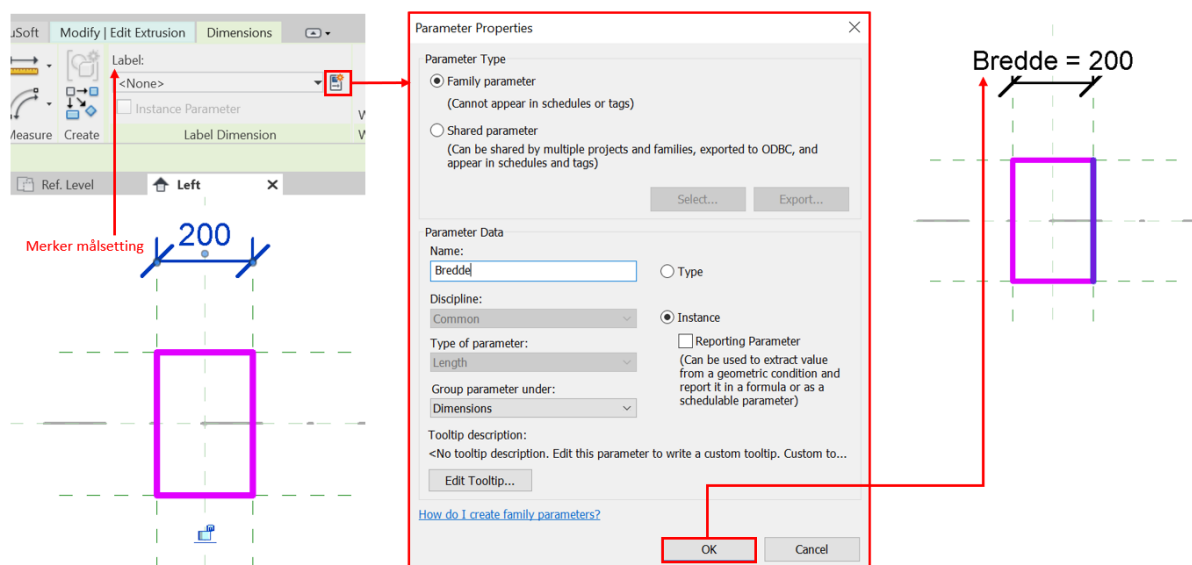
Egendefinerte familier er et av de viktigste automatiseringselementene i Rvt. Disse er smarte og fleksible, og kan tilpasse seg flere modell-forhold. Rvt tilbyr et standard sett av egendefinerte familie-maler, basert på målesystemer i henhold til typene «imperial» og «metric». [22] Ved nedlastning av Rvt medfølger et utvalg familie-maler. For opprettelse av modell-elementer, benyttes familietypen «Metric Templates», i dialogen for å lage en ny familie. [18, s.321] Fremgangsmetoden for opprettelse av ny familie og utvalg av familie-maler, er illustrert i Figur 5.1.



Figur 5.1: Familie-maler

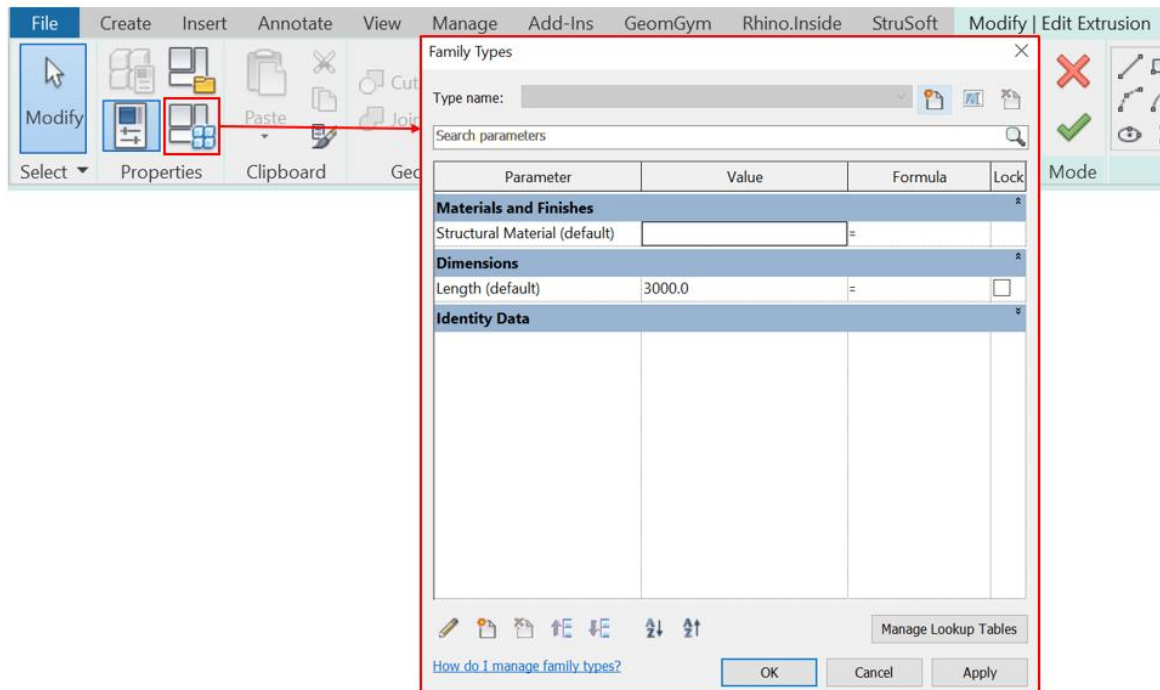
Det er nettopp disse malene som benyttes ved opprettelse av en ny familie. Her det det nødvendig å starte i riktig mal, for at Rvt skal kunne tildele familien riktig kategori. Det kommer av at kategoriene verken kan egendefineres, eller endres på. På den måten plasseres familien på riktig plass i hierarkiet, og får en lik avbildning som de andre familiene, i samme kategori. [18, s.321]

I mal-filene defineres enten origo av brukeren, eller så er origo forhåndsdefinert. Videre skal referanseplan og referanselinjer utarbeides, som støtte for geometriutførelsen. Disse brukes for å styre høyde, lengde og bredde på geometrien. For parametrisk målsetting, gjennomføres dette i henhold til referanselinjene. For at målene skal bli parametere, legges det til «Labels» på målsettingen. Hvordan dette gjennomføres i en familie-mal, er illustrert i Figur 5.2.



Figur 5.2: Sette «Label» på målsetting i en familie

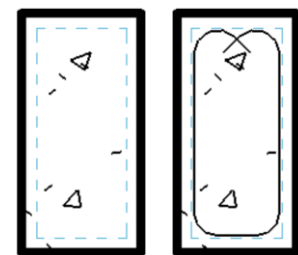
I Figur 5.2 fremkommer det at målsettingen må merkes for å få opp fanemeny «Label Dimentions». Målsettingen tildeles «Labels» (merknad), enten i nedtrekksmenyen eller ved opprettelse av en ny. En ny merknad, produseres i dialogen «Parameter Properties». Her defineres navnet på parameteren, samt om det er forekomst eller type. For å kontrollere parameterne, etablert med «Label», brukes knappen «Family Type» i fanen «Modify|Edit Extrusion», se Figur 5.3. [18, s.332-334]



Figur 5.3: Kontroll av «Label» opprettede parametere, til egendefinert familie i Rvt

## Armering

Armeringen er en fleksibel geometri i Rvt, som forholder seg til en definert armeringsform. Den utformes i henhold til geometrien i betongelementet, i tillegg til plasseringen av andre armeringsjern i samme elementet. Endres dimensjonene eller formen på elementet, endres også armeringen i henhold til overflatene, som elementet består av. Armeringen forholder seg også til en overdekningsparameter og diameterstørrelse. Hvordan armeringen i et snitt kan se ut, er vist i Figur 5.4. [56]

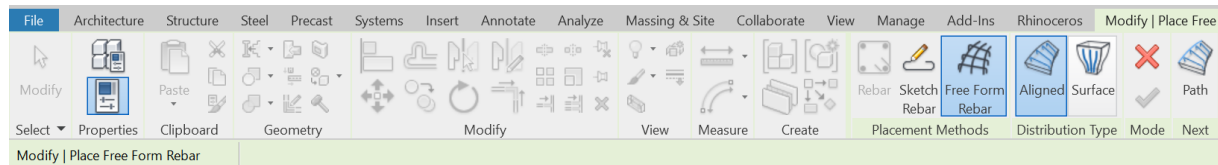


Figur 5.4: Armert bjelke, snittvisning

Armeringstenger i et betongelement, kan enten utarbeides som «Rebar», «Sketch Rebar» eller «Free form» [57]. Ved bruk av «Rebar», plasseres individuelle forekomster av armering i snittvisninger [58]. Mens ved «Sketch Rebar», gjennomføres modelleringen i 2D- og 3D visninger. Denne metoden skisserer opp en friformet armering, i henhold til arbeidsplan. [59] Den siste måten å armere på er med «Free Form» [57].

Metoden for å produsere «Free form» armering, gir mulighet for å modellere tverrarmring (Aligned) og lengdearmring (surface). Disse to typene produseres i henhold til overflater og kurver, til et irregulært betongelement. Tverrarmringen utformes ved å først definere overflatene, som armeringen skal legges i henhold til. Deretter velges en linje/kurve som definerer retningen den skal

legges i. For lengearmeringen derimot, defineres først overflatene som armeringen skal legges i henhold til, før start og sluttoverflatene defineres. Begge modellerer armering i henhold til bestemt overdekning. Hvor disse funksjonene er plassert i Rvt-menyen, er presentert i Figur 5.5. [60]



Figur 5.5: "Free Form" armering i Rvt

Det er flere generelle parametere som definerer egenskapene til armering, modellert i et betongelement. Det er blant annet type armering (standard, bøylar), type endeforankring og regel for utplasseringen av stengene (enkle stenger, fast nummer, maksimal avstand, minste avstand, og antall og avstand). Valgmulighetene på type utplassering, er et parametrisk element i denne armeringsmetoden.

### 5.1.2 FEM-Design-3D structure

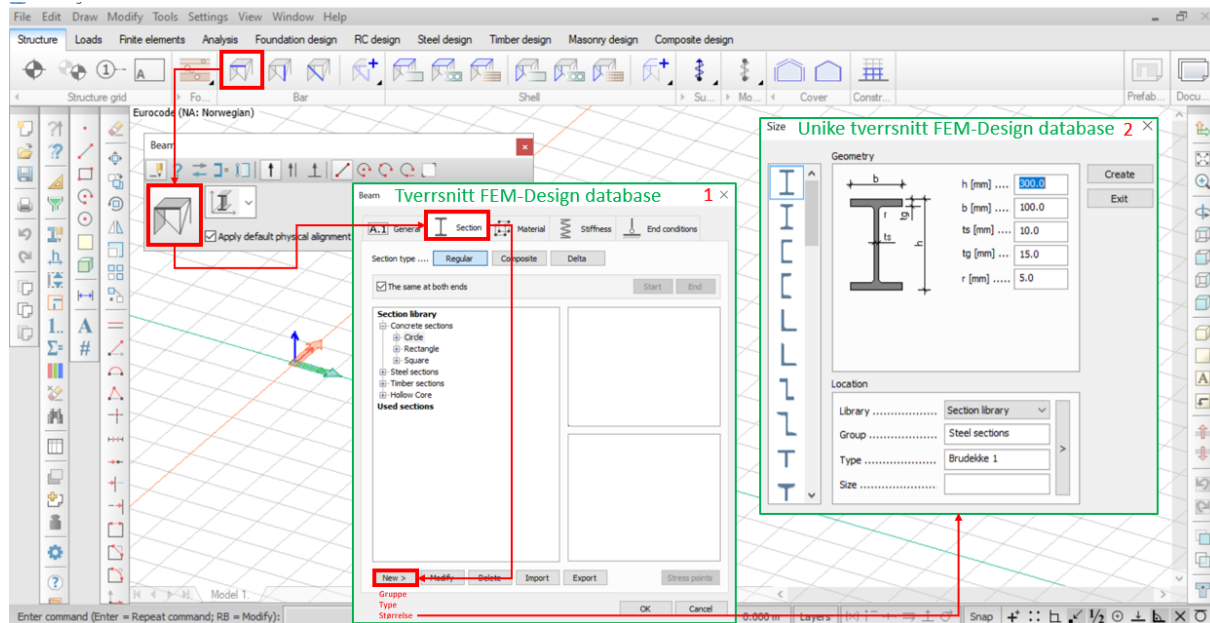
Som beskrevet i teorikapittelet, er fanemenyene til Fd plassert i en rekkefølge som indikerer den anbefalte arbeidsflyten. De relevante elementene i Fd, i henhold til casen av brudekket som presenteres i kapittel 5.3 senere, blir videre presentert i samme rekkefølge som anbefalt arbeidsflyt.

Første del av arbeidsflyten, er definisjonen objektstyper. Her er to av objektstypene i fanemenyen «Structure», presentert i Tabell 5.1. Dette er objektstypene bjelke, og punkt- og linjeopplegg gruppe. Hvordan disse blir definert, og hvilken posisjon de kan ha i rommet, fremkommer også i denne tabellen.[30]

Tabell 5.1: To objektstyper i Fd, med definisjon og posisjon

Objektstype	Definisjons modus	Posisjon
Bjelke	Linje	Valgfri
Punktopplegg gruppe	Punkt	Valgfri

For objektstypen bjelke, produseres denne ved at den tildeles en tverrsnittprofil. Type tverrsnittprofil kan velges fra et standardisert bibliotek, eller fra et bibliotek for produksjon av egendefinerte tverrsnittprofiler. Den førstnevnte muligheten for valg av tverrsnittprofil, er presentert som punkt 1 i Figur 5.6. Mens hvordan egendefinerte tverrsnitt kan utarbeides, er presentert punkt 2 i figuren. I punkt 2, vises det til at det er mulig å lage et tilnærmet unikt tverrsnitt, hvor tilgjengelige parametriske former kan velges, og defineres med unike dimensjoner. Disse kan defineres for materialtypene stål, betong og treverk.[30]



Figur 5.6: Tverrsnitt som strukturelle objekter i Fd








Betongelementer produseres også i henhold til materialegenskaper. Disse avhenger av valgt eurokode. Databasen til Fd har et eget materialbibliotek. Den er tilgjengelige i dialogen som er presentert som punkt 1 i Figur 5.6, ved siden av «Section» knappen. Her er materialene inndelt i grupper og typer. De tilgjengelige standardiserte materialgruppene er betong, stål og tre, samt generelle materialer. Det er på lik linje med tverrsnittsprofiler mulig å definere et nytt materiale, hvor materialegenskapene egendefineres. [30]






### Laster

Etter at den strukturelle modellen er utarbeidet som et statisk system, er neste steg i arbeidsflyten, å definere belastningene, lasttilfellene og lastkombinasjonene. Objekter for disse formålene, er plassert under fanemenyen «Loads». [61]

De fysiske lastene, kan defineres med krysningspunkt, kraftlinje eller kraftregion. Hver lasttype har en egen «verktøy palett», hvor egenskaper bestemmes. Disse bestemmes i henhold til lastens belastningstype, verdi/intensitet, utplassering og retning. [61] For modulen 3D-Structure er det mulig å modellere ni ulike belastningstyper. I Tabell 5.2 er fire av lasttypene presentert, med tilhørende belastningstyper, retning og utplasseringsverktøy. [61]

Tabell 5.2: Noen av lasttypene i Fd 3D-Structure [61]

Type last	Belastningstype	Retning belastning	Utplassering
Punkt	 Kraft/moment retning	 Vinkelrett på plan	 Punkt Velg punkt
Linje		 Vinkelrett på plan  Objektets lokale plan	

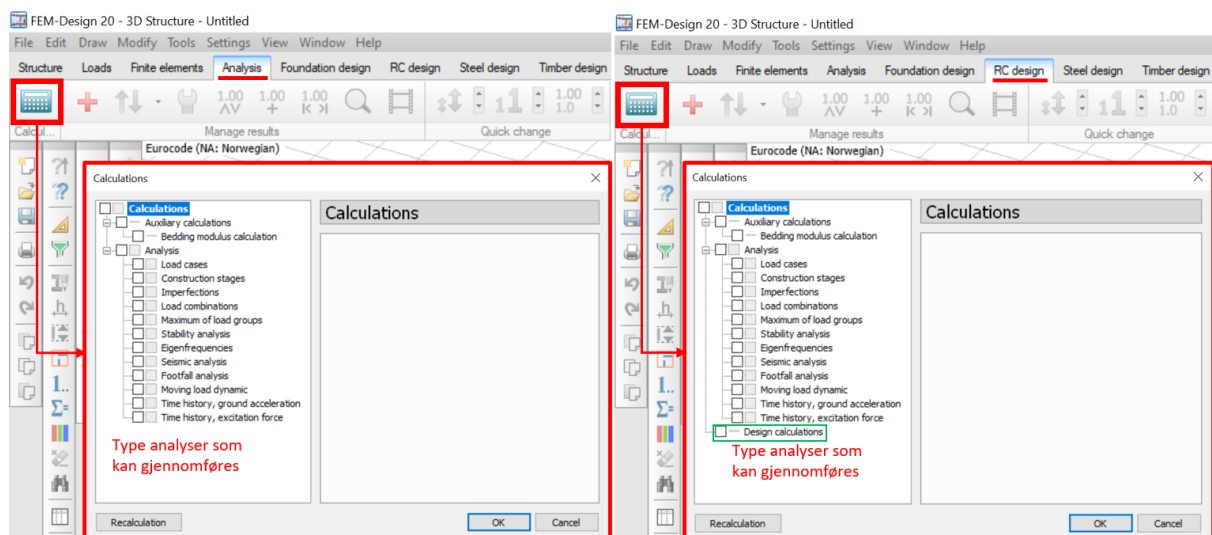
	Kraft/moment retning		Rett linje definert med to punkt Bue definert med senter, start- og endepunkt Bue definert med tre punkt Bue definert med start-, ende-punkt og tangent Velg linje
Temperatur-variasjon	Retning	<ul style="list-style-type: none"> <li> Forhåndsbestemt retning</li> <li> Parallell med linje</li> <li> Vinkelrett på planet</li> <li> Objektets lokale plan</li> </ul>	 Rett linje definert med to punkt Bue definert med senter, start- og endepunkt Bue definert med tre punkt Bue definert med start-, ende-punkt og tangent Velg punkt/linje/region

Lasttilfellene og lastkombinasjonene utarbeides i hver sin dialog i Fd. I dialogene spesifiseres typer og varighetsklasser til lasttilfellene, samt typer og lastfaktorer for lastkombinasjonene.

## Beregninger

Etter at lastene, lasttilfellene og lastkombinasjonene er utarbeidet, kan beregningene av modellen gjennomføres. For beregninger uavhengig av designvalg, brukes kommandoen «Calculate», i fanen «Analysis». Mens for beregninger som inkluderer designvalg, benyttes «Calculate», i fanene «RC design», «Steel design» og «Timber design». [62]

Begge metodene får frem en liste med et utvalg analysetyper, som vist i Figur 5.7. Brukeren har her mulighet til å bestemme hvilke analyser som skal gjennomføres på FEM-modellen. Figuren til å beregningene som kan gjøres via «Analysis», har én mindre valgmulighet sammenliknet med beregninger via «RC design». Denne er navngitt «Design calculations», og er uthevet i grønn innramming. [62]

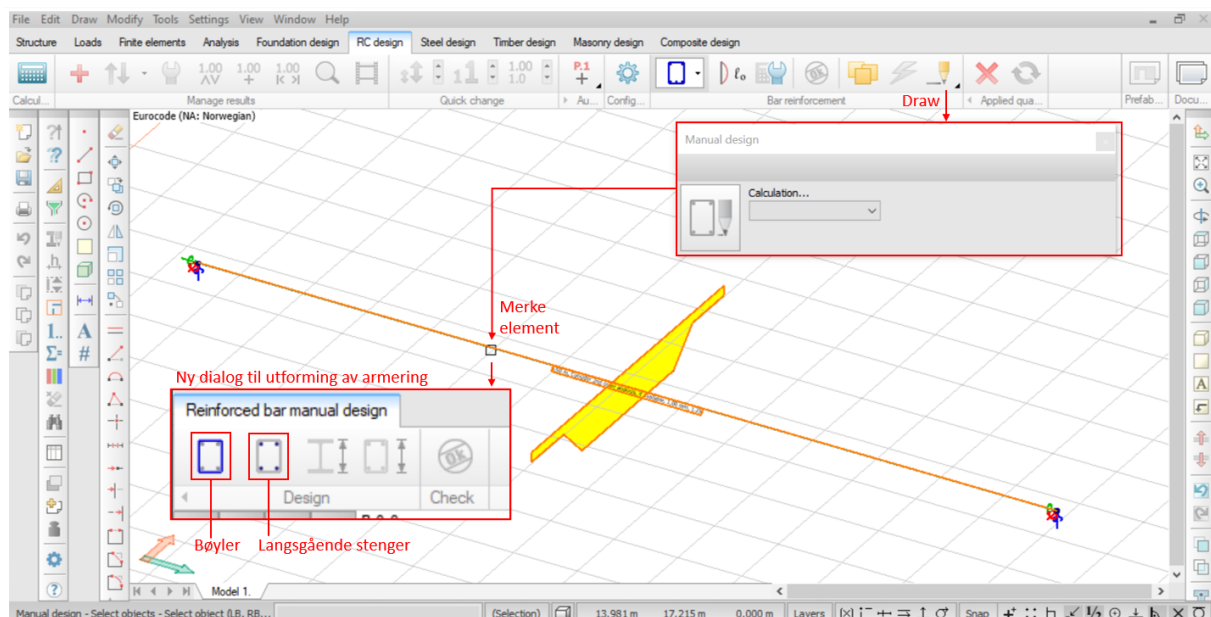


Figur 5.7: Beregninger i fanen "Analysis" og "RC design", figur utarbeidet fra [62]

## Armering av betongelementer

For prosjektering av armering i betongelementer, gjennomføres dette ved å ta i bruk verktøyene i fanen «RC design». Her kan armering blant annet modelleres i bjelker, søyler og avstivere. [28, s.3] Metoden for å gjennomføre dette, navngitt «Bar reinforcement», og har utformings-funksjoner i form av «Auto design» og «Manual design». [63]

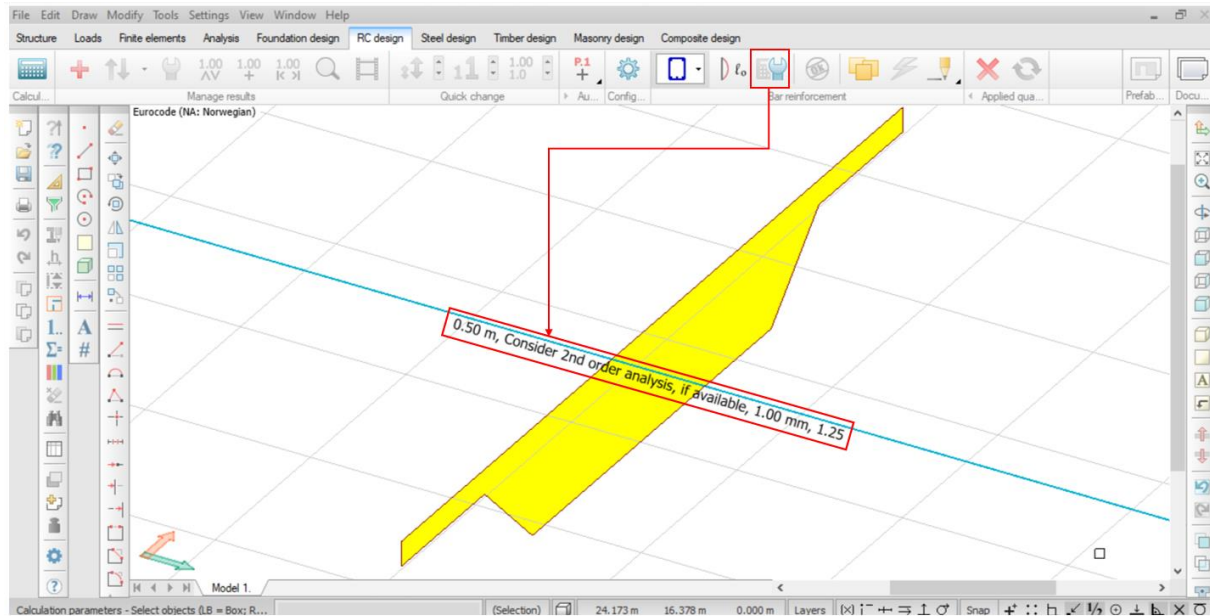
Manuell armering («Manual design») kan påføres uten at beregningene er gjennomført. Hvordan dette gjøres, er presentert i Figur 5.8. For manuell utforming brukes verktøyet «Draw», før elementet som skal armeres blir merket. Her kan det produseres bøyler og lengdearmering, i dialogen «Reinforced bar manual design». Utforming av stenger og bøyler skjer i tverrsnittsformen til elementet, før utstrekningen i lengderetning til begge typene defineres. [63]



Figur 5.8: Manuell armering i Fd

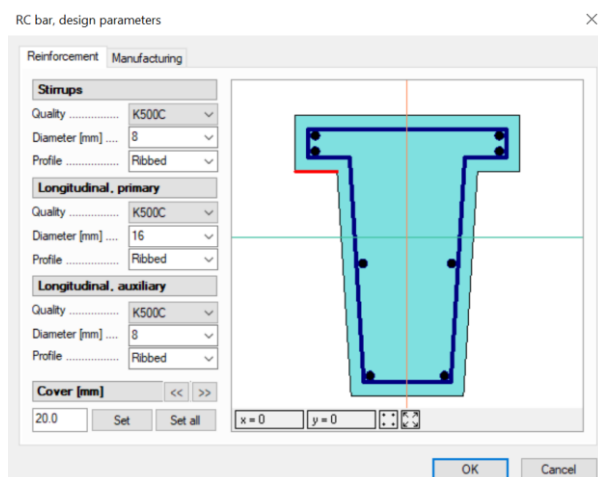
Når betongelementene er armert manuelt, analyseres disse i beregningene. Det er derfor av hensikt å vite hvilke parametere for utformingsbetingelser som benyttes i analysen. Parametere for utformingsberegninger er mulig å finne i «Calculation parameters» hvor parametere og analyseform defineres. I Figur 5.9 illustreres fremvisningen av de standardiserte utformingsparametere, hos et brudekkeelement. [63]





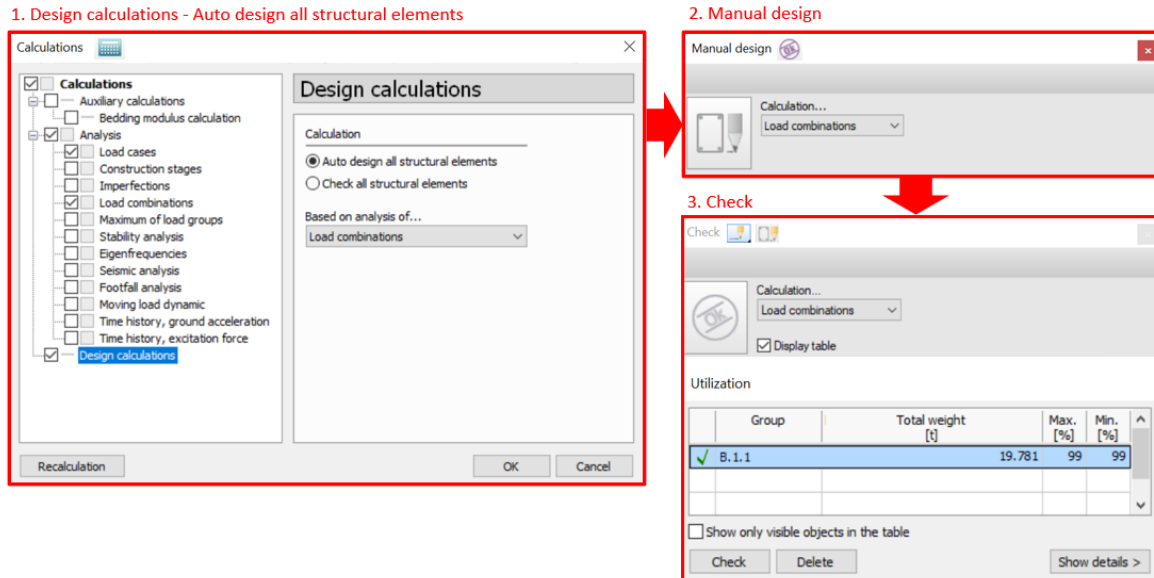
Figur 5.9: Utformingsparametere til armeringsanalysen i Fd

Automatisk armering («Auto Design, skjer ved å velge Auto-design når modellen analyseres som «RC design», som vist i den grønne innrammingen i Figur 5.7 over. Denne metoden betegnes som global Auto (RC) design, og påfører armering i alle betongelementer. Etter at analysen er kjørt, kan brukeren selv klikke på Auto design ikonet. Da fremkommer parameterne til armeringen, sammen med en utnyttelsestabell. Her kan det grønne spørsmålstegnet i Auto design dialogen, benyttes for å få opp parameterne til et spesifikt betongelement. [63] Innholdet i denne dialogen for et bjelke-element, med parametere og tilsvarende verdier, er presentert i Figur 5.10.



Figur 5.10: Auto design parametere til et brudekke i Fd

En fremgangsmåte for prosjektering av armering i betongelementer, hvor både automatisk- og manuell utforming benyttes, er presentert i 3 steg i Figur 5.11. Første steg er å ta i bruk beregningstypen «Design Calculations», som gir muligheten for å prosjektere elementet i sin helhet. Til dette benyttes analysen «Design calculations», med «Auto design all structural elements». Etter at beregningen er gjennomført, utføres detaljert manuell design, før endelig utforming blir gjennomført med komponenten «Check». [64]

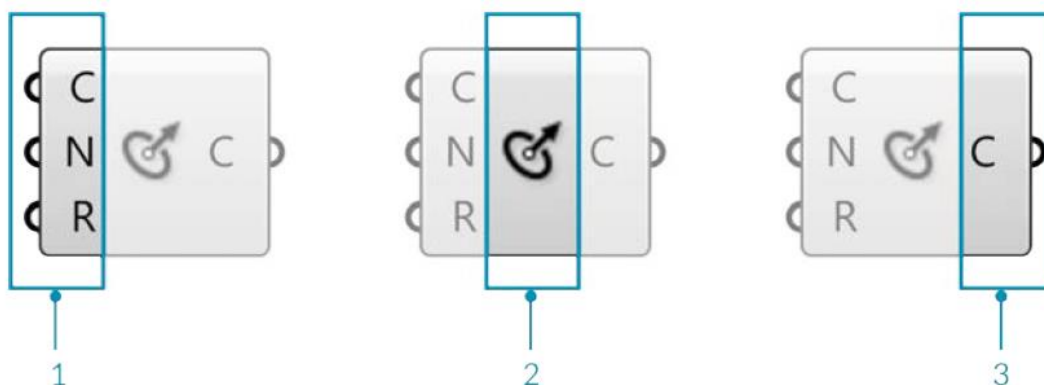


Figur 5.11: En av metodene for beregning og modellering av armering i Fd

### 5.1.3 Grasshopper

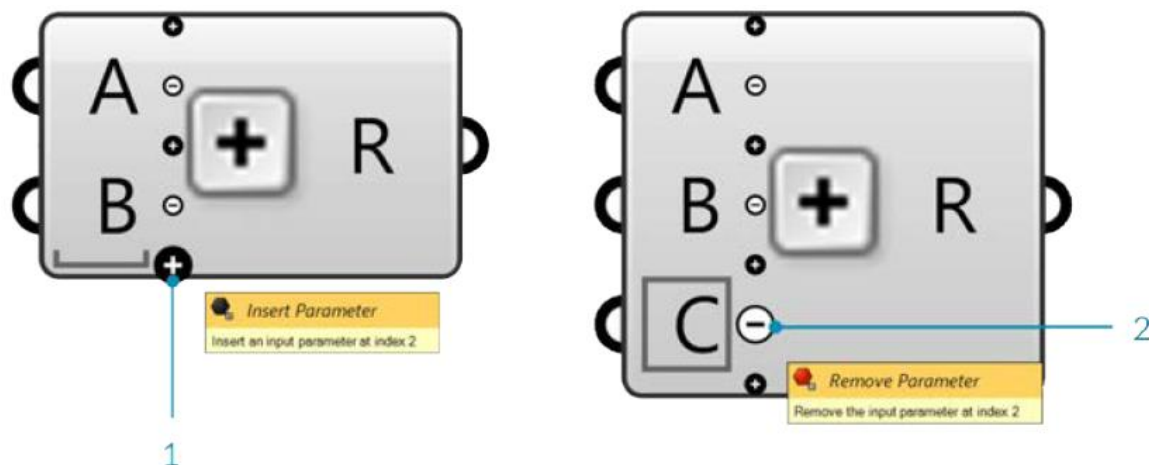
GH består i hovedsak av to typer objekter: parametere og komponenter. Her er det komponentene som utfører handlingene og gir resultater i form av data, mens parameterne lagrer data.

Den første objektstypen som vil bli beskrevet, er komponenter. Komponenter utfører handlinger, basert på dataen som føres inn i inngangs-tilkoblinger. De utfører ulike oppgaver [1, s.38], og representerer Rhino-geometri eller operasjoner i form av matematiske funksjoner. Felles for alle komponenter er at de består av innganger og utganger. Dette er eksempelvis presentert, for komponenten «Circle CNR» i Figur 5.12. Punkt 1 indikerer tre ulike innganger, punkt 2 indikerer type komponent, mens punkt 3 indikerer utgangs parameteren. [1, s.41]



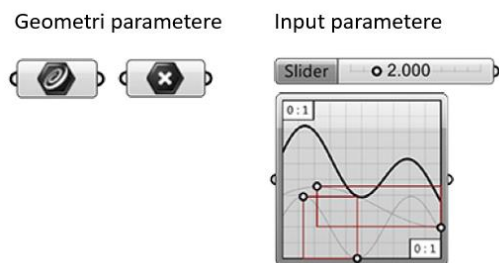
Figur 5.12: Oppbygningen til komponenter, med CNR komponenten som eksempel [1, s.41]

Enkelte komponenter gir mulighet for modifikasjon, ved reduksjon av antall innganger og utganger. Dette kan gjennomføres ved å benytte «Zoomable User Interface» (ZUI). Dette innebærer å zoome inn på en komponent på lerretet, hvor det da dukker opp tilleggsmuligheter. I Figur 5.13 presenterer punkt 1 metoden for å legge til innganger, mens i punkt 2 fremkommer metoden for å trekke fra innganger. De samme funksjonene kan også benyttes på enkelte utgangsparametere. [1, s.45-46]



Figur 5.13: Eksempel på «Zoomable User Interface» funksjonen på en komponenter i GH [1, s.46]

Den andre objektstypen i GH er parametere. Parametere lagrer data i form av tall, farger og geometri. Denne informasjonen sendes gjennom variere og til komponenter nedstrøms. Eksempler på noen parametere er presentert i Figur 5.14. [1, s.38]



Figur 5.14: Eksempler på parametere i Grasshopper

De fleste parametere har mulighet til å lagre to typer data: ustabile og vedvarende. Ustabile data er ikke permanente, og vil bli ødelagt hver gang løsningen utløper. Dette utløser en hendelse, som igjen må bygge opp løsningen og oppdatere dataen. Generelt er det meste av dataen som genereres, flytende, og ansett som ustabil data. [1, s.48] Mens vedvarende data, er spesifisert av brukeren selv, og avhenger av en annen type parameter. Et eksempel på dette er en punkt-parameter, hvor ett eller flere punkter defineres. [1, s.47]

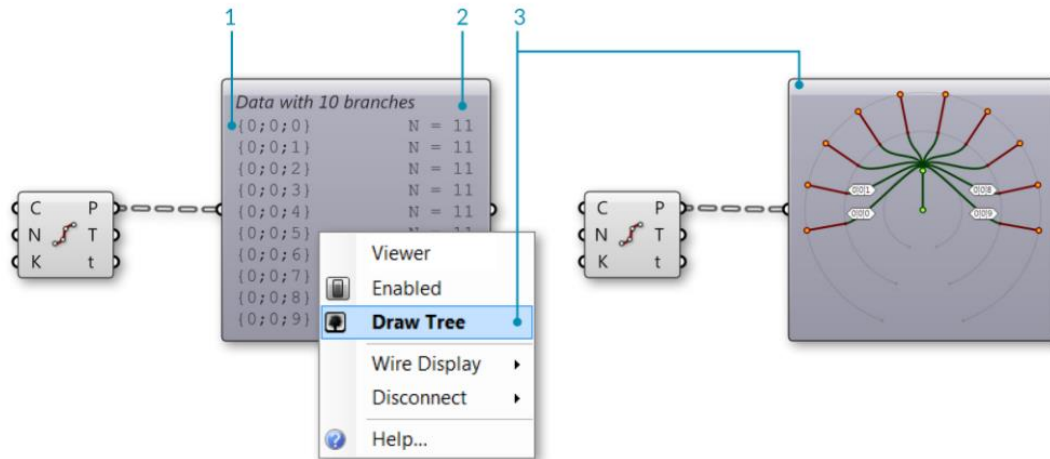
De ustabile parameterne i GH, tilbyr muligheter til å levere dataen til inngangene hos komponentene. De gir dermed mulighet til å opprettholde kontrollen på resultatene og mulighet til endringer av dem. To eksempler på slike parameter-komponenter er «Number Slider» og «Value List». Komponentene «Number Slider» er universelt mye brukt, som inngangs-parameter. Mens komponenten «Value List», er mer spesifikk, for produksjon av alternative utgangsverdier. [1, s.49]

### Data tree

GH systematiserer data i lister, og data-trær. Data trær er den norske oversettelsen av «Data Tree». Data trærne i GH identifiserer lister i lister. De inneholder dermed en hierarkisk struktur, for å sortere data som inneholder lister med del-lister. Slike datatrær fungerer på samme måte som mappestrukturen på en datamaskin. Datatrærne identifiseres med ulike «Path», mens listene i seg selv identifiseres med indekser. [1, kap.1.5.2]

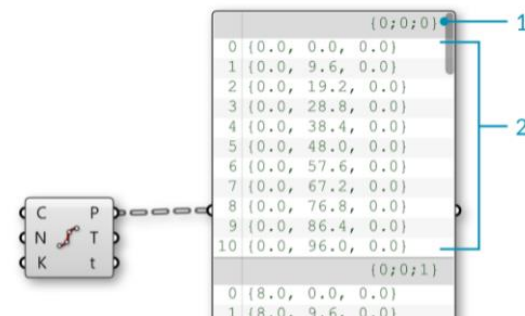
Data trær er komplekse, og kan være vanskelig å forstå. Derfor tilbyr GH, verktøy for å visualisere disse, som er et forsøk på å lettere forstå oppbygningen av listene, og deres innhold. Et verktøy for å visualisere et data-tre med tekst, kan gjøres med komponenten «Param Viewer». Denne er presentert i Figur 5.15, for utgangsdataen til en «Divide Curve» komponent. Her er resulterende data

presentert i form av 10 greiner i punkt 1, som korresponderer til de 10 kurvene som er produsert i komponenten oppstrøms. I punkt 2 fremkommer det at hver gren inneholder en liste på 11 punkter, som deler kurven inn i 10 deler. Ved å høyreklikke på komponenten, og velge «Draw Tree», fremkommer en visualisert fremvisning av treet. [1, kap.1.5.2]



Figur 5.15: Tre-grener [1, kap.1.5.2]

En annen komponent som presenterer hierarkiet til utgangsdataen hos en komponent, er «Panel». Denne er presentert i Figur 5.16, hvor «Path» (stien) (1) er fremvist på toppen av hver liste (2). Disse korresponderer til informasjonen presentert i punkt 1 og 2, i Figur 5.16. [1, kap.1.5.2]



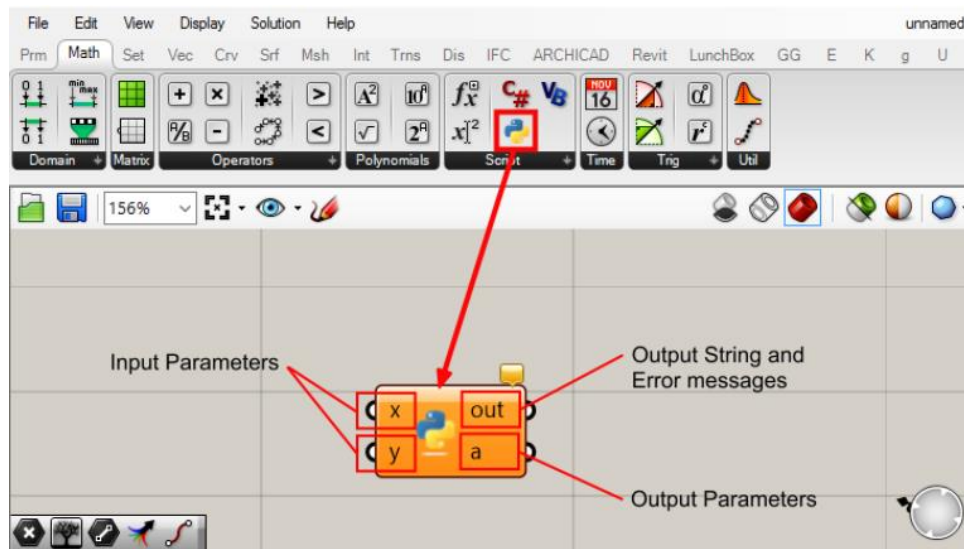
Figur 5.16: Data-tre med "Panel" komponenten [1, kap.1.5.2]

For å lage datatrær i GH, kan utføres direkte i komponentene, ved å endre strukturen av treet eller listen. Dette gir mulighet til å få tilgang til spesifikk data, og skjer ved å endre rekkefølge, identifisere og/eller sortere måten treet er bygd opp. Tre av manipulasjonene er «Flatten», «Graft tree» og «Simplify tree», og fremkommer i form av nedtrekksmenyer til komponentene. [1, kap.1.5.2]

«Flatten» fjerner alle nivåene av innholdet i treet, som da resulterer i én enkel liste. «Graft tree» har på den andre siden motsatt funksjon. Denne skaper en ny gren for hvert datainnhold (hver indeks hos en liste). For å få vekk overlappede grener i et tre, brukes «Simplify tree». Her forenkles inndelingsdefinisjonen av grenen, og innebærer eksempelvis at stien (0;0;1) forenkles til (1). [1, kap.1.5.2]

## GHPython

GH har tre skriptkomponenter som gjør det mulig å tilpasse logikken til GH-komponenter. De tar i bruk tekstprogrammering integrert i GH. En av disse komponentene er IronPython (Python). Denne brukes som utgangspunkt for egendefinert innhold av koder, og inneholder innganger og utganger som selv defineres av brukeren. [65] Hvor komponenten er plassert i menyen til GH, og hvordan en standardisert Python komponent ser ut, er presentert i Figur 5.17. Ved å dobbeltklikke på Python-ikonet, fremkomme et vindu som gir mulighet til å redigere og/eller opprette eget innhold. [66]

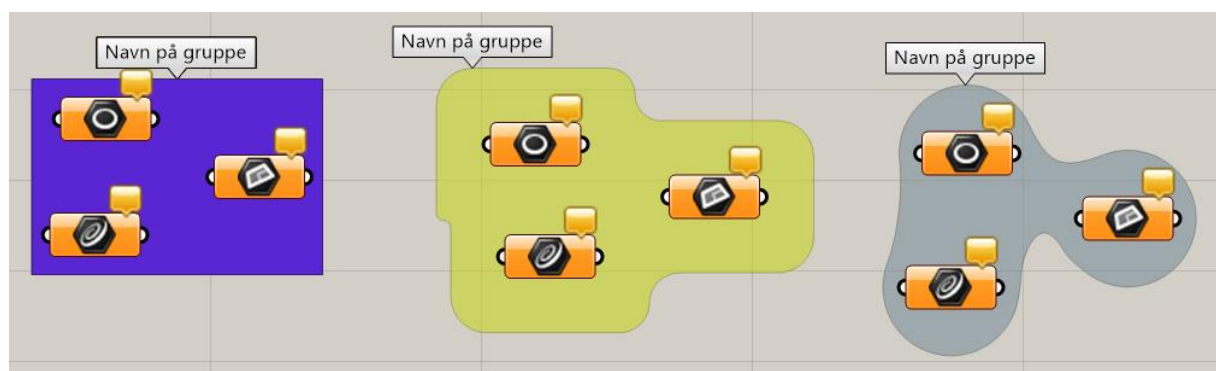


Figur 5.17: Plasseringen av Python komponenten i GH, som standardisert komponent.[66]

En slik komponent kan tas i bruk for å lage spesielle funksjonaliteter, og åpner opp for enormt potensiale bak standard-komponenter. En annen universell funksjon Python har, er å få tilgang til API-et hos andre programvarer, eksempelvis Rhino og Rvt [2]. [65] For å få tilgang til ønskede API-er, må disse importeres i vinduet, hvor kodene defineres i Python-komponenten. [66]

### Gruppering

For produksjon av et lesbart og forståelig skript, kan komponentene i GH grupperes med fargekoder og overskrifter. Når komponenter grupperes, kan den raskt velges og flyttes rundt, som en samlet kohort. Opprettelse av grupper kan gjøres ved å skrive Ctrl+G, etter at de valgte komponentene er merket. Egendefinerte farger, gjennomsiktighet, navn og utformingstype kan bestemmes til gruppene. [1, s.20] Mulige gruppeprofiler, eksempler på farger, samt hvordan navnet på grupper presenteres, er illustrert i Figur 5.18.

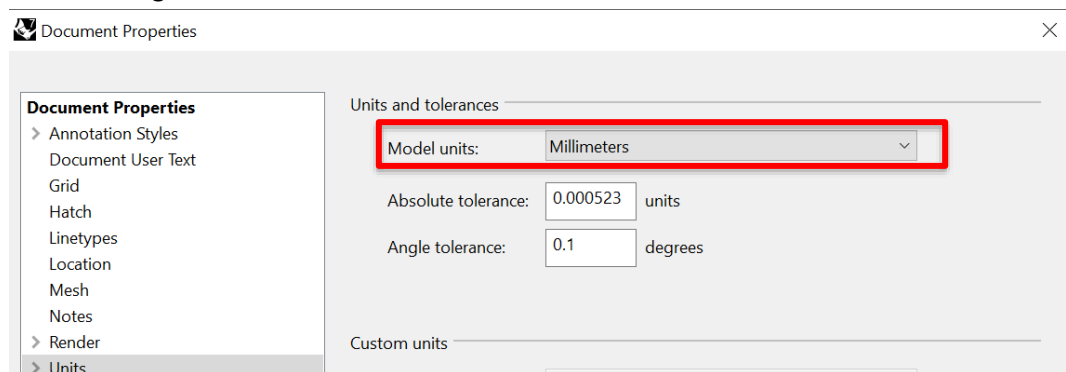


Figur 5.18: Måter å gruppere komponenter[44]

### 5.1.4 Rhino.inside.revit

Versjonen 0.1 av RiR, er i hovedsak en oppdatering av grensesnittet til Rhino inne i Rvt. Endringene gjør det enklere å bruke RiR, ved at tilgjengelige oppdateringer er synlig i programvaren. Denne versjonen har i tillegg utarbeidet en metode for å få tilgang på GH-skriptene, på en enklere måte gjennom Rvt-UI<sup>3</sup>. [49]

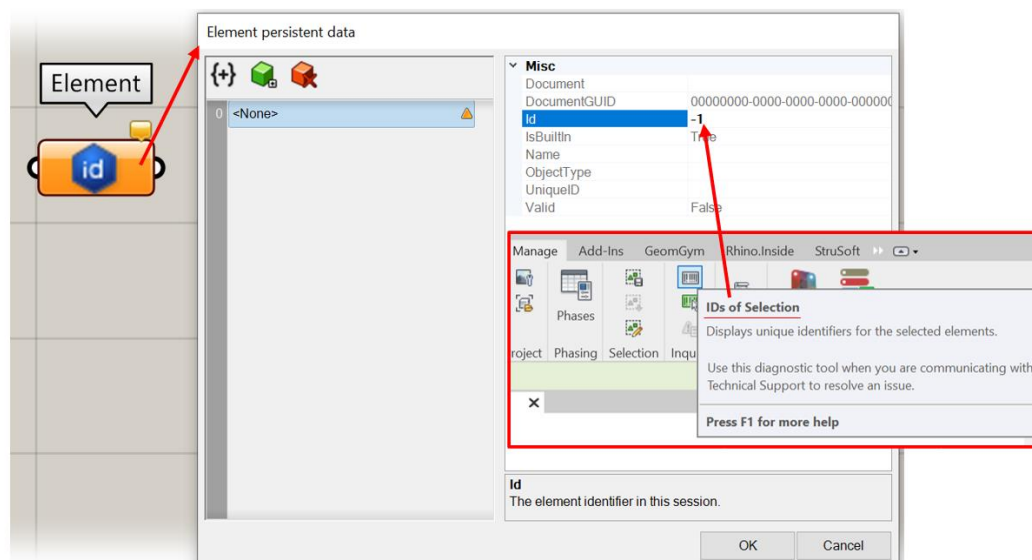
Ved bruk av Rhino og GH i Rvt, er det nødvendig at Rhino og Rvt opererer med samme standardenheter. Dette er nødvendig for å få korrekt forhold på geometri og elementer. Enhetene i GH er hentet fra Rhino, og følger enhetene her [1, s.31]. For å endre enhetene til Rhino i Rvt, kan dette gjennomføres i fanemenyen Rhino.Inside, hos Rvt. Dialogen hvor endringene gjennomføres er illustrert i Figur 5.19.



Figur 5.19: Hvordan endre enheter i GH i Rhino «Document Properties» menyen [1, s.32]

### Komponenter

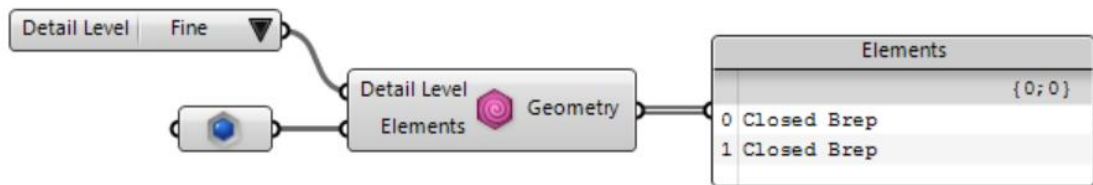
En av komponentene som referer til elementer i Rvt, er «Element». Denne refererer til et spesifikt element, i henhold til elementets id. Får å få en ID på elementet, må det det være utplassert i arbeidsområdet til Rvt. Rvt-elementets ID er mulig å finne i fanen «Manage», som vist i Figur 5.20. ID-en hentes fra Rvt, og brukes i dialogen som fremkommer ved å velge «Manage Revit Element Collection» i nedtrekksmenyen. [67]



Figur 5.20: Finne elementers ID og bruke dem i GH

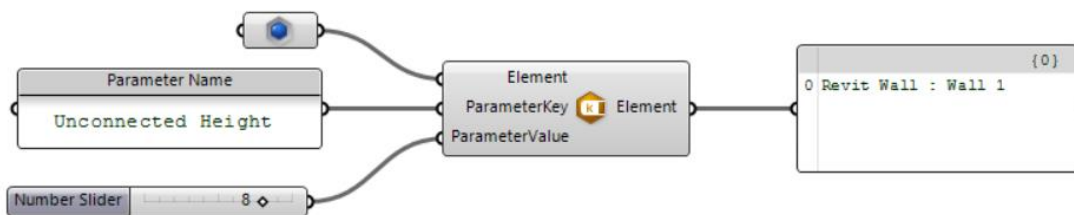
<sup>3</sup> UI = User Interface, som på norsk betyr brukergrensesnitt

Forekomster er individuelle geometriske elementer plassert i en Rvt-modell (enkel dør, vegg). Rvt-elementer arver forekomster, med en rekke parametere fra deres kategori og type. Geometrien til disse forekomstene kan hentes ut i Rhino, ved å ta i bruk komponenten «Element Geometry». Her avhenger geometrien av detaljeringsgrad i Rvt. Et eksempel hvor denne komponenten er brukt, er vist i Figur 5.21: [67]



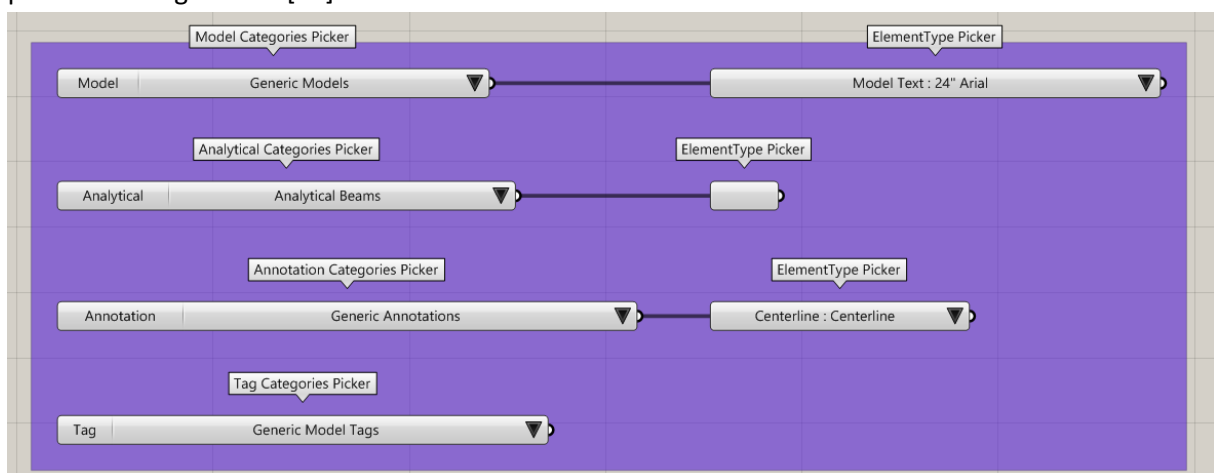
Figur 5.21: Bruk av komponenten «Element Geometry»

Parameterne til Rvt-elementer, kan hentes ut og oppdateres i GH. Det er flere komponenter som henter ut parameterne til et element i Rvt, hvor en av dem er «Get Element Parameter». For å endre parameterne til et spesifikt element, kan den benyttes sammen med «Set element parameter». Her må navnet på parameteren, elementet parameteren tilhører, og ny verdi på parameteren, defineres. Dette er illustrert i Figur 5.22. [68]



Figur 5.22: Bruk av komponenten «Set Element Parameter»

Ved å arbeide med Rvt API i GH, innebærer som oftest håndtering av Rvt-typer og egendefinerte familier. Dette er det viktigste aspektet av modelleringsprosessen, og handler mer om at data blir modellert, fremfor geometrisk utforming. To komponenter som er hensiktsmessig å ta i bruk for produksjon av Rvt-elementer, er å hente inn kategorier og typer fra Rvt, ved bruk av komponentene presentert i Figur 5.23. [69]



Figur 5.23: Hente ut kategorier og typer fra Rvt til GH

## GHPython for Rhino.Inside.Revit

Siden RiR bruker Rhino og GH inne i Rvt miljøet, får også skriptkomponenten Python i GH, tilgang til Rvt API-et. For å få tilgang til Rvt API-et i Python, må CLR (Common-Language-Runtime) modulen importeres. Dette gjøres for å legge til nødvendige bibliotekreferanser fra Rvt. Her kodes de mest vanlige API språkene, som Rvt bruker for å produsere data. Denne informasjonen må alltid ligge først i et Python-skript. [66]

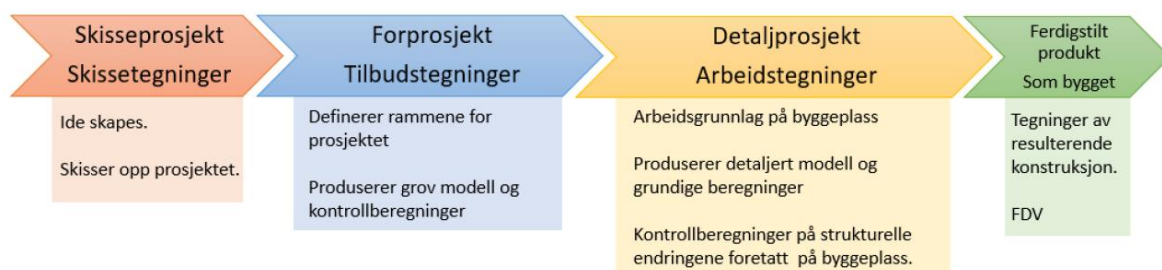
Alle endringer i en Rvt-modell må fullføres i en transaksjon, i Python-skriptet. RiR bruker metoden «Revit.EnqueueAction», som pakker funksjonen inn i en transaksjon og kobler på Rvt, når transaksjonen er klar til å foreta endringer i det aktive dokumentet. Denne mekanismen er utformet for å sikre at kun ett Rvt-tillegg, kan gjøre endringer i dokumentet om gangen. [66]

## 5.2 DS prosjektering

Programvarene som er beskrevet over, skal brukes i samsvar med prosjekteringsprosessen hos bedriften Dagfin Skaar AS (DS). Dette er et rådgivende ingeniørfirma, og inndelt i seks avdelinger. En av avdelingen er RIB<sup>4</sup>, og de har ansvar for prosjekteringen av bæresystemet til konstruksjoner. En mer detaljert presentasjon av bedriften er beskrevet i forprosjektet, se [3]. RIB-prosjektering hos DS, gjennomføres blant annet ved å ta i bruk programvarene Rvt og Fd, for produksjon av geometrisk- og beregningsmodeller. Disse programvarene har i dag ikke en kompatibel samhandling hos DS, hvilket fremkommer fra studiet gjort i [3, kap. 7.1.3]. Dette er et studie hvor prosjekteringsprosessen hos DS ble kartlagt. Her fremkommer det at praksisen hos DS i dag, utarbeider to separate modeller, for produksjon av CAD- og FEM-modell. [3, s.41] Videre innhold i dette kapitlet, er hentet fra forprosjektet til denne masteroppgaven, og definerer casen for prosjekteringsprosessen og prosjekteringsinnhold. [3]

### 5.2.1 Prosjekteringsprosessen

Prosjekteringsprosessen hos DS deles inn i fire faser, hvor hver fase innebærer ulik detaljeringsgrad, og type tegningsprodukter. Disse fire fasene er presentert i Figur 5.24, med innhold av korte beskrivelser. [3, s.39] I studiet ble det i henhold til implementering av parametrisert design i denne prosjekteringsprosessen, konkludert med at prosjekteringsarbeid gjennomført på en utførelsesentreprise, gir størst sannsynlighet for at én rådgivende ingeniør får bli med i alle fire fasene av prosjekteringsprosessen. [3, s.66]



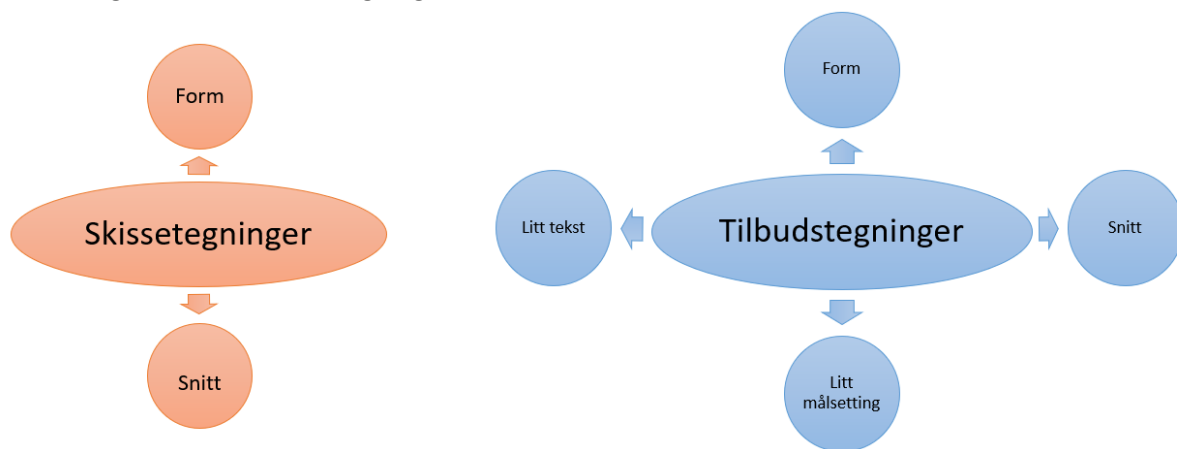
Figur 5.24: Fasene som prosjekteringsprosessen inneholder [3, Figur7.5]

Tegningsproduktene i de ulike fasene av prosjekteringsprosessen, er presentert i Figur 5.25 og Figur 5.26. I første avbildning, presenteres produktene for skissetegninger i «Skisseprosjektet», og

<sup>4</sup> RIB = Rådgivende byggingeniør

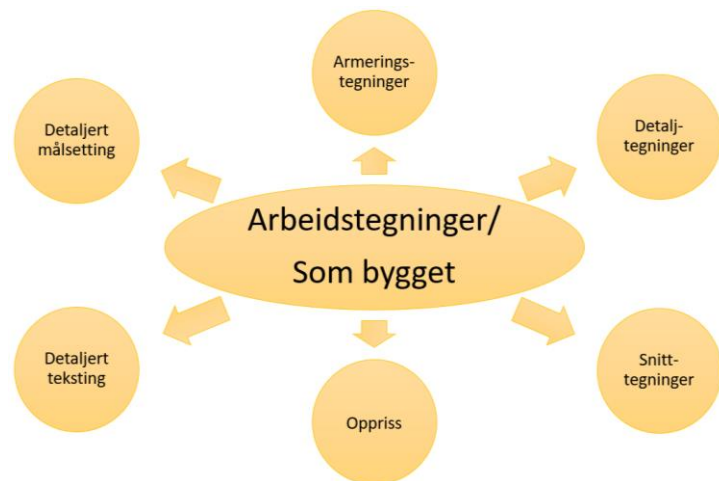


tilbudstegningene i «Forprosjektet». I begge fasene skal det utarbeides både plan- og snitt-tegninger. Grad av detaljer er ulikt, fordi tilbudstegninger består av litt teksting og målsetting, mens dette ikke nødvendigvis utføres i skissetegninger. [3, s.40]



Figur 5.25: Tegningsprodukter i skisse- og forprosjekt, [3, Figur 7.6]

For den tredje fasen i prosjekteringsprosessen, «Detaljprosjektet», produseres detaljer i form av å beregne og modellere knutepunkter og armering. Arbeidstegninger inneholder elementene med grundig teksting og målsetting, i armerings-, oppriss- og snitt-tegninger. Disse produktene produseres også i prosjekteringsfasen «Som bygget». Her justeres endringer, slik at tegninger tilsvarer faktisk utført konstruksjon på byggeplassen. [3, s.41]

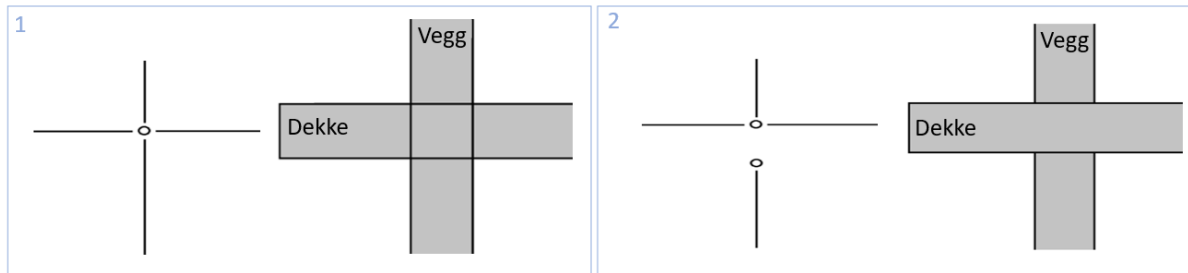


Figur 5.26: Tegningsprodukter i detalj og slutfasen, [3, Figur 7.7-7.8]

### 5.2.2 Samhandling mellom Revit og FEM-Design

For å få til import i Fd, må Rvt-modellen være analytisk. Det innebærer at elementene i modellen må inneholde analytiske linjer, som Fd kan lese. Både utarbeidelse av analytisk modell, samt gjennomføring av analyse, er en tidkrevende prosess. Et annet problem her, er at Fd ikke gjenkjenner egenskapene til elementer i RVT-filen. Dette innebærer at det må gjennomføres en manuell kontroll, for å spesifisere tilsvarende elementer i Fd biblioteket. [3, s.43-44]

Ved oppbygging av en analytisk modell, kan det medføre ukorrekt fremvisningen av en intuitiv produksjonstegning. Eksempelvis vil ikke en analytisk modell for oppbygging av vegger og dekker, få den detaljeringsgraden som er nødvendig for produksjonstegningene. Dette er illustrert i Figur 5.27, hvor tilfelle 1 er analytisk korrekt, men produksjonsmessig feil. Mens i tilfelle 2, er produksjonsmessig riktig, men analytisk ukorrekt. [3, s.44-45]



Figur 5.27: Analytisk modell versus visuell fremvisning, figur 7.12 og 7.14 hentet fra [3, s.44-45]

I de fleste prosjekter er det ikke hensiktsmessig å utarbeide en like detaljert statisk modell, som geometriske modell. Beregningsmodeller må forenkles, og renkes for detaljer som kan gi problemer for gjennomføring av FEM-analysen. Fra spørreundersøkelsen gjennomført i [3], hvor erfaringer med utveksling av Rvt- og Fd-modeller ble innhentet, opplevde en av respondentene at arbeidet som kreves for optimal import og eksport av RVT- og FEM-modeller, ikke overstiger nytten. [3, s.43]

### 5.2.3 utfordringer med tradisjonell modellering

#### Revit

Konstruksjonstypene som har størst begrensning innenfor tradisjonell modellering i Rvt, er elementer med tredimensjonal variasjon. Dette innebærer krumninger i alle tre plan. Det er ikke er mulig å modellere slike elementer i Rvt, og samtidig opprettholde kontroll på kurver og endringer i dimensjoner. Det kommer av begrensninger i metodene, som foreligger i programvaren. Et typisk eksempel på et konstruksjonselement som er problematisk å modellere i Rvt, er et brudekke med krumning i tre plan. [3, kap.7.1.5]

En funksjon som utvider metodene for modellering i Rvt, er familier. Dette er en tidkrevende prosess, samtidig som den er fordelaktig, da familier fungerer som egne Rvt-modeller. De kan kopieres og endres på, uavhengig av hovedmodellen de benyttes i. De kan også tildeles spesifikke egenskaper og statisk sammenkobling med andre elementer. [3, kap.7.1.5]

Enkelte problemer som respondentene opplevde med å modellere ikke-lineær geometri i Rvt er som følger: [3, kap.7.1.5]

- Begrenset kompetanse innenfor verktøyene.
- Modellere kurvatur i to plan.
- For å modellere krumningene og variasjonene, må brukeren benytte flere arbeidsoperasjoner.
- Krevende å modellere nøyaktig, da alt må Custom-lages.
- Det er komplisert å modellere objekter som består av ikke-lineære endringer.
- Brukertilskelen for Add-ins er høy.

#### FEM-Design

Det er som oftest krevende å modellere ikke-lineære strukturelle elementer, tradisjonelt i Fd. Det kommer av at modelleringsverktøyene i programvaren ikke er avanserte nok. Men de geometriske effektene til et element, vil påvirke beregningsresultatene i Fd. En betydelig krumning vertikalt, kan

eksempelvis gi gunstige trykk-krefter i en bjelke, som igjen er vesentlig for beregningene. [3, kap.7.1.5]

For elementene bjelke og dekke, med krumninger i to plan, finnes det ikke en intuitiv metode for modellering, og samtidig ha kontroll over geometrien. Dette har tidligere blitt løst ved å dele en irregulær form opp i små korte linjer, som til sammen utgjør en visuell ikke-lineær kurve. [3, kap.7.1.5]

En annen måte det er mulig å utforme én ikke-lineær kurve, er å utforme en geometrisk ikke-lineær kurve, som tildeles et strukturelt element. Ulempen med denne metoden igjen, er at elementet fort får unøyaktige mål. Å få nøyaktige mål blir ansett som tidkrevende, og dermed unødvendig sammenliknet med nytteverdien. [3, kap.7.1.5]

### 5.3 Brudekket

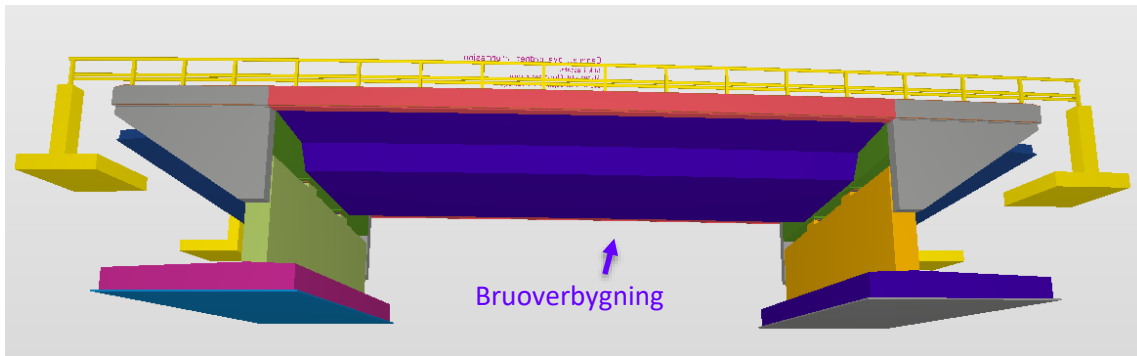
I dette prosjektet er ikke målet å utvikle en innovativ utforming på et brudekke. Målet er å finne en effektiv måte for håndtering av revisjoner til brudekker, med krumninger i mer enn to plan. Det kommer av resultatene fra forprosjektet [3], som er presentert i delkapittelet over. For å definere grensesnittene til brudekker, som den parametriske modellen skal favne, var det hensiktsmessig å ta utgangspunkt i et reelt bru-prosjekt. Det ble derfor valgt å ta utgangspunkt i den geometriske utformingen og beregningsfaktorene, til brudekket til Sagåne bru. For lastplasseringene til vindlast, ble prosjektet Birkelid bru benyttet som utgangspunkt. Begge disse bruene er kategorisert som vegbruer. Her defineres en veibru, som er en bru som fører kjøreveien over en hindring [70].

#### 5.3.1 Sagåne bru

Kriteriene for valg av type vegbru, ble tatt i henhold til at den skulle bestå av krumninger i tre plan. Med dette kriteriet, ble det i dialog med Robert McShane, Svein Erik og Martin bestemt å ta utgangspunkt i Sagåne bru. Innhold som presenteres i dette kapittelet er derfor i hovedsak hentet fra RIB-tegninger [Vedlegg B] og beregningsrapporten [Vedlegg A] fra Rambøll, til dette prosjektet. Forutsetningene for prosjekteringen, er blitt utført i henhold til SVV håndbøker og NS-er.

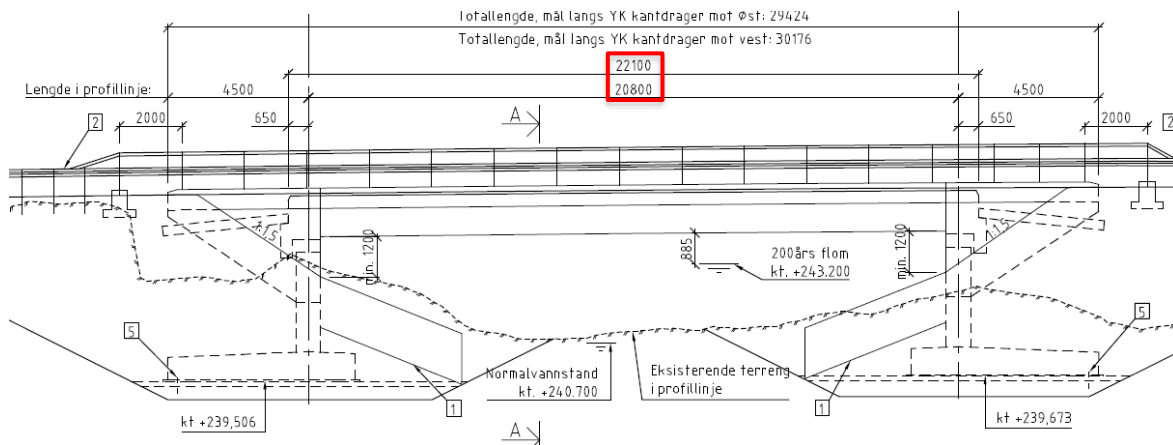
Sagåne bru er en veibru over elveløpet Kvernåni. Denne bruene er en del av nye ringvei 9, fra Besteland til Helle, i Valle kommune, i Aust-Agder. Strekningen er antatt å åpne i oktober 2021, og er prosjektert med en ÅDT på 1350 (år 2040) og en fartsgrense på 80 km/t. Disse to siste faktorene, klassifiserer bruene til å inngå i veiklasse er H2 [71]. Videre er byggherren på prosjektet, Statens Vegvesen region sør, hvorav prosjekterende aktør er Rambøll Norge AS.

Det er en kontinuerlig plasstøpt ett-spenns bjelkebru, som både er slakkarmert og spennarmert. Et av elementene som er prosjektert i dette prosjektet, er overbygningen til bruene. Her er overbygningen opplagret på både glide- og fastlager. Videre blir dette elementet omtalt som brudekke. Prosjektet innebærer også andre elementer, men disse neglisjeres på grunn av tidsbegrensningen på masteroppgaven.

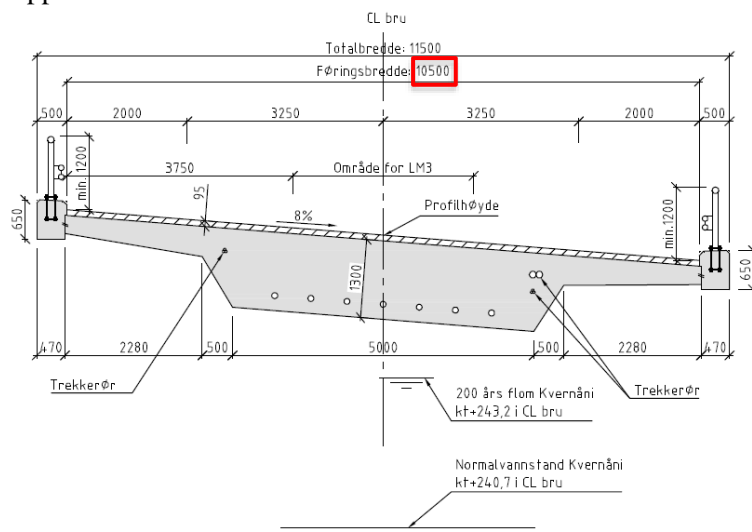


Figur 5.28: Hva som inkluderes fra Sagåne bru, bilde hentet fra IFC-fil til Sagåne bru

I de røde rektanglene i Figur 5.29, fremkommer det at brudekket har en platelengde på 22,1 meter, en føringsbredde på 10,5 meter, og en spennvidde på 20,8 meter. I tillegg består den av krumninger i vertikalplan, horisontalplan og tverrfall. Her er tverrfallet på 8 prosent, horisontalkurveradien på 450 meter og vertikalkurveradien på 4000 meter.



Oppriss



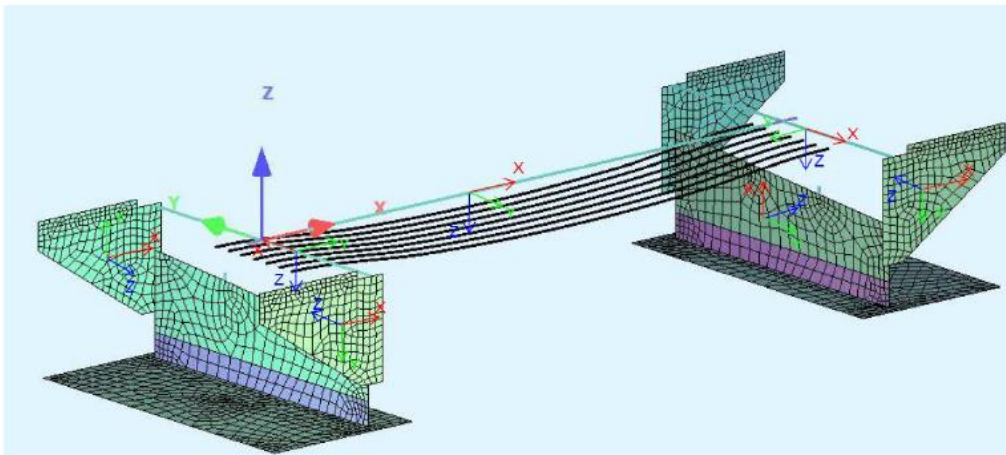
Snitt A-A

Figur 5.29: Oppriss og snitt av Sagåne bru

I modellen og beregningene for slakkarmeringen, ble kvaliteten B500NC benyttet. Her ble det benyttet to ulike overdekningsverdier for brudekket. Disse var definert i henhold til OK og UK av brudekke, hvor begge hadde eksponeringsklasse XD1. Nominell overdekning for OK ble på 75 millimeter, mens på UK ble den 65 millimeter. I tillegg til overdekningen, var det beregnet en minimumsarmering, i henhold til N400 og NS-EN 1992-1-1, 9.3.

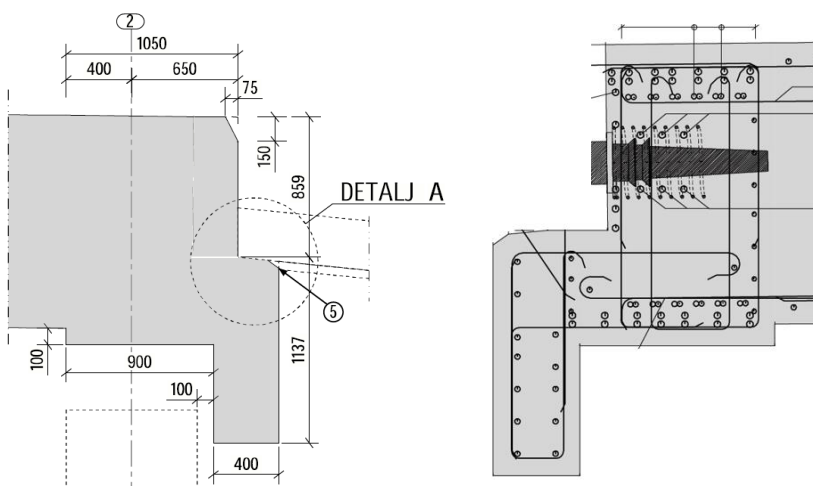
### 5.3.2 Idealisert modell /Statisk modell

Både brudekket og endetverrbærerne ble modellert i SOFiSTiK, som bjelke-elementer med ett gitt tverrsnitt. Dette er visualisert i Figur 5.30. Brudekket er her opplagret på to tverrbærere, som tilsvarer opplagringsbetingelsene av et brudekke opplagret på to langkar. Siden brudekket er opplagret på to lager på hvert langkar, ble disse plassert som opplegg til endetverrbærerne. Brudekket er modellert horisontalt i begge retninger, med eksakt tverrsnittsgeometri.



Figur 5.30: FE-modell av Sagåne bru

Tverrbærer-forankringen er støpt sammen med brudekket, ved at det er armert sammenhengende. Brudekke opplagres som fast innspent på tverrbærer da delene er støpt som en enhet.



Figur 5.31: Tverrbærer-forankringen til brudekket av Sagåne bru

Siden denne masteroppgaven er avgrenset til ikke å omhandle kontroll av analyseresultatene, blir kontrollen av SOFiSTik modellen opp imot manuell kontroll neglisjert. Men dette skal gjennomføres i et reelt prosjekt, da det i N400 punkt 1.3.3 er et minimumskrav å manuelt kontrollere påført egenlasten, i analysemodell. Hensikten med dette, er å kontrollere at randbetingelser og geometri, er riktig modellert i FE-modellen.

### 5.3.3 Inndata til lastdimensjonering

Laster som brudekket i Sagåne bru er dimensjonert for, er listet opp under. Her er lastene for setninger, påkjøring av kjøretøy, horisontale trafikklaste, kryp og svinn, jordskjelv, og lagerfriksjon ikke videre utdypet. Det kommer av at de blir neglisjert i dette prosjektet, grunnet tidsbegrensningen på oppgaven. Dette påvirker ikke resultatene i oppgaven, da modellens resultater ikke tas med i betraktning. De aktuelle lasttypene i denne masteroppgaven, er beskrevet i Vedlegg D, med tilhørende karakteristiske lastverdier.

1. Egenlaster
2. Trafikklaste vertikalt
3. Trafikklaste horisontale
4. Vindlaster
5. Temperaturlaster
6. Lagerfriksjon
7. Påkjøringslaste fra kjøretøy
8. Jordskjelv
9. Kryp og svinn

### Lastkombinasjoner

Superponering av lastene ble gjennomført i SOFiSTik, i henhold til kombinasjonslikninger i NS-EN 1990. Men i denne masteroppgaven er gjeldende lastkombinasjoner og lastfaktorer, hentet fra tabellen på side 17 fra SVV v668. Dette valget kommer av kjennskapet til disse lastkombinasjonene, fra bacheloroppgaven som ble skrevet i 2019. V668 er en universell veileder og en åpen ressurs. Den blir derfor vurdert som generell, med tanke på å standardisere brudekker, med revisjonsmuligheter. Disse lastkombinasjonene, med tilsvarende lastfaktorer er presentert i Tabell 5.3. De gjelder for små og mellomstore betongbruer, og inkluderer både lasttilfellene i brudd- og bruksgrensetilstand. [72, s.17]

Tabell 5.3: Lastkombinasjoner og lastfaktorer [72]

Lastkombinasjoner	G	PT	CSR	TR	TE	V-TR	V
ULS-STR							
1 (6.10a) – m/TR	1,35	0,9/1,1	0,0/1,0	0,95	0,84	1,12	-
2 (6.10a) – u/TR	1,35	0,9/1,1	0,0/1,0	-	0,84	-	1,12
3 (6.10b) – TR dom	1,20	0,9/1,1	0,0/1,0	1,35	0,84	1,12	-
4 (6.10b) – TE dom	1,20	0,9/1,1	0,0/1,0	0,95	1,20	1,12	-
5 (6.10b) – V-TR dom	1,20	0,9/1,1	0,0/1,0	0,95	0,84	1,60	-
6 (6.10b) – V dom	1,20	0,9/1,1	0,0/1,0	-	0,84	-	1,60
SLS-KAR							
1 TR dom	1,0	1,0	1,0	1,0	0,7	0,7	-
2 TE dom	1,0	1,0	1,0	0,7	1,0	0,7	-
3V dom	1,0	1,0	1,0	-	0,7	-	1,0
SLS-OFTE							
1 TR dom	1,0	1,0	1,0	0,7	-	-	-
2 TE dom	1,0	1,0	1,0	0,2	0,7	-	-
SLS-PERM							
1 TR dom	1,0	1,0	1,0	0,5	-	-	-
2 TE dom	1,0	1,0	1,0	0,2	0,7	-	-

- G – egenvekt
- PT – forspenning
- CSR – kryp/svinn/relaksasjon
- TR – trafikklast
- TE – temperaturlast
- V-TR – vindlast på bru med trafikk
- V – vindlast på bru uten trafikk

Forkortelsen ULS står for bruddgrensetilstand, mens SLS for bruksgrensetilstand. Videre står forkortelsene STR for kapasitetskontroll, mens KAR, OFTE og PERM står for karakteristisk-, ofte forekommende- og tilnærmet permanent kombinasjon. [72, s.16]

## 6 Metode

Dette kapittelet skal gi en systematisk beskrivelse av metodene, som ble tatt i bruk i dette prosjektet. Prosjektet innebar gjennomføring av fire ulike metoder, listet opp under. Hver av metodene vil bli beskrevet grundig, for at resultatene skal kunne etterprøves, i henhold til gyldighet og holdbarhet.

- Litteraturstudie
- Innhente erfaringer fra andre industrier
- Anvendte programvarer
- Parametrisk prosjektering

### 6.1 Litteraturstudie

Litteraturstudiet var det første som ble utført. Hensikten med dette, var å opparbeide kunnskap for et teoretisk grunnlag, innenfor relevante temaer. Her ble litteraturstudiet gjennomført i henhold til to formål. Det første formålet innebar innhenting av litteratur, til bruk som teorigrunnlag. Mens det andre formålet, omhandlet innhenting av tidligere studieoppgaver. Målet var her, å sette oppgaven i et bredere perspektiv.

Litteraturstudiet bestod av en søkeprosess. Denne prosessen ble gjennomført i henhold til fire strategier. Dette ble gjort, for å kunne evaluere kvaliteten på søkene, samt gi mulighet til å etterprøve funnene. De fire strategiene er listet opp under og vil videre utdypes, i henhold til formål én og to.

1. Definere tematikk
2. Valg av søkemotorer
3. Valg av søkeord
4. Gjennomføring av søk

Startet søkeprosessen med å definere tematikken til formål én og to. Her ble det bestemt at relevant tematikk som teorigrunnlag, var CAD, FEM, parametrisk design og parametriske verktøy. For å kunne finne litteratur om disse temaene, ble det innhentet synonymer og relaterte ord, til hvert av disse begrepene. Disse er presentert i Tabell 6.1, hvor enkelte av synonymene er hentet fra oppslagsverkene Google og Wikipedia.

Tabell 6.1: Synonymer til temaene i denne oppgaven

Tema	Synonym	Tema	Synonym
CAD [73]	Computer Assisted Detection Dataassistert konstruksjon Computer-aided design	Parametrisk design [74]	Parametric design Generative design Design computing Visual Programming language
FEM [24]	Finite Element Method Endelige elementmetoder Kraft-forskyvningsmetoden	Parametriske verktøy	Parametriske programvarer Parametric programs Parametric modelling



			Parametric calculation
--	--	--	------------------------

Til formål to ble det besluttet at tematikken skulle omhandle parametrisk design/modellering/prosjektering, av brukonstruksjoner. Her ble både design/modellering og prosjektering inkludert, da det var av usikkerhet hvordan disse var benyttet i andre vitenskapelige caser.

Etter at tematikkene var definert, ble neste steg i søkeprosessen å gjennomgå en utvalgelse av søkemotorer for litteratursøkene. De utvalgte søkemotorene, er presentert i Figur 6.1. Her ble det besluttet å ta i bruk Oria som hoved-søkemotor. Andre tilsvarende søkemotorer som GoogleScholar, ScienceDirect og Scopus, ble neglisjert for å begrense omfanget til formål to.



Figur 6.1: Søkemotorer benyttet i litteraturstudiet [75, 76]

Oria er en «metadatabase» [77, s.57], og er det nettbaserte biblioteket til UiA. I denne databasen ble bøker og studentoppgaver benyttet. Der bøker ble benyttet til formål én, mens studentoppgaver ble benyttet til formål to. For søkemotorene Youtube og Google, ble disse kun benyttet som oppslagsverk til spesifikke bruksområder. Det kommer av at internettet inneholder en stor informasjonsmengde, som har en enorm økning [78, s.151]. Her bestod bruksområdene av å finne manualer for de relevante programvarene, samt erfaringer gjort av andre; innenfor bruk av parametriske programvarer.

Til formål én, ble søkeord valgt i henhold til begrepene som definerte temaene. I tillegg ble flere av synonymene benyttet som søkeord. Søkene ble utført i de tre søkemotorene, presentert i Figur 6.1. Mens søkingen pågikk, ble søkeordene utvidet, da mer kunnskap og tilknytninger ble etablert. De endte blant annet opp med å inkludere søkeordene GH og Dynamo.

Valg av søkeord til formål to, ble også tatt i henhold til tematikken. Her ble det spesifikt valgt å benytte søkeord på norsk og på engelsk. Hvilke søkeord som ble brukt, er presentert i Tabell 6.2. Her ble kun søkemotoren Oria benyttet, hvor det var nødvendig å uttrykke sammensatte ord med anførselstegn, for å få relevante treff.

Videre ble søkene til formål to, utarbeidet som avanserte søk. Dette var søk i Norges bibliotek, og ikke fagbiblioteket. Denne beslutningen ble tatt for å inkludere skoleoppgaver fra hele Norge, og ikke kun ved UiA. Spesifikasjonene som ble benyttet, for alle søkeordene var: masteroppgaver, ingen avgrensning på språk, siste fem år. Det viste seg at disse begrensningene, gjorde det mulig å unngå bruk av boolske operasjoner (som: og, eller, ikke).

Benyttede søkeord med antall treff, antall relevante treff og inkluderte masteroppgaver, er presentert i Tabell 6.2, til formål to. Utformingen av tabellen er hentet fra [79]. Treff som ble potensielt relevante, var oppgaver som omhandlet brukonstruksjoner. Alle andre konstruksjonstyper og vinklinger, ble ikke ansett som relevante. Til tross for at det ikke var satt begrensning til språkvalg, ble kun innhold på norsk og engelsk inkludert. Denne begrensningen ble tatt, for å ha mulighet til å forstå innholdet, uten total oversettelse.

Tabell 6.2: Søk, søkeord, antall treff, potensielle treff og inkluderte masteroppgaver

Søk	Søkeord	Treff	Pot. Relevant	Inkludert
1	«Parametrisk design»	5	3	2019 - Optimalisering av nettverksbru «Parametrisk design som optimaliseringsverktøy»  2018 - Bruprosjektering med bruk av parametrisk design «en arbeidsflyt mellom Dynamo-Revit-SOFiSTiK»  2017 - Konseptuell design av Gangbruer- parametrisk modellering som verktøy i tidlig designfase
2	«Parametrisk modellering»	2	1	2017 - Konseptuell design av Gangbruer- parametrisk modellering som verktøy i tidlig designfase
3	«Parametrisk prosjektering»	0		
4	“Parametric design”	9	2	2019 – Master - Optimalisering av nettverksbrubue «parametrisk design som optimaliseringsverktøy»
5	"Parametric modelling"	5	2	2017 – Master - Conceptual structural design of footbridges – A case study of asymmetrically suspended bridges using parametric modelling
6	“Parametric engineering”	0		
Totalt inkludert:				4

## 6.2 Innhentet erfaringer fra mekatronikkingeniøren

I denne metoden, ble det satt søkelys på en av grenseindustriene til byggingeniøren, nemlig mekatronikkingeniøren. Hensikten her, var å innhente erfaringer og arbeidsflyt, benyttet til produksjon av CAD- og FEM-modeller. Til dette formålet ble det gjennomført et møte med én mekatronikkingeniør, som er ansatt ved UiA i Grimstad. Valg av kontaktpersoner, ble tatt i samarbeid med internveilederen.

Det ble besluttet at Jon Lunden Hustad skulle kontaktes. Avgjørelsen ble tatt på grunnlag av at vedkommende underviser mekatronikkstudenter, i CAD og FEM. Tidligere har han arbeidet som mekanisk ingeniør i MHWirth og MacGregor. Programvarene som han har god kjennskap til, er Abaqus, Ansys (FEM-programmer) og Solidworks, PTC Creo (CAD programmer).

### 6.3 Anvendte programvarer og opplæringsprosessen

For gjennomføring av parametriske modellering, var det nødvendig å benytte et utvalg av programvarer. Hvilke programvarer som ble anvendt, er presentert i Tabell 6.3. I tabellen er versjonsnummeret, siste nedlastingsdato, og type lisens for hver programvare inkludert. Siste nedlastingsdato er av relevans, da programvarene er i stadig utvikling og fornyelse. I tillegg, er type lisens presentert for å presisere hvilke programvarer, som krever kjøp av lisenser. I tabellen viser det til at det var nødvendig med lisens, til bruk av Rvt, Rhino og Fd.

Tabell 6.3: Programvarer benyttet, samt versjonstype, nedlastingsdato og lisenstype

Programvare	Versjonsnummer	Siste nedlastingsdato	Lisenstype
Autodesk Revit 2021	21.0.0.383	07.09.2020	Studentlisens
Rhino 7	7.3.21053.23031	04.03.2021	Zoo lisens fra UiA
Rhino.Inside for Revit	0.1.7747.22602	24.03.2021	-
Grasshopper	1.0.0007	04.03.2021	-
FEM-Design 20	20.0.1	04.03.2021	Nettlisens DS

Programvarene Rvt og Fd ble valgt, fordi bedriften DS benytter disse til å utarbeide tegnings- og beregningsprodukter. Mens valg av det parametriske programmet GH, ble tatt i henhold til tre påvirkningsfaktorer.

Den første faktoren kom av opparbeidet erfaring fra forprosjektet [3]. Dette innebar å ha mulighet til å få hjelp med å benytte programvaren Dynamo, hvilket ikke var tilfelle i henhold til kontaktpersonene som ble brukt på forprosjektet. Ingen av dem hadde erfaring innenfor denne programvaren. For GH programmet, kunne derimot to av kontaktpersonene bistå med hjelp. Det kom av erfaringer de selv hadde, med å benytte GH til arkitektrelaterte prosjekt.

Den andre årsaken til valg av GH, var styrt av lisenstilgang. I forprosjektet ble Dynamo utelukkende valgt, på grunnlag av at DS allerede hadde lisensen til dette verktøyet. Men i 2021, ble det kjent at UiA hadde fått tilgang på en lisens for Rhino 7, som kunne benyttes for studenter. Dette åpnet opp for muligheten til å benytte GH.

Siden begge programmene har tilgang på API-et til Rvt og Fd, ble den avgjørende faktoren for valg av GH, tatt i henhold til informasjonen som er presentert i kapittel 3.4.1, samt resultatene funnet i litteratursøk formål to. I kapittel 3.4.1 fremkommer det at det flere fordeler knyttet til bruk av GH.

### 6.3.1 Opplæring

Fd og Rvt var allerede kjente programvarer fra tidligere prosjekter, hos DS. Derfor ble det kun gjennomført opplæring av GH, RiR, samt verktøykassen til Fd. Denne opplæringsprosessen fant sted før selve gjennomføringen av den parametriske prosjekteringen, hvor opplæringen ble styrt i henhold til prosjekteringselementene. Disse elementene blir beskrevet i kapittel 6.4.2. Til opplæringen ble hovedsakelig de digitale manualene benyttet.

#### **Grasshopper**

Til tross for at det er flere likheter med brukergrensesnittet til Dynamo og GH, var det nødvendig med en opplæring her. Dette kom av at GH kjøres via programvaren Rhino, som ikke var brukt tidligere. Det ble startet med å ta i bruk en visuell innføring av brukergrensesnittet, da dette ble ansett som en rask og effektiv metode i forprosjektet. Til dette ble Youtube-kanalen «Rhino Grasshopper», med filmen «Rhino Grasshopper Tutorial (Beginner)» benyttet [45]. Denne kanalen hadde per 10.04.2021, 48,6 tusen abonnementer [80].

Lengre ut i prosjektet, ble det nødvendig med en opplæring av å ta i bruk Python i GH, for å få tilgang til spesifikke Rvt-API. Dette ble nødvendig, der det var mangel på ferdigkomponerte komponenter. Opplæring for Python i GH, tok i bruk nettsidene [81] og [82]. Her ble det satt søkelys på ordbruk i Python. For særskilt utarbeidelse av Python koder for Rvt API-et, ble nettsiden [83] benyttet, hvor det ble det innhentet API-er til metoder og elementer.

#### **Rhino.Inside.Revit**

Opplæringsprosessen for RiR ble betydelig mer omfattende, sammenliknet med GH i seg selv og verktøyene for Fd. Det kom av at oppbygningen av Rvt-elementer opplevdes mer kompleks, sammenliknet med oppbygningen av Fd-elementer. Til opplæringen ble primært to hjelpemidler tatt i bruk: hjemmesiden til RiR [84], og ett «Crash Course in Rhino» fra Youtube-kanalen til Aussie BIM Guru [85]. Hvordan disse ble tatt i bruk blir beskrevet videre.

Hjemmesiden ble tatt i bruk for å finne ut hvordan GH og Rvt kunne benyttes i en arbeidsflyt. Her ble fanene «Get Started» og «Guides» (manualen) bruk. De relevante kapitlene i manualen ble gjennomgått, i henhold til utvalget av prosjekteringselementene som fremkommer i kapittel 6.4.2. Manualen inneholder eksempler, som var gjennomførbare. Her var det enkelte Rvt-komponenter som var utdatert, i henhold til versjonen benyttet i dette prosjektet. For disse, ble det da funnet tilsvarende komponenter, i egen versjon. Dette var en god lærestrategi, fremfor kun gjennomlesning. Den samme erfaringen ble gjort i forprosjektet, ved opplæring av programvaren Dynamo.

Det viste seg at manualen ikke var ferdigutviklet. Enkelte temaer hadde begrenset innhold, eller ingen innhold i det hele tatt. Hvilke deler som ble gjennomgått, samt temaer som var av ønske å gjennomgå, men med manglende innhold, er presentert i Tabell 6.4. Her indikerer sirkler, kapitler med tilstrekkelig innhold for prosjekteringselementene. Kryssene derimot, indikerer kapitler som ikke inneholdt informasjon. Mens kombinasjonen av sirkel og kryss, var kapitler som hadde noe tilgjengelig innhold.

*Tabell 6.4: Kapitlene i "Guide" veilederen som ble gjennomgått*

Levels og Grids	x	Structural elements	o
Project settings	o	Materials-gjennomført	o
Styles and patterns	o	Adaptive components	x
Dimentions	x	Modify geometry	o
View	o	Custum families	o x
View templates	x	Tags and keynotes	x
Sheets	x	Drafting	o x
Revisions	x	Python components in Revit	o

Etter gjennomgangen av manualen, ble serien av videoene fra «Crash Course in Rhino» sett. Dette ble gjort fordi det var av nødvendighet å repetere bruken av Rvt-komponentene, mer visuelt. Her ble videoene, på lik linje med kapitlene i manualen, valgt i henhold til relevans for casen. De gjennomgåtte videoer er presentert i Tabell 6.5 [85], og disse tilhører Youtube-kanalen til Aussi BIM Guru. Aussi har en bakgrunn som arkitekt, og har arbeidet tett opptil ingeniører og andre AEC industrier. Kanalen ble opprettet 10. mars 2019 og hadde allerede 16.3 tusen følger [86]. I henhold til dette, ble kanalen ansett som en anerkjent ressurs for opplæring.

Tabell 6.5: Videoene gjennomgått på Youtube-kanalen til Aussi BIM Guru

Navn på video	Publiseringsdato
Rhino inside: Revit elements into Rhino	27.10.2019
Rhino inside: Get Revit parameters	30.10.2019
Rhino inside: Set Revit parameters	03.11.2019
Rhino inside: Create Revit Grids	24.11.2019
Rhino inside: Create Revit levels	27.11.2019
Adding Revit data to Rhino models using Rhino inside!	29.11.2020

Verken manualen eller videoene, inneholdt informasjon om modellering av 3D armering med Rvt verktøyene i GH. Derfor ble det spesifikt søkt etter dette på Youtube og på RiR forumet<sup>5</sup> (forumet er å finne på hjemmesiden deres). I søkemotoren Youtube, ble søkeordene «Rebar an element in Revit with grasshopper» brukt. Den 18.03.2021 fikk dette søket tre relevante treff. To filmer fra Structures Parametric og en av Bulent Karakus. Disse presenterte metoder som innebar egendefinerte Python-komponenter. For å få tilgang på innholdet av disse komponentene, måtte de kjøpes.

Målet var i utgangspunktet å finne metoder som tok i bruk ferdigkomponerte komponenter, da kunnskapen om Python var tilnærmet lik null. Fortsatte derfor søkeprosessen på forumsiden, med søkeordet «Rebar». Dette søket fikk 14 treff den 18.03.2021. De tre første treffene var av relevans, fordi de presiserte problemstillinger som omhandlet produksjon av armering i henhold til kurver. Det relevante innholdet fra disse treffene, er presentert i Tabell 6.6. I tabellen presenteres navnet på hvem som har publisert problemstillingen, når det ble publisert, metoden det henvises til, problemet

<sup>5</sup> Dette er en plattform hvor brukere av programvaren, tidligere har stilt spørsmål, og fått eventuelt svar av andre brukere.

med metoden, samt om skribenten vedla filen. Dette ble eneste informative kilde til opplæring av produksjon av armering, med Rvt-verktøy i GH.

Tabell 6.6: Relevant kunnskap hentet fra forumet til RiR for opplæring av produksjon av armering

Skribent	Tidspunkt	Metode	Problem	Tilgang på fil
Jorgen Sorbol	Juli 2020	Egenkomponert Python-komponent som generer armeringsstenger med «CreateFromCurves» metoden i Revit API-et	Feilmelding i komponenten	Ja
Odtj	Oktober 2020	Armeringsjernene modellert med komponenten «AddDirectShape.ByGeometry»	Inkluderer ikke formkode (00, 11, 21) eller armeringstype (ø8, ø12...)	Nei
Lindgren Eirik	Januar 2020	Armeringsjernene modellert med komponenten «AddDirectShape.ByGeometry». få	Inkluderer ikke formkode (00, 11, 21) eller armeringstype (ø8, ø12...)	Ja

### FEM-Design verktøykasse

For bruk av Fd verktøyene i GH, var det få tilgjengelige nettressurser. Både for opplæring av konsept, og virkemåte for verktøyene. Ett av kapitlene i Fd manualen, inneholdt en beskrivelse av å ta i bruk Fd API-et i GH [87]. Her ble de ulike komponentene presentert, i henhold til produksjon av elementer i Fd. For en nybegynner, var dette alt for kortfattet og lite beskrivende til å kunne starte skriptproduksjonen på egenhånd. I tillegg var eksempelfilene, som var av ønske å få gjennomgått, verken mulig å åpne eller laste ned.

Manualen kunne derfor ikke benyttes til å få en forståelse av hvordan komponentene benyttes, til å utarbeide en datastrøm. Dette ble imidlertid funnet i videoen [88], på Youtube. Denne gav et innblikk i hvordan komponenter fra verktøykassen til Fd, kunne utarbeides for å produsere Fd produkter. Ut over dette var det begrenset med nettbaserte hjelpemidler, for beskrivelser av hvordan benytte Fd API-et i GH.

## 6.4 Parametrisk prosjektering

Målet med parametrisk prosjektering, var å utarbeide innhold i Rvt og Fd, ved å benytte «ferdigkomponerte komponenter»<sup>6</sup> i GH. Denne arbeidsflyten er presentert i Figur 6.2. Dette ble gjort for å belyse potensialet GH har til å produsere irregulære former, samt undersøke muligheten til å generere en effektivisert revisjonsprosess. Der hvor denne arbeidsflyten ikke var oppnåelig, ble produksjonene fullført i den relevante programvaren. Dette ble gjennomført for å oppnå tilstrekkelig innhold, for produktene tegnings- og beregningsprodukter. Dette tillegget i arbeidsflyten, ble utført

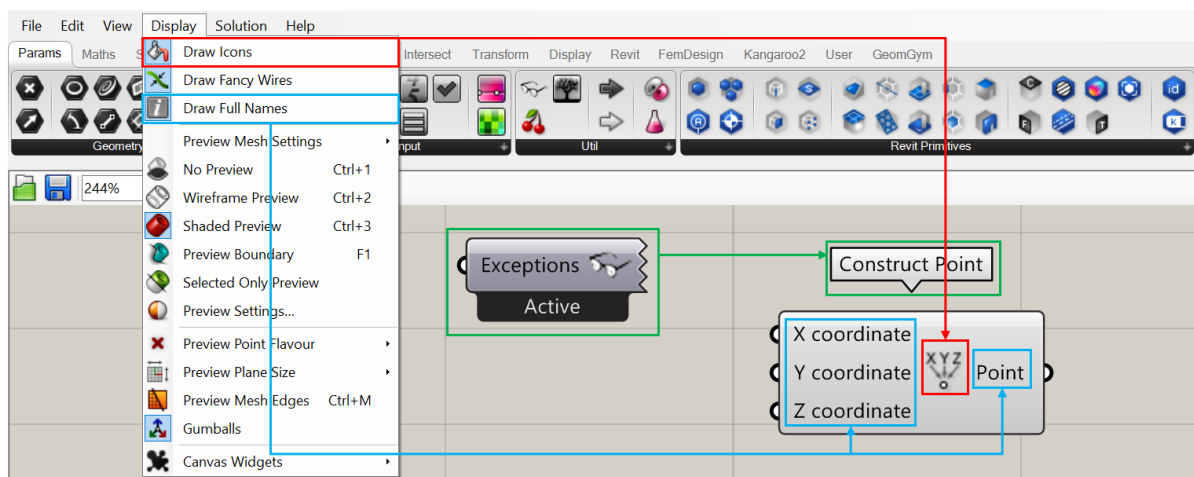
<sup>6</sup> Begrepet «Ferdigkomponerte komponenter» i denne rapporten, innebærer alle verktøy som ikke krever tekstlig koding (slik som Python, C## eller liknende).

for å kunne undersøke hvordan revisjonene ble gjennomførbare. Hvilke prosjekteringselementer dette gjaldt, presenteres i resultatkapittelet.



Figur 6.2: Informasjonsflyten/arbeidsflyten mellom GH, Rvt og Fd

Før den parametriske prosjekteringen startet, ble det bestemt at alle komponentene i GH skulle ha samme fremvisning, slik som komponenten presentert i Figur 6.3. Fremvisningen ble definert i fanen «Display». Her ble det bestemt at komponentenes ikon skulle brukes (rødt), fremfor navnet i form av tekst. Videre ble inngangs- og utgangs typene presentert med fullt navn (blått), fremfor forkortelser. For å få optimal informasjon om hver komponent, ble også komponenten «Bifoscale» tatt i bruk. Denne ble lastet ned som en Plug-In i GH [89], og ble brukt for å navnsette alle komponentene i arbeidsområdet til GH (grønt). Denne navnsettingen var uavhengig av innstillingene i «Display» fanen.



Figur 6.3: Fremvisning av komponentene i GH

For utarbeidelse av parametriske modeller i Rvt og Fd, ble casen for brudekket i kapittel 5.3, benyttet som grunnlag. Dette innebar RIB tegningene og beregningsrapporten til Sagåne bru, hvorav lastplasseringen av vindlast her ble erstattet med utførelsen på Birkelid bru. Dette var informasjon som ikke var vedlagt i beregningsrapporten som ble sent av Rambøll. I tillegg ble lastkombinasjoner og lastfaktorer fra Sagåne bru, erstattet med v668, for å oppnå en mer universell og standardisert modell.

Etter samtale med prosjekteringsansvarlig på prosjektet Sagånde bru, fremkom det at Rambøll selv hadde modellert denne i CAD-programmet Tekla, med å benytte GH. Rambøll kunne ikke dele innholdet i GH skriptet, begrunnet med at de fremdeles anser dette som et konkurransefortrinn i ingeniørbransjen. Hvilke elementer som her var produsert i GH, var dermed uvisst. Det innebar at alt

arbeid som presenteres videre, ble gjennomført uten hjelp fra eksisterende gh.fil fra prosjektet Sagåne bru.

For gjennomføringen av parametriske prosjektering av tegnings- og beregningsprodukter, ble flere personer kontaktet for å bistå med råd og veiledning. Spørsmål knyttet til prosjektering og modellering i hver av programvarene (ikke med GH), ble stilt til følgende ansatte ved DS:

- BIM koordinator: for hjelp med Rvt.
- Thomas, for hjelp med Fd. Thomas bistod spesifikt med hjelp med plassering og tilnærming av laster på konstruksjonen.
- Martin, Erik og Robert for hjelp med å definere brukonstruksjonens grensesnitt.

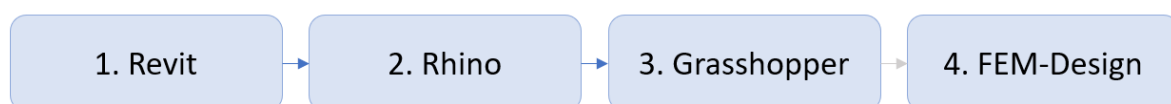
For spesifikk hjelp med programvaren GH, bistod Renata Carolina Ribeiro Portelada fra bedriften Trollvegg med svar på problemene. Her ble det avholdt ett møte, hvor problemstillingene som ikke ble besvart med hjelpemidlene presentert i kapittel 6.3.1, ble besvart av henne.

Når det videre refereres til komponenter, innganger og utganger, vil disse konsekvent presentere med følgende spesifikasjoner.

- Komponentene: faktiske navn (engelsk) inni anførselstegn
- Inngangene og utgangene: Kursiv tekst, med stor forbokstav

### Arbeidsmetodikk

For gjennomføring av parametriske prosjektering av brudekker, var det nødvendig å åpne programvarene i en spesifikk rekkefølge, for å kunne utarbeide skript i GH som produserte elementer i Rvt og Fd. Rekkefølgen er presentert i Figur 6.4, hvor Rvt ble åpnet først, før Rhino ble åpnet fra Rvt. Denne rekkefølgen ble brukt for å få tilgang til Rvt verktøyene i Rhino. Deretter ble GH åpnet i Rhino, fordi Rhino ble benyttet som et hjelpemiddel til å produsere skriptene. Det siste programmet som ble åpnet, var Fd. Dette programmet var ikke avhengig av rekkefølgen oppstrøms, men måtte være åpent for å få data i GH til å kjøre i Fd (årsaken til grå pil i Figur 6.4).

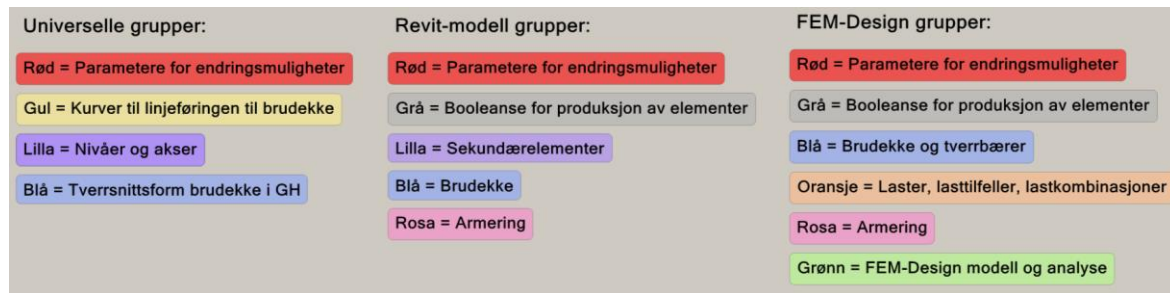


Figur 6.4: rekkefølgen på hvordan programvarene må åpnes

Dermed var det i alt fire programvarer som var åpne samtidig, for utarbeidelsen av GH-skriptene. Det var her viktig at enhetene i Rvt, Rhino og Fd var like, for at elementene skulle få riktig forhold. Enheten ble satt til meter. Grunnen til at ikke ble satt til millimeter, som er standard for RIB tegninger, fremkommer i resultatene.

Både skript for produksjon av Rvt- og Fd modeller, ble utarbeidet i samme GH-fil. Filen inneholdt dermed store mengder komponenter. Derfor ble komponentene tildelt ulike fargegrupper, hvor fargene ble brukt til å indikere funksjonen de hadde. En oversikt over funksjonene til fargegruppene, er presentert i Figur 6.5. Det var av hensikt å arbeide i GH med fargegrupper for å holde oversikt over innhold, samt gi mulighet til kvalitetskontroll og intuitivt avdekke eventuelle feil.





Figur 6.5: Grupperingen av gruppene i GH

Det fremkommer i figuren at grupperingene var inndelt i tre hovedkategorier, henholdsvis universelt, Rvt-modell og Fd-modell. Universelle grupper, var skript som ble brukt for både produksjon av Rvt- og Fd-modeller. Hvordan disse tre hovedkategoriene ble utarbeidet i GH, blir beskrevet i kapitlene 6.4.3-6.4.5.

#### 6.4.1 Valg av variable parametere

Målet for valg av variable parametere, var at disse kunne benyttes til både Rvt- og Fd modellen, for å gi effektive endringsmuligheter. Variable parametere innebærer i denne sammenheng, parametere som skulle benyttes som grunnlaget for revisjonsmuligheter på prosjekteringselementene. Hvilke parametere dette skulle gjelde, ble bestemt i samarbeid med prosjekteringsansvarlig på Sagåne bru, samt DS konsulentene Martin Malo, Robert Stenersen McShane og Svein Erik Marum.

Det ble besluttet at både tverrsnittprofilen og lengden til brudekket, var av nytteverdi å utarbeide som variable parametere i GH. Disse parameterne er hensiktsmessig å holde variable, fordi de er vanskelige å forutse for en konsulent, som ikke har flere års erfaring med bruprosjektering.

Utover disse to variable parameterne, ble det også bestemt at tverrfall, horisontal- og vertikalkurve, samt høydeforskjell skulle utarbeides som variable parametere. Utenom tverrfall, er dette parametere som definerer veilinjen til brudekker. Disse er nyttige å holde variable, for eventuelle endringer som er gjort av byggherre. I prosjektet Sagåne bru, ble disse parameterne sendt fra SVV som utgangspunkt for prosjekteringen.

I henhold til egne erfaringer med bruprosjektering, etablert ved gjennomføringen av bacheloroppgaven 2019, ble en ekstra variable parameterer integrert. Her var det kjent at bredden til et brudekke blant annet avhenger av antall kjørebener den skal prosjekteres for. Dette var et aspekt som skulle kobles opp imot definisjonen på bredden av brudekket. Faktorene som definerer antall kjørebener, ble neglisjert i skriptet. Det kom av at denne betraktningen kan tas av ingeniøren utenom skriptet. Dette innebærer at faktorene ÅDT, fartsgrense, og dimensjoneringsklassen ikke ble tatt med. Samme begrensning ble satt for utformingskravene for minste tverrfall [90, s.30] og minste horisontalkurveradius og vertikalkurveradius [90, s.32].

Armerings-parametere som er mulig å definere i programvarene Rvt og Fd, dannet grunnlaget for valg av variable parametere til armeringen. Her ble det bestemt å ta utgangspunkt i nominell overdekning, da dette avhenger av eksponeringsklassen til betongen [91, Tabell 7.2]. I tillegg skulle parameterne senteravstand, diameterstørrelse på stengene, forankring og minimumskravet for armering utarbeides som variable parametere. Minimumskravet omhandler senteravstanden til

armering, som en ferdigutregnet verdi. Dette er parametere som er dimensjonerende for prosjektering av armering.

#### 6.4.2 Valg av prosjekteringsselementer

Prosjekteringsselementene som skulle modelleres, var det neste som ble bestemt før selve gjennomføringen av den parametriske prosjekteringen. Dette var elementer i både Rvt og Fd. Hensikten med å spesifisere prosjekteringsselementene, kom av at disse skulle benyttes som utgangspunkt, for å finne riktige komponenter i GH i henhold til produksjon og håndtering.

Prosjekteringsselementene ble klassifisert i to grupper, som primær- og sekundærelementer. Her inngår armering og brudekket i primærelementene, mens resterende prosjekteringsselementer defineres som sekundærelementer. Valg av sekundærelementer ble tatt i henhold til produktene som produseres hos DS, som allerede er presentert i casekapittelet 5.2.1. Hvilke sekundærelementer dette omhandler i Rvt og Fd, fremkommer videre i dette delkapittelet.

Her er det av hensikt å presisere at primærelementene ble undersøkt i flere etapper, for å finne forbedringspotensialer. For sekundærelementene derimot, ble det utarbeidet én metode som ble benyttet gjennomgående i alle undersøkelsene av primærelementene. Grunnen til dette var at brudekket og armering var hovedfokuset i dette prosjekt, da de kan inneholde irregulære former.

I teorikapittelet ble det presentert at datastrømmen i GH flyter fra venstre mot høyre. I utarbeidelsen av skript i GH, ble det derimot arbeidet med en arbeidsmetodikk motsatt vei, altså fra høyre til venstre. Dette innebar, å finne komponenter til produksjon av de ulike prosjekteringsselementene. Disse ble benyttet som utgangspunkt, hvor inngangsdata til hver av dem, ble utarbeidet nedstrøms i skriptet. Hvilke komponenter som dannet utgangspunktene for prosjekteringsselementene, og logikken som ble tatt i bruk for inngangsdata til hver av komponentene, blir beskrevet i kapitlene 6.4.3-6.4.5.

I henhold til de utvalgte prosjekteringsselementene i Rvt og Fd, vil de engelske navnene videre omtales med en de norske oversettelsene fra Tabell 6.7 . Dette er både oversettelse av prosjekteringsselementene i hver programvare, samt andre definisjoner til de tilhørende programmene.

Tabell 6.7: Norske definisjoner på ordinære Rvt og Fd definisjoner

Revit	Norsk betydning	FEM-Design	Norsk betydning
Category	Kategori	Beam	Bjelke-element
Family	Familie	Point support group	Punktopplegg gruppe
Type	Type	Story	Nivå
Instance	Forekomst	Axis	Akser
View	Visning	Loads	Laster
Sheet	Ark	Loads cases	Lasttilfeller
Section	Snitt	Duration class	Varighetsklasse
Plan view	Planvisning	Load combinations	Lastkombinasjoner

Level	Nivå	Point load	Punktlast
Grid	Akse	Line load	Linjelast
Dimentions	Målsetting	Line temperatur variation load	Linje temperaturvariasjonslast
Rebar	Armering	Calculate	Beregne
Beam	Bjelke	RC design	Prosjektere armert betong
Component	Komponent		

## Revit

I henhold til casen, ble et utvalg av Rvt-elementer valgt ut. Disse er listet opp i Tabell 6.8, hvor elementene er presentert med tilhørende definisjon i Rvt-hierarkiet. Her er armeringsjern og betongbjelke primærelementene, mens resterende innhold i tabellen er sekundærelementene.

Tabell 6.8: Rvt-elementene for parametrisk prosjektering med tilhørende objektstype beskrivelse

Revit-element	Objekt type	Kategori	Familie	Type
Armeringsjern «Freeform»	Autodesk.Revit.DB.Structure.Rebar	Structural Rebar	Rebar bar	Ø-størrelse
Betong bjelke	Autodesk.Revit.DB.FamilyInstance	Structural Framing	Brutverrsnitt original	Brutverrsnitt original
Målsetting	Autodesk.Rebar.DB.Dimension	Dimensions	Linear Dimension style	Utforminstype
Snitt	Autodesk.Rebar.DB.Element	Views	Section	Snit
Level	Autodesk.Rebar.DB.Level	Levels	Level	Utformingstype
Akse	Autodesk.Rebar.DB.Grid	Grids	Grid	utformingstype
Ark	Autodesk.Rebar.DB.Viewsheet	Sheets	Sheet	Sheet
Visning	Autodesk.Rebar.DB.Viewplan	Views	Floor plan	Floor plan

Anskaffet denne informasjonen, for å ha kontroll på hvordan de skulle behandles i GH. Informasjonen ble hentet fra GH-komponenten «Element», hvor det ble tatt utgangspunkt i elementers id, hentet i Rvt. Tok i bruk metoden presenter i kapittel 5.1.4, hvor ID-nummeret til hvert av elementene i tabellen ble kopiert inn i GH-dialogen til denne komponenten. Filen som inneholder denne GH-informasjonen, samt elementene som ble benyttet i Rvt til dette formålet, er vedlagt i Vedlegg F.

Informasjon innhentet i denne komponenten, ble spesifikt benyttet til å finne ut hvilke kategorier og typer som skulle defineres, for å produsere Rvt-elementer med GH. Innholdet ble også benyttet for å forstå hvordan produksjonen av Python-komponenter kunne utarbeides, der hvor ferdigkomponerte komponenter ikke var tilstrekkelig.

## FEM-Design

Valg av prosjekteringselementer i Fd, ble også tatt i henhold til casen. Her var utvelgelsen mindre kompleks, ved at utvelgelsen av elementene tok i bruk arbeidsflyten som fremkommer i rekkefølgen

på fanemenyene. Her ble elementene listet opp under, valgt. Bjelke og armering er primærelementene, mens resterende innhold i oppramsingen er definert som sekundærelementene.

1. Bjelke
2. Opplegg; punkt
3. Laster; punkt og linje
4. Lasttilfeller; type og varighet
5. Lastkombinasjoner; type og lastfaktor
6. Beregning
7. RC beregning
8. armering; auto- og manuell gjennomføring

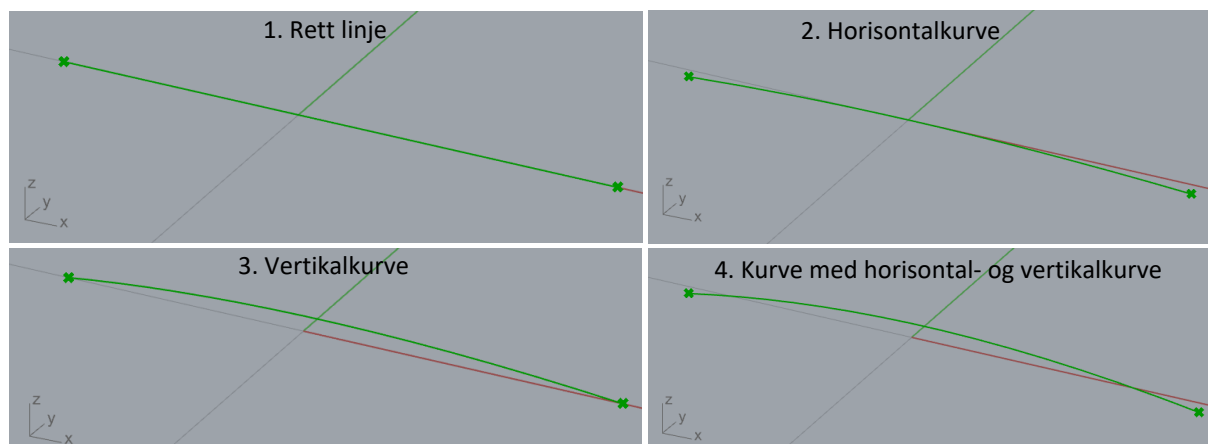
### 6.4.3 Universelle grupper

Det ble utarbeidet et sett med universelle grupper, som ble benyttet som referansegeometri til produksjon av både Rvt- og Fd modeller. Dette var geometri som omhandlet kurver som skulle definere veilinjen til brudekket, nivåene og aksene i modellene, samt tverrsnittsformen på brudekket. Kurvene ble modellert som geometri i GH, og fremkom i Rhino fortløpende. Hvilken fremgangsmetode som ble benyttet for produksjonen av disse, vil bli beskrevet her.

#### Referansegeometri

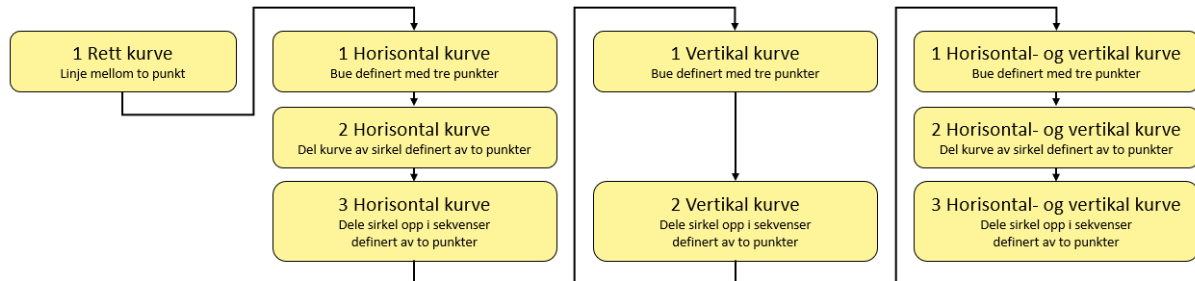
Veilinjene ble definert som kurver, i henhold til fire ulike kriterier. Dette ble gjennomført, for å gi valgmuligheter til utforming av brudekkets veilinje. Disse ble utarbeidet, ved å produsere en rett linje, definert av brudekkes lengde, én bue definert av horisontalkurveradius, én bue definert av vertikalkurveradius, samt én bue som var definert av både horisontal- og vertikalkurveradius. I hvilken del av rommet, disse ble utarbeidet, er listet opp under, og illustreres som eksempler i Figur 6.6.

1. Rett linje – utarbeidet med x-retning
2. Horisontalkurve – utarbeidet i x- og y-retning
3. Vertikalkurve – utarbeidet i y- og z-retning
4. Kurve med horisontal- og vertikalkurve – utarbeidet i x-, y-, og z-retning



Figur 6.6: Visuell fremvisning av utformingsmulighetene til brudekkets veilinje

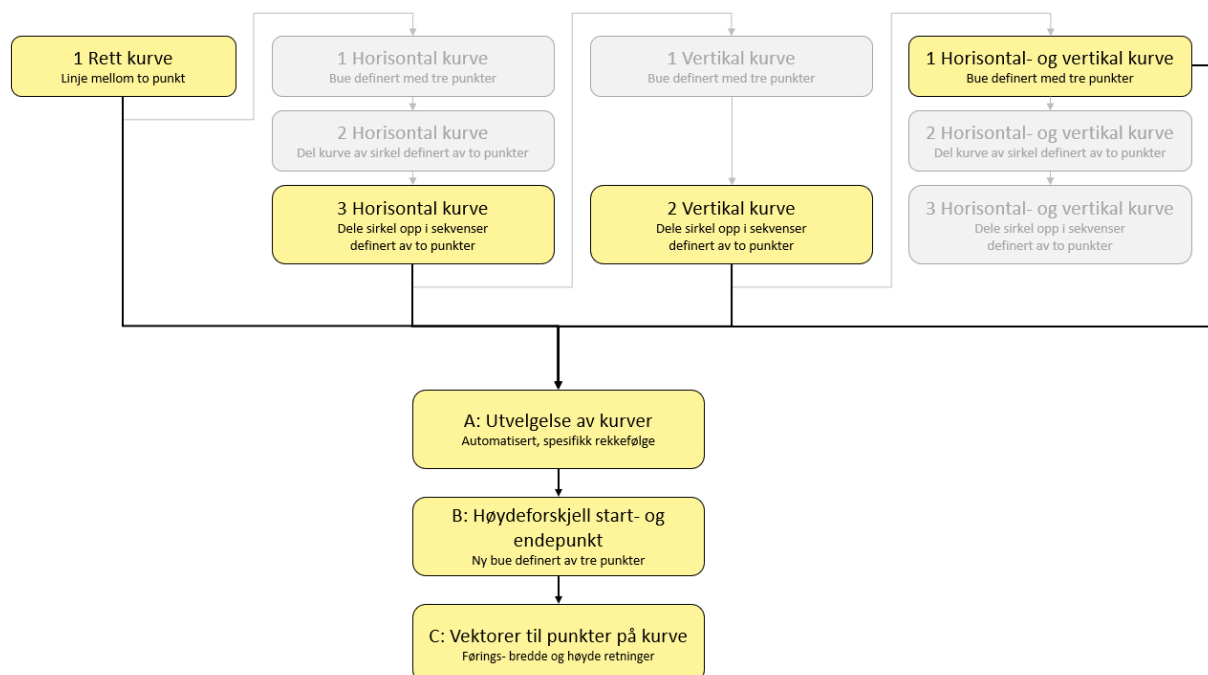
Utarbeidelsen av de fire kurvene, ble utformet i rekkefølgen som er illustrert i Figur 6.7. Det ble først utarbeidet en rett linje, som var definert som en linje mellom to punkter. Deretter ble det modellert tre horisontalkurver, to vertikalkurver og tre kurver med horisontal- og vertikalkrumning. Årsaken til at flere metoder ble benyttet, ble utført for å finne ut hvilke metoder som gav mest nøyaktige kurver (i henhold til sirkler som referansegeometri).



Figur 6.7: Arbeidsflyt kurvene som definerer brudekkets føringslinjen

Start og endepunkt til alle kurvene, ble symmetrisk plassert om origo, både i x- og y-retning. De ble videre definert med ha lik start- og endeparameter, hvorav startparameter hadde verdien 0 og endeparameteren verdien 1. Her ble positiv x-akse, definisjonen på startpunktet, mens negativ x-akse, ble definisjonen på endepunktet av brudekket. Dette ble gjennomført, for å ha kontroll på kurvenes oppførsel, som viste seg å være hensiktsmessig til senere bruk, i form av inngangsparametere.

De mest nøyaktige metodene, for produksjon av kurve-geometriene, ble tatt med for videre utvelgelse. Utvelgelsen fanget opp kurven, etter hvilke variable parametere som var gjeldende for veillinjen. Hvilke kurver fra Figur 6.7 som ble tatt med i denne utvelgelsen, er presentert som (A) under. Her fremkommer også videre arbeid som ble gjennomført i henhold til utvalgt kurve, (B) og (C). Skriptet som er utarbeidet til dette, er vedlagt i Vedlegg F.



Figur 6.8: Logikken bak produksjon for gjeldene kurve

Utvelgelsen (A) ble fastsatt ved at de 4 kurvene i Figur 6.8, ble sortert med en automatisert utvelgelse. Får å få en riktig utvelgelse, i henhold til parameterne som var gyldige for definisjonen av veilinjen, ble kurvene systematisk valgt ut ifra rekkefølgen de ble listet opp i. Her er rekkefølgen på opplistingen, presentert punktvis under. Øverste gjeldende kurve i listen, ble hentet ut, og avhenger av gjeldende parametere til produksjon av veilinjen.

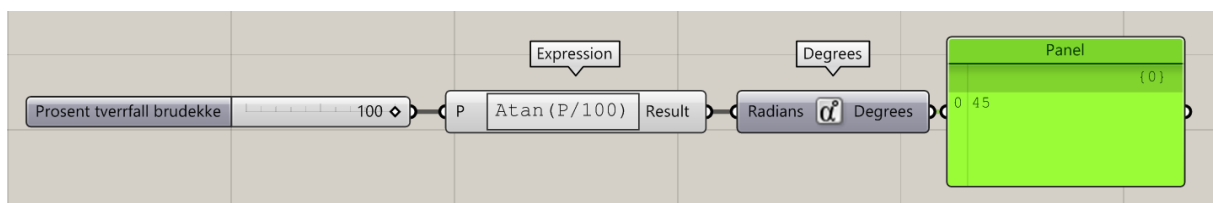
1. Kurve med horisontal og vertikal krumning
2. horisontalkurve
3. vertikalkurve
4. Rett kurve

Etter at «gjeldende kurve» var valgt, ble denne tilført muligheten for å inneholde høydeforskjell mellom start og endepunkt (B). Her ble det tatt utgangspunkt i symmetrisk høydeforskjell uten endring av plassering i xy-planet. Det ble etablert en ny bue, som var avhengig av start-, midt- og endepunkt (A). Tok i bruk z-verdien til start- og midtpunkt fra (A), hvor disse ble addert med en potensiell høydeforskjell-verdi. Begge punktene ble benyttet for å opprettholde et symmetrisk brudekke om origo i lengderetning. Den kurven som her ble definert, blir videre referert til som «Gjeldende kurve». Det kommer av at denne kurven ble referansegeometrien for produksjon av Rvt- og Fd elementer.

For å ha referansegeometri i tverretningen av veilinjen, ble det utarbeidet en tre vektorer(C), som var avhengig av et definert antall punkter på «Gjeldende kurve». Dette var retninger langs veilinjen, bredden av brudekket og høyden av brudekket. De to sistnevnte ble definert ved å stå vinkelrett på retningen langs veilinjen. Dette ble gjort for å få riktig vinkel, samt kunne kontrollere tverrsnittene, som skulle definere produksjonen av brudekker i Rvt og Fd.

### Tverrfall

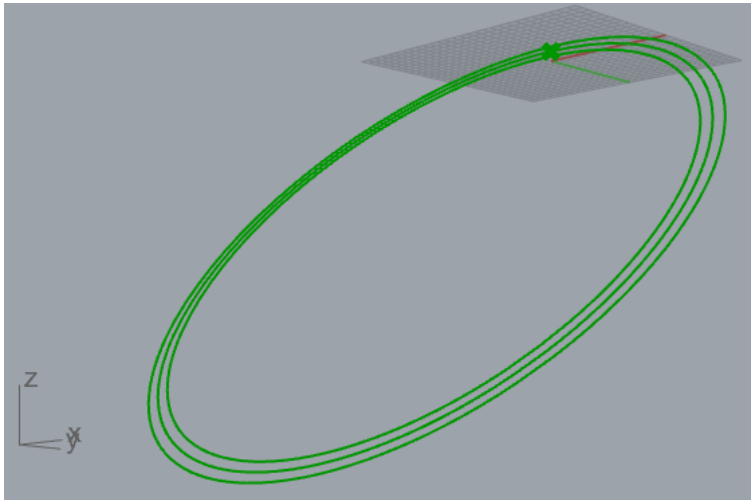
Tverrfallet ble utarbeidet ved å påføre rotasjon til brutverrsnittet. I casen ble tverrfallet beskrevet som prosentvis verdi. Denne verdien ble tatt i bruk, og omgjort til grader, ved at det ble utarbeidet en trigonometrisk formel for denne transformasjonen. Tok i bruk tangens invers med prosentverdi som motstående katet, og verdien hundre som hosliggende katet. For dette uttrykket, ble det utført en kontroll, for å verifisere at logikken var riktig. Prosentverdien ble derfor satt til hundre, da riktig svar her var kjent som 45 grader. Uttrykket bestod kontrollen (se Figur 6.9), og ble derfor benyttet til å påføre tverrfall på brudekkene.



Figur 6.9: Transformere enhet fra prosent til grader

For kurvene med vertikal og horisontalkurve, inneholdt disse et naturlig tverrfall; for linjer i planet til kurven. Dette fremkom da buen ble utvidet til en sirkel, for så å bli skalert. Skaleringen ble utarbeidet

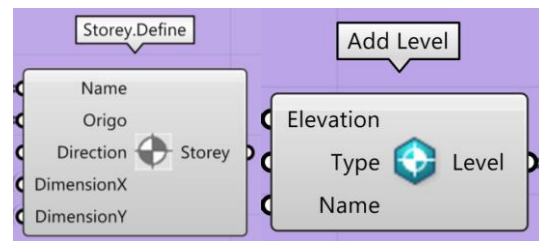
med en avstand på halve bredden av brudekket, på inn- og utside, som illustrert i Figur 6.10. Størrelsen på det naturlige tverrfallet ble regnet ut, og trukket fra i uttrykket i Figur 6.9. Dette ble gjort for at påført tverrfall skulle være kontrollerbart. Dette ble i tillegg gjennomført for at produksjon av elementer, i henhold til veilinjen, ikke skulle få tverrfall når det ikke var spesifisert.



Figur 6.10: Eksempel på når "Gjeldende kurve" får naturlig tverrfall

### Nivåer og akser

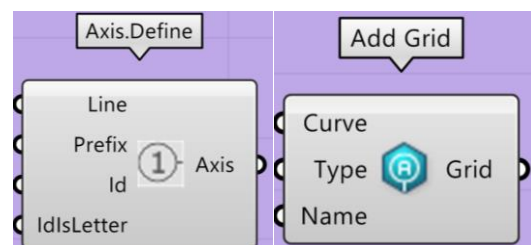
For modellering av nivåer i GH, ble «Add Level» benyttet for til produksjon i Rvt, mens «Storey.Define» ble benyttet til Fd. Disse ble tatt i bruk for å modellere parametriserte nivåer. Begge disse ble utarbeidet i henhold til linjer i det horisontale planet xy, da dette var begrensninger som begge komponentene krevde.



Figur 6.11: Komponenter for produksjon av nivåer i Rvt og Fd

For «Add Level» ble type utforming fra Rvt hentet inn, og koblet til inngangen *Type*. Videre ble navnene på hvert nivå utarbeidet parametrisert, og koblet inn i hver sin *Name*-inngang. For definisjonen av nivåer, var det nødvendig å utarbeide høydene, som nivåene skulle legges på. Høydene ble koblet inn i *Elevation* og i *Origo*. Inngangene *DimensionX* og *DimensionY* ble også definert i GH, for å spesifiser utstrekningen som nivåene i Fd skulle ha.

Også for modellering av akser i hvert av programmene, ble det benyttet hver sin komponent. For Rvt-akser ble «Add Grid» benyttet, mens «Axis.Define» for akser i Fd. Disse ble tatt i bruk for å modellere parametriserte vertikallakser. Det ble ikke modellert horisontallakser, da veilinjen til brudekket skulle benyttes for målsetting i lengderetning.



Figur 6.12: Komponenter for produksjon av akser i Rvt og Fd

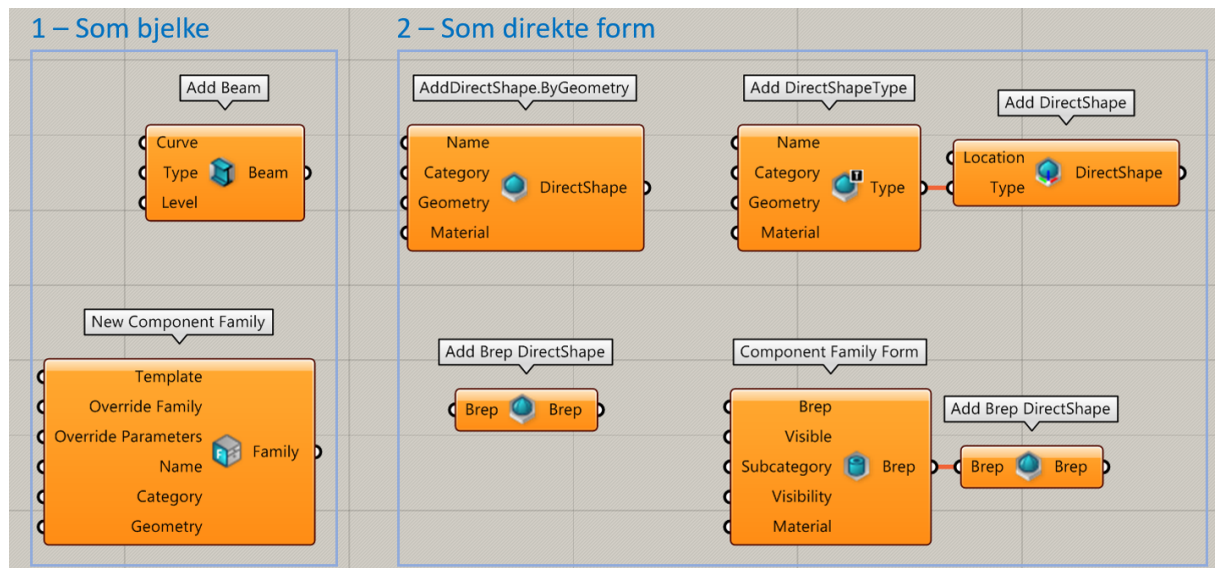
Vertikallaksene ble produsert i henhold til linjer i GH, i retning av brudekkets bredde. Dette var linjer som hadde x-, y-, og z variasjoner. Retningen på linjene ble flippet, for å få aksenumrene på riktig side av brudekket i xy planet. Data som definerte linjene, ble ført inn i inngangen *Curve* og *Line*. For

inngangene *Type* og *Name/Id*, ble disse utarbeidet med samme logikk; som for tilsvarende innganger hos nivåene.

#### 6.4.4 Revit modeller

##### Brudekker

Det ble tatt i bruk to ulike funksjoner, til produksjon av brudekker i Rvt, og fremlegges i Figur 6.13. De ble først utarbeidet som et bjelke-element (1), og deretter som direkte form (2).



Figur 6.13: Komponenter for produksjon av bjelke-element i Rvt

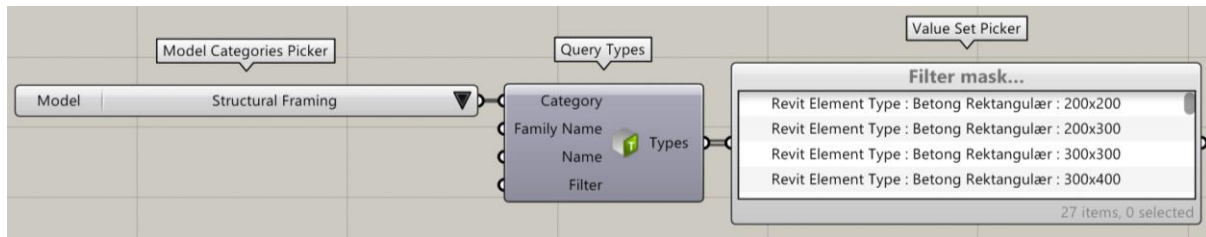
Produksjonen av et bjelke-element (1) med «Add Beam», definerte inngangen *Type*, med to metoder. Den første metoden, hentet inn en egendefinert familie, produsert i familie-malen «Metric Structural Framing - Beams and Braces». Familien ble tildelt forekomst-parametere, og filen er vedlagt i Vedlegg F. Den detaljerte beskrivelsen av hvordan familien ble utarbeidet, er vedlagt i Vedlegg E. For den andre metoden, ble komponenten «New Component Family» brukt. Den skulle utarbeide en egendefinert familie, i GH istedenfor i Rvt. For begge metodene ble «Gjeldende kurve» koblet inn i *Curve*, og Nivå 1 i *Level*. Nivået var hentet inn fra utgangsdataen til «Add level», presentert over.

Produksjon av elementer som direkte former, tok i bruk fire fremgangsmåter. Disse inkluderte ulike «DirectShape» komponenter, og hadde som hensikt å se hvilket globale parametere hver av dem gav resulterende Rvt-element. Her ble hele geometrien av brudekket produsert i GH, fordi komponenten «Sweep1» og «CapHoles» ble benyttet. «Sweepen» ble utformet i henhold til «Gjeldende kurve» og tverrsnittprofilen (plassert i startpunktet til «Gjeldende kurve»). De to sistnevnte komponentene hadde som hensikt å produsere et solid brudekke. Denne geometrien ble koblet inn i inngangene *Geometry/Brep*, til alle fire fremgangsmåtene.

For innhenting av den egendefinerte familien, utarbeidet i Rvt, ble komponentene i Figur 6.14 sammenkoblet. Dette var komponentene «Model Categories Picker», «Query Types» og «Value Set Picker». Den egendefinerte familien kunne hentes inn i GH herifra, i overensstemmelse med organiseringsnivået type. Typen ble hentet fra kategorien Structural Framing, da dette er kategorien

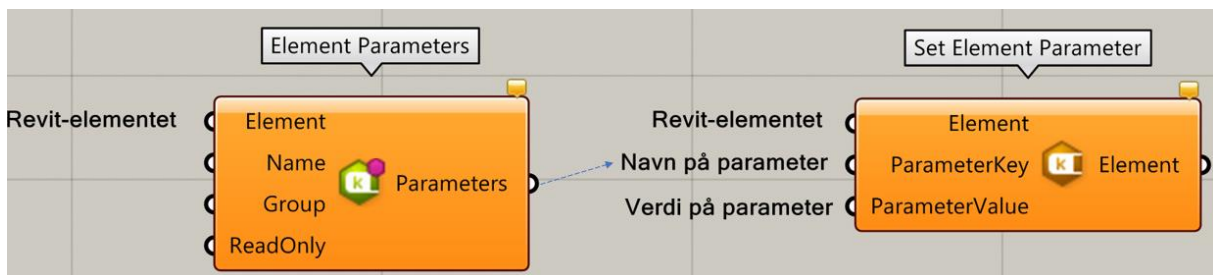


til et bjelke-element. Denne metoden for innhenting av typer for ulike kategorier, ble brukt i flere liknende sammenhenger.



Figur 6.14: Hente inn tverrsnittform til brudekket fra Rvt

Videre ble Rvt-parameterne til bjelke-elementet hentet inn, og overstyrt i GH, med komponentene fra Figur 6.15. Disse hentet ut parameterne til bjelke-elementet, ved at «Get Element Parameters» var koblet på utgangsdataen til «Add Beam», og at et utvalg av utgangsdata ble videreført til *ParameterKey* i «Set Element Parameters». Koblet så inn egendefinerte parameterverdier i *ParameterValue*.



Figur 6.15: Hente ut parameterne og overstyre dem i Rvt

Årsaken til at «New Component Family» også ble tatt i bruk for metode (1), kom av målet om å produsere flest mulig elementer direkte i GH. Her ble familien tildelt samme kategori og mal-fil, som for tverrsnittet utarbeidet som en ny familie, i Rvt.

### Armering

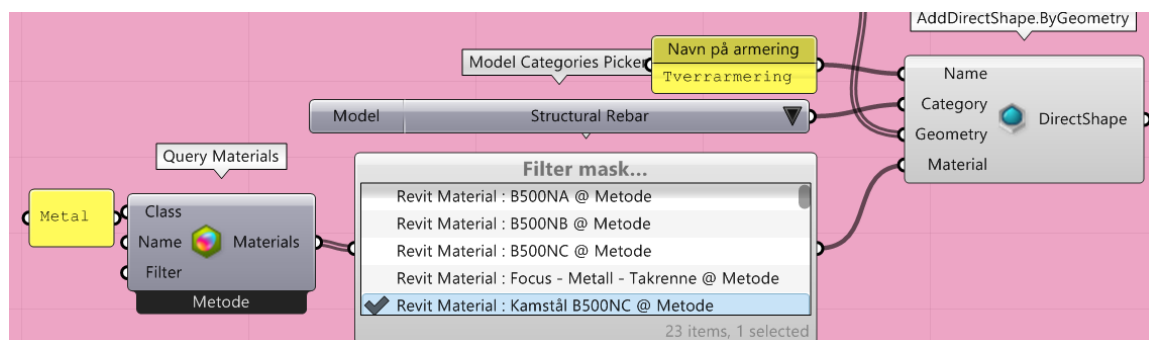
Startet prosessen med utarbeidelse av armering, ved å ta utgangspunkt i allerede etablert kunnskap om GH og Rvt. Startet med å undersøke hvilke parametere som var å hente ut fra selve brudekket, produsert i komponenten «Add Beam». Det ble eksperimentert med å overstyre de armeringsrelaterte parametere, uten å lykkes.

Den andre metoden som ble tatt i bruk, hadde sin bakgrunn fra «Free Form» metoden i Rvt, beskrevet i kapittel 5.1.1. Denne metoden bruker som sagt overflatene og kurven til et bjelke-element, for produksjon av armering i lengde- og tverretning. Til uthenting av overflatene av bjelke-elementer i GH, ble «Element Geometry» brukt. Denne komponenten konverterer Rvt-elementer inn i Rhino, for uthenting av kantlinjer og overflater. Dette ble gjennomført for å undersøke hvilke muligheter som kunne blitt brukt i en potensiell egendefinert Python-komponent, for produksjon av «Free Form» armering slik som metoden i Rvt. Her ble detaljeringsnivået på Rvt-elementet avgjørende, for gyldig inngangsdata. Det kom av at detaljeringsgradene definerer hvordan elementet presenteres i Rvt visningen.

Den tredje metoden, som ble gjennomført for produksjon av armering i GH, var å ta i bruk «AddDirectShape.ByGeometry». Denne komponenten var allerede benyttet til produksjon av brudekker fra Figur 6.13. Metoden ble belyst i forumet til RiR, nevnt i kapittel 6.3.1. Komponentens bruk ble utvidet til å produsere geometri i Rvt, altså produksjon av armering i henhold til kurver. Dette var den eneste tilgjengelige metoden, for bruk av ferdigkomponert komponent, funnet i opplæringsprosessen. Metoden fra forumet ble utvidet, ved å benytte lukkede rør, som geometri for produksjon av 3D armeringen. Denne tankegangen ble etablert etter utarbeidelsen av bjelke-element med samme komponent.

Produksjonen av armering med «AddDirectShape.ByGeometry», ble gjennomført ved å dele opp armeringskurvene i to deler; tverrarmring og lengdearmring. Hver av dem ble produsert med hver sin «AddDirectShape.ByGeometry», for å skille produktene i Rvt. Begge kurvene, ble utformet i henhold til tverrsnittformen produsert som GH-linjer, og plassert i startpunktet til brudekket. Den samme geometrien som tverrsnittsprofilen, ble benyttet til produksjon av en «Sweep». Armeringen ble i tillegg utformet i henhold til veilinen til brudekket. Dette ble gjort, for at armeringen skulle bli styrt av omrisset til tverrsnittet og lengden av brudekket.

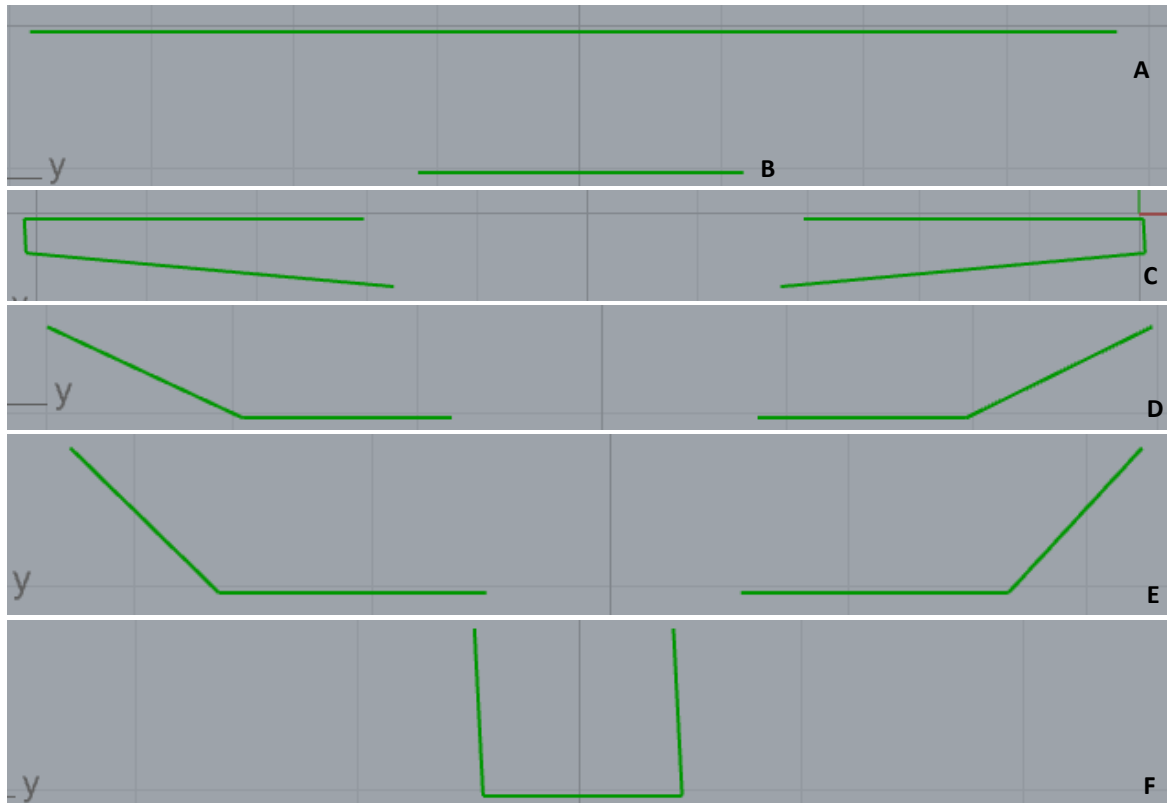
Etter at kurvene var definert, ble disse brukt som referansegeometrien for produksjon av lukkede rør. Rørene ble utarbeidet for å tilsvare armeringsjern som 3D elementer. Disse ble derfor koblet inn i hver sin *Geometry* inngang. Data til de andre inngangsparameterne, er fremstilt i Figur 6.16.



Figur 6.16: Armering med komponenten "AddDirectShape.ByGeometry"

Tverrarmeringen ble utformet som vist i A-E i Figur 6.17. Denne kurveutformingen ble valgt i henhold til samtale med Robert, fremfor innhold i RIB tegningene til Sagåne bru. Det kom av at denne bruene var prosjektert med både slakkarmert og spennarmert armering. Løsningen her, var ikke optimal i dette prosjektet, da spennarmert ikke er inkludert. I henhold til samtalen med Robert, ble tverrarmeringen bestående av seks type armerings-elementer. Disse er listet opp under, samt presenter i Figur 6.17.

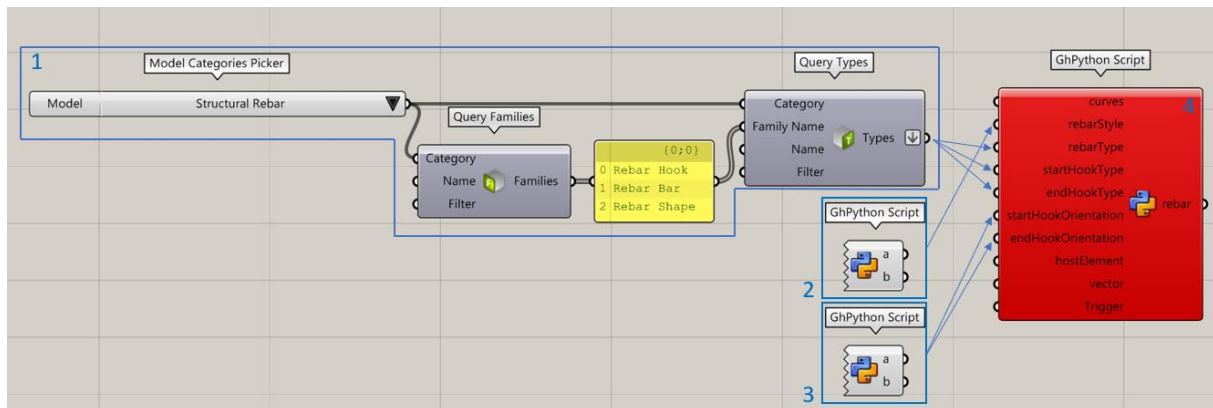
- A. Lengdearmring tverretning OK dekke
- B. Lengdearmring tverretning UK bjelke
- C. Bøyler i dekket
- D. 1. sett bøyler i bjelke
- E. 2. sett bøyler i bjelke
- F. 3. sett bøyler i bjelke



Figur 6.17: Referansekurvene til tverrarmeringen

Armeringen i lengderetningen ble bestående av stenger, med formen til «Gjeldende kurve». Her ble det ikke utarbeidet forankring, eller spesifikke prosjekteringsdetaljer. Kurven ble utarbeidet mindre detaljert, grunnet tidsbegrensningen på oppgaven. Lengdearmeringen ble produsert for å visualisere bruken av parametrisk armering til prosjektering av et brudekke.

Det siste forsøket på å modellere armering, var et forsøk på å produsere 3D armering i Rvt. Til dette ble det tatt utgangspunkt i en Python-komponent, for å få integrert at elementet ble strukturelt i Rvt, og ikke bare visuelt. Tok i bruk kunnskap beskrevet i 6.3.1, for å bruke Python-komponenten «RebarFromCurve», produsert av Jorgen Sorbol, i juli 2020. Komponentene var tilgjengelig for nedlastning, og ble derfor tatt i bruk. Den var utarbeidet i henhold til innhold av tilsvarende komponent for Dynamo. Tankegangen bak inngangsparameterne til «RebarFromCurve», samt komponenten i seg selv, ble tatt i bruk. Hvordan dette ble gjennomført er vist i Figur 6.18, og beskrevet punktvis under.



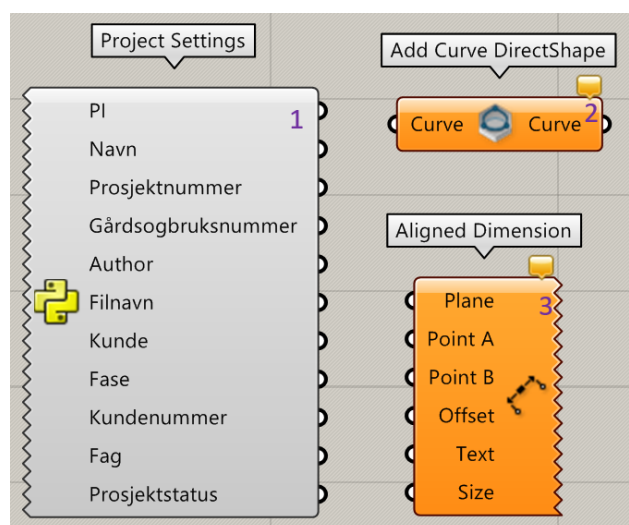
Figur 6.18: Bruk av Python-komponent for 3D modellering av armering i Rvt

1. Hentet inn kategorien til strukturell armering, og alle typene i denne kategorien (tre stykker). Disse ble brukt for å definere armeringstype (Rebar Shape) og vinkelen på endeforankringen (Rebar Hook), i inngangene til (4).
2. Benyttet Rvt API fra nettsiden [92], for å gi valgmuligheten til utformingen av armeringen, i form av stenger eller bøyer. Utgangsdataen her ble koblet til *RebarStyle*.
3. Tok i bruk Rvt API fra nettsiden [92] for å gi valgmuligheten til hvordan endeforankringen skulle være orientert; til høyre eller til venstre for armerings-stangen. Utgangsdataen her ble koblet til *RebarHookOrientation* (både start og end).
4. (1), (2) og (3) ble koblet inn i inngangene til Python-komponenten, for produksjon av armering i henhold til kurvene koblet inn i *Curves*. Her ble også resterende innganger, koblet på med datainnholdige variere.

### Sekundærelementer

Sekundærelementer i denne gruppen, var tilleggselementer til nivåene og aksene beskrevet i kapittel 6.4.3. Dette var elementer som kun ble utarbeidet for Rvt-modellen, og som ikke ble benyttet i samhandling med Fd modellen. Sekundærelementene var målsetting, snitt og prosjektinnstillinger. Til disse formålene ble det tatt i bruk komponentene som er presentert i Figur 6.19. Fremgangsmåtene som er benyttet i henhold til hver av komponentene, er punktvis gjengitt.

1. Tok i bruk «Add Curve DirectShape», for produksjon av fysiske linjer i Rvt. Disse ble utformet i henhold til veillinjen til brudekket, i xy-planet. Denne beslutningen ble tatt, da et snitt utformet i Rvt utarbeides i 2D visninger. Begrensningen var produksjon av snitt i skråplan (xz- eller yz- planet).
2. Hentet inn prosjektinnstillingene til den åpne Rvt-filen. Dette ble gjennomført med komponenten «Project Settings», lastet ned fra [93]. Her ble parameterne overstyrt, med samme fremgangsmåte som beskrevet for Figur 6.15, tidligere.



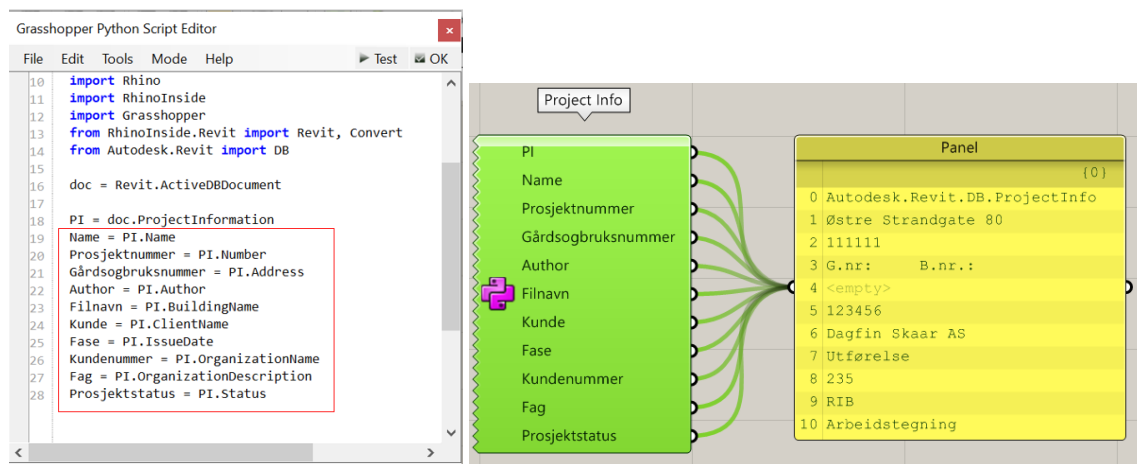
Figur 6.19: Komponenter for produksjon av sekundærelementene i Rvt

3. For produksjon av målsetting til brudekket, ble komponenten «Aligned Dimension» tatt i bruk. Dette er en GH-komponent, som produserer målsetting i Rhino. Det er altså ikke en Rvt-komponent.

Linjeutformingen til bruk som referansegeometri, for modellering av snitt i Rvt, kom av at det er størst nytteverdi å få et snitt vinkelrett på et brudekke i rommet. Dette er viktig for å få kontroll av nøyaktig fremvisning av elementene i snittet.

Prosjektinnstillingene ble overstyrt, for at de kunne benyttes for produksjon av ark i Rvt.

Komponenten «Project Settings» [93] ble endret til å bestå av norske inngangsnavn. Disse navnene ble også integrert i kodene til komponenten, se rød ramme i Figur 6.20. Navnene som ble benyttet var hentet fra DS-definisjonene, som brukes på tegningsprodukter.

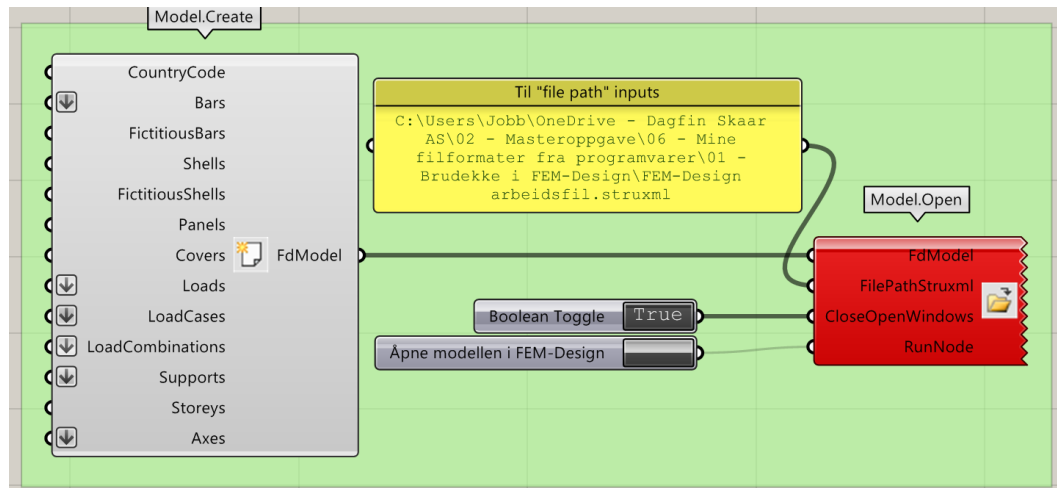


Figur 6.20: Kodene som ble endret i komponenten til prosjektinnstillingene

#### 6.4.5 FEM-Design modeller

Der hvor det var nødvendig for en dypere forståelse av komponentenes innhold, ble nettsiden GitHub benyttet. Her var det mulig å finne kodene som var innebygd i komponentene til Fd [54]. Denne informasjonen er tilgjengelig, da disse komponentene er en åpen ressurs.

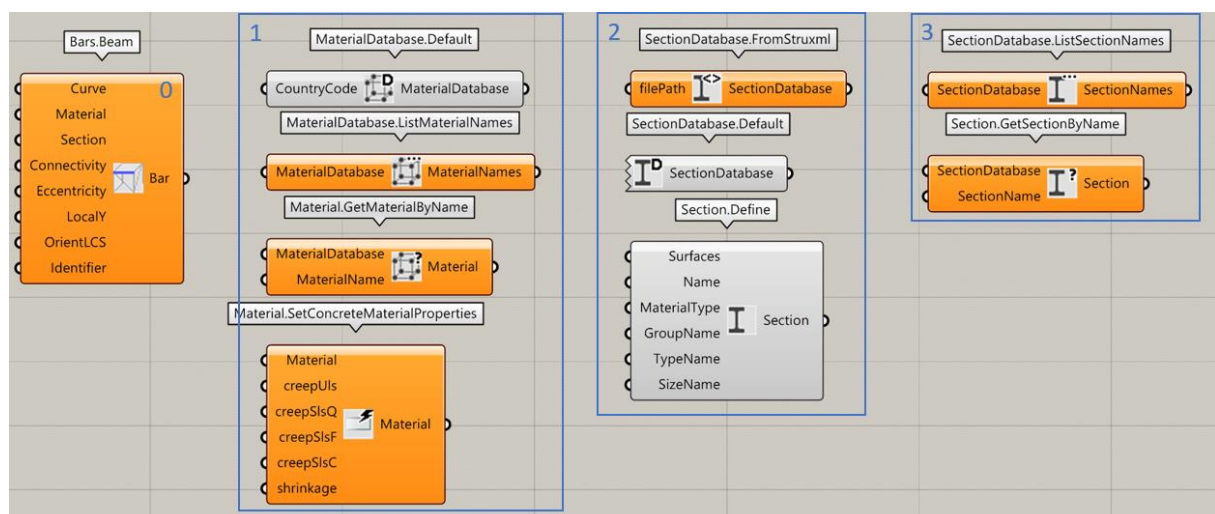
GH ble ikke åpnet via Fd, slik som for Rvt. Det medførte at produksjonen av Fd innhold i GH, ikke automatisk ble presentert i Fd, mens produksjonen av skriptene pågikk. Produktene til Fd i GH ble overført ved å ta i bruk én struXML fil, lagret på intern server. Det var nødvendig å utarbeide muligheten til overføring, allerede ved første produkt som ble produsert i GH. Hvert produkt i GH måtte kontrolleres i Fd, for å sikre at skriptet representerte tilsvarende innhold i Fd. Til dette formålet, ble det utarbeidet en gruppe med komponenter som gjennomførte overføringen, se Figur 6.21.»



Figur 6.21: Gruppen benyttet for produksjon av Fd-modell

I denne gruppen, ble innhold av prosjekteringselementene i beregningsmodellen, koblet til inngangene uthevet med «Flatten»-funksjonen (pil ned), hos inngangene til «Model.Create». I tillegg ble den norske eurokoden valgt i *CountryCode*. For at ikke Fd modellen skulle åpnes hver gang ny data ble koblet inn i inngangene, ble komponenten «Model.Open» utstyrt med en «Button». Denne gav en manuell kontroll over når data produsert i GH, skulle overføres og åpnes i Fd, direkte fra GH.

For produksjon av prosjekteringselementene i Fd, ble disse etablert i henhold til rekkefølgen på fanemenyene i Fd. Derfor ble det startet med utarbeidelse av brudekke og tverrbærere. Disse ble utformet som bjelke-element, hvor komponenten «Bars.Beams» ble benyttet. Denne komponenten er presentert som 0 i Figur 6.22. Brudekket ble opplagret på tverrbæreren, for å idealisere en realistisk modell; grunnet forholdet mellom bredden og lengden til brudekket. Her ble kantdrager integrert i modellen som linjelast, fremfor modellert som et fysisk element. For inngangen *Curve*, ble «gjeldende kurve» for produksjon av brudekke benyttet. For tverrbæreren, ble det utarbeidet hver sin rette linje i start- og endepunkt til «gjeldende kurve», i retning med brudekkets bredde. Hvordan de resterende inngangene til (0) ble utarbeidet, er presentert punktvis under Figur 6.22.

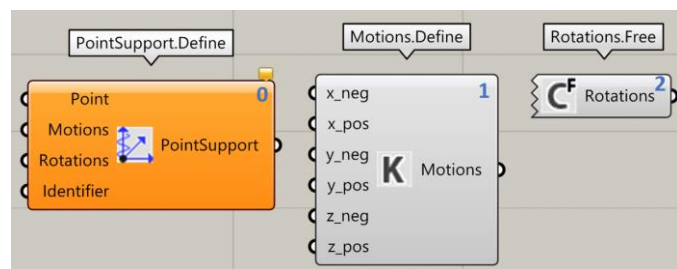


Figur 6.22: Komponenter til produksjon av brudekke og tverrbærere i Fd

1. Benyttet betongkvaliteter fra databasen i Fd. Her var den norske eurokoden brukt. Navnene på kvalitetene ble listet opp, og hentet ut fra databasen. Deretter ble egendefinerte verdier for kryp og svinn definert.
2. Hver komponent definere tverrsnittsprofilen, på hver sin måte. Første komponent, tok utgangspunkt i databasen til struXML-filen i Fd. I denne filen var det utarbeidet et egendefinert brudekke-tverrsnitt i Fd, med passende maltype. For tverrsnittet til tverrbærerne, ble «SectionDatabase.Default» benyttet, som tok i bruk eksisterende tverrsnittsprofiler i ordinær FEM-Design database. Den tredje og siste metoden for valg av tverrsnittsprofil, tok i bruk «Section.Define». Denne ble brukt for å produsere egendefinerte tverrsnitt i GH, med en overflate i xy-planet som referansegeometri. Denne metoden utarbeidet tverrsnittsprofilen til brudekket og tverrbærerne.
3. Begge komponentene her, ble benyttet i sammen med hver av komponentene i (2). De hentet ut tverrsnittsformen fra Fd databasen, til bruk i inngangen *Section*.

For inngangsparameteren *Surfaces* i «Section.Define», var gjeldende for tverrsnittsformen definert i xy-planet. Dette var samme tverrsnittsform, som ble benyttet til produksjon av Rvt-brudekket med «DirectShape» til Rvt. Her ble overflaten tilført mulighet for tverrfall, ved at den ble rotert i xy-planet. rotasjonen ble utført med påført tverrfall og påført tverrfall uten naturlig tverrfall.

Videre ble oppleggene til brudekket, plassert på tverrbæreren. Disse ble produsert med «PointSupport.Define», hvor bevegelsene (1) og rotasjonsgraden (2) ble egendefinert i GH, se Figur 6.23 . Her ble det utarbeidet punkter på linjene som definerte senterlinjen av tverrbæreren, med en spesifikk avstand mellom dem.



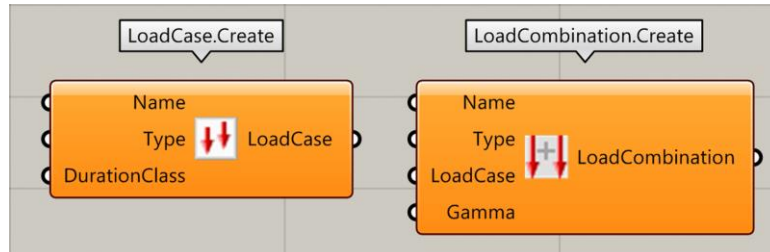
Figur 6.23: Komponenter for produksjon av opplegg i Fd

### Laster

Lasttilfellene, lastkombinasjonene og lastene ble definert og modellert i GH, med ferdigkomponerte komponenter. Utarbeidelsen av inngangsdata til disse, tok utgangspunkt i lastplasseringen i brudekkets lengderetning. Her ble det tatt utgangspunkt i å plassere lastene for dimensjonering av maks moment og maks skjær, i lengderetningen til brudekket, og torsjon i tverretningen. Denne avgjørelsen ble tatt i samarbeid med Robert fra DS. Det ble bestemt at lastene skulle plasseres i henhold til punkter og linjer på brudekket, på følgende måter:

1. punktlast på midten av spennvidden
2. x antall punktlaster på halve spennvidden
3. x antall punktlaster på hele spennvidden
4. punktlast med x størrelse fra opplegg
5. punktlaster over hele spennvidden, med x-avstand fra start- og endepunkt
1. linjelast over hele spennvidden
2. Linjelast over halve spennvidden
3. Linjelast over hele spennvidden, med x-avstand fra start- og endepunkt

Etter at lastplasseringene var avklart, ble lasttilfellene og lastkombinasjonene utarbeidet i GH med komponenten «LoadCase.Create» og «LoadCombination.Create». I begge komponentene, ble det utarbeidet inngangsdata med delvis parametriserte avhengigheter, for å gi mulighet til automatiserte oppdateringer.



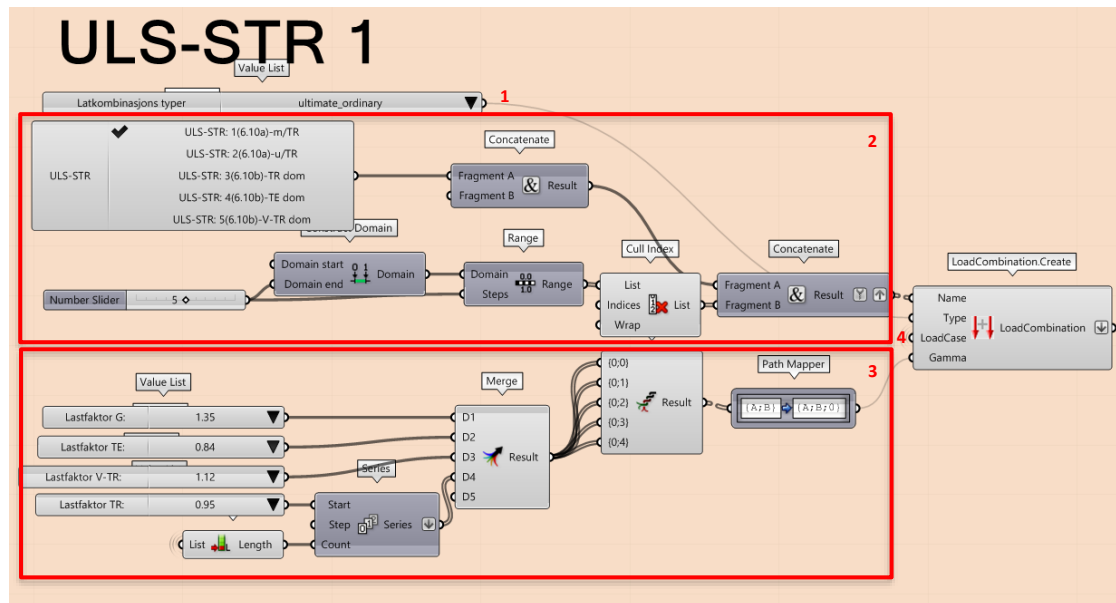
Figur 6.24: Komponentene for produksjon av lasttilfeller og lastkombinasjoner

Lasttilfellene ble gruppert i henhold til type last: egenlast, vindlast, temperaturlast og trafikklast vertikalt. Hver av gruppene fikk tildelt egne muligheter til valg av varighetsklasse og lasttype (definisjonen i Fd). Både varighetsklasse og lasttypene ble listet opp for manuell utvelgelse, i henhold til hver av gruppene, for å lettere endringsmuligheter. I hver gruppe ble det utarbeidet spesifikke lasttilfeller i henhold til lastplasseringen. Dette ble gjort for at det skulle være enklere å utarbeide lastkombinasjonene i henhold til lasttilfellene.

Lastkombinasjonene ble utarbeidet i henhold til type lastkombinasjon og lastfaktor. Disse ble hentet fra V668 SVV, som er presentert i casen. Denne beslutningen ble tatt i samarbeid med Martin og Robert, da dette er generelle lastkombinasjoner, som benyttes til standard bruprojektering. For å enklere beskrive hvordan lastkombinasjonene ble utarbeidet i GH, er eksempelet på første lastkombinasjon av typen ULS.STR, presentert i Figur 6.25. Denne metoden ble brukt for hver lastkombinasjon, som totalt innebar 11 ulike lastkombinasjonstyper. Her ble ikke ULS.STR tilfellet med vindlast uten trafikk medregnet, da denne kombinasjonen ikke var dimensjonerende på Sagåne bru.

Hver av de 11 lastkombinasjonene fikk tildelt en liste med typer å velge mellom (1), samme fikk navn (2) og sikkerhetsfaktor (3). I hver lastkombinasjon ble de tildelt lasttilfeller som inkluderte alle mulige kombinasjoner av lastplasseringene (4), beskrevet over. Listene ble utarbeidet for å enkelt gi mulighet til å endre innhold, i henhold til lastkombinasjonene i v668.





Figur 6.25: Skriptoppsett for produksjon av lastkombinasjoner

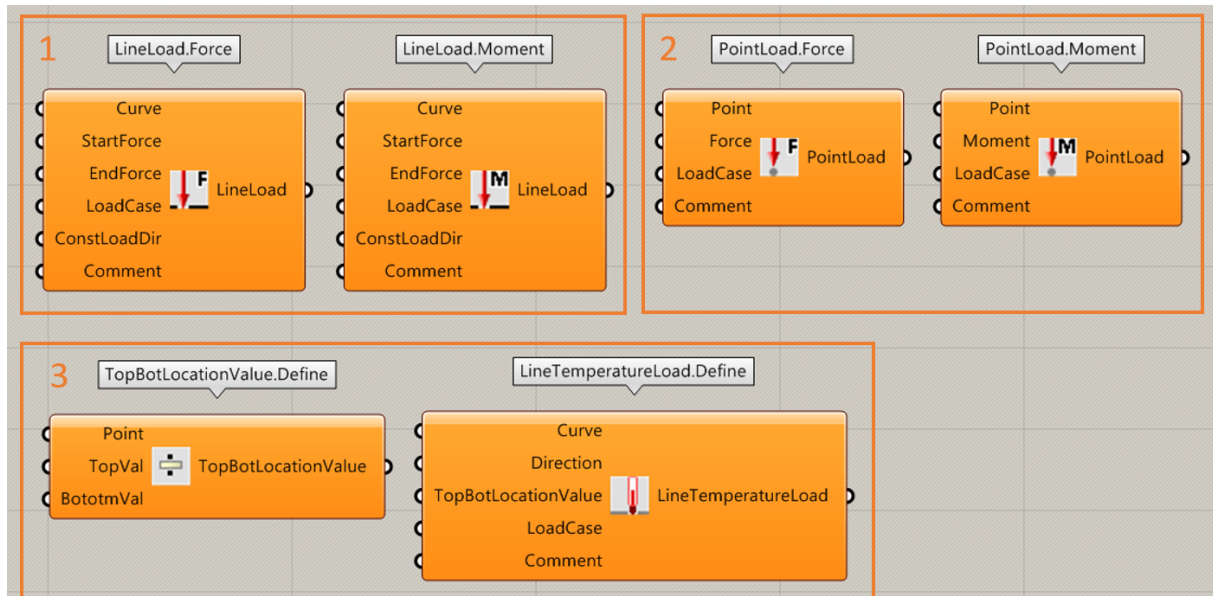
Det ble i tillegg gjennomført en sammenlikning av definisjonene i Fd og GH, som er presentert i Tabell 6.9. Dette er definisjonene til lasttilfelle typene, varighetsklassene og lastkombinasjonstypene. Denne informasjonen ble inkludert i skriptet, for at andre brukere enkelt skal få tilgang på denne informasjonen.

Tabell 6.9: Definisjonene av de ulike typene av lasttilfeller i Fd versus i GH

	FEM-Design	Grasshopper
Lasttilfelletyper	Ordinary	static
	+Struc. Dead load	dead_load
	+Soil dead load	soil_load
	+Shrinkage	shrinkage
	+Chamber sim.	prestressing
	+Fire	fire
	+Seis load, F+Mx	seis_sxp
	+Seis load, Fx-Mx	seis_sxm
Varighetsklasser	Permanent	permanent
	Long-term	long-term
	Medium-term	medium-term
	Short-term	short-term
	Instantaneous	instantaneous
Lastkombinasjonstyper	U (Ultimate limit state)	ultimate_ordinary
	Ua (Ultimate -accidental limit state)	ultimate_accidental
	Us (Ultimate – seismic limit state)	ultimate_seismic
	Sq(Ultimate – seismic limit state)	serviceability_quasi_permanent
	Sf (serviceability- frequent limit state)	serviceability_frequent
	Sc (Serviceability – characteristic limit state)	serviceability_characteristic

### Laster på konstruksjonen

Lastene på konstruksjonen ble produsert som punktlaster (1), linjelaster (2) og temperaturlaster (3). Disse ble benyttet for produksjon av krefter og moment, og ble tildelt kurve/er (*Curve*) eller punkt/er (*Point*), i henhold til plassering av lastene i lengderetning (beskrevet over). Til disse formålene ble komponentene i Figur 6.26 benyttet.



Figur 6.26: Komponenter benyttet til produksjon av laster

1. Definerte start og endekraft, i begge komponentene. Disse ble definert ved å ta i bruk vektorer for å indikere størrelse og retning på kraften (vinkelrett på veilinjens vertikalt)
2. Her ble også krefter og momenter definert slik som i (1).
3. Definerte topp og bunnverdi på temperaturfordelingen i første komponenten her. Denne ble brukt i inngangen *TopBotLocationValue*.

Egenlastene ble modellert som kraftpåkjenninger, grunnet symmetrisk tverrsnitt. For alle vind- og trafikklastene ble disse modellert både som moment- og kraftpåkjenning. Disse lastene har ikke symmetrisk plassering i tverretning, og vil gi en effekt av torsjon. Alle lastene ble plassert på «Gjeldende kurve». Her ble vindlastene forenklet til et statisk system, i henhold til vindlastpåføring i prosjektet Birkelid bru. Lasten ble tiltenkt å være trekantlast med en resultantkraft/moment påført «Gjeldende kurve», med en eksentrisitet. For mer detaljert beskrivelse av konseptet, se Vedlegg C.

Trafikklastene til LM1, ble delt inn i gruppene TS og UDL. TS ble modellert som punktlaster i form av moment og kraft. UDL ble modellert på samme måte, bare som TD og i form av linjelaster. Antall krefter og momenter per punkt og linje, ble utarbeidet som en parametrisk sammenheng, hvor styrende faktor ble valgt til å være antall kjørebane, bredden til kjørebane og restområdet. Det ble i henhold til casen, produsert tre punktlaster og fire linjelaster per punkt og kurve på veibanen.

### FEM-Design armering

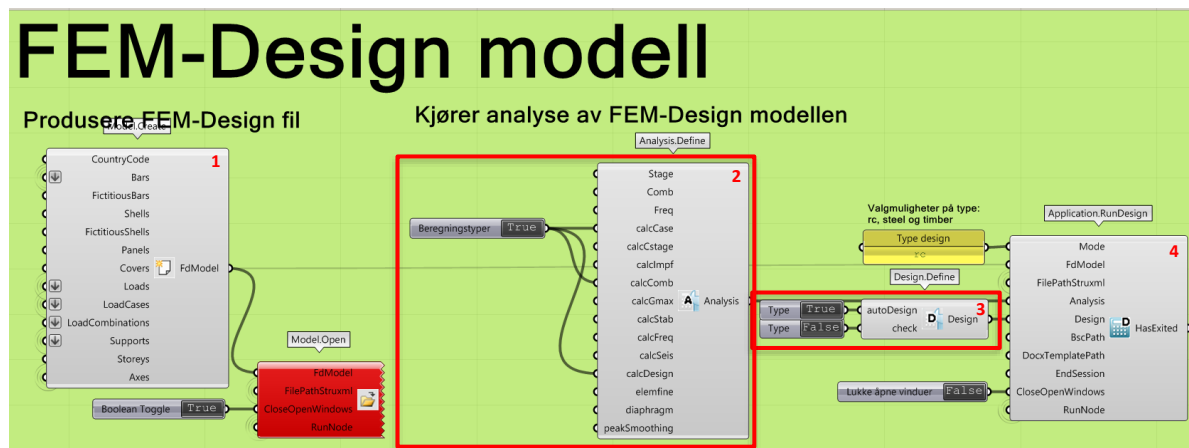
For produksjon av armering til brudekket i GH, var det ikke mulig å ta i bruk de ferdigkomponerte komponentene, da dette innebar produksjon av armering i plater. Den eneste relevante

komponenten, var «Wire-Define», som ble bruk til å produsere en egendefinert armeringsstang. Det ble ikke funnet komponenter for utarbeidelse av stang- og bøylearmering i et bjelke-element. Ut ifra opplysningen som fremkom med å holde musetasten over inngangene, ble det utarbeidet data til inngangsparametere i «Wire.Define». Men komponenten ble ikke brukt til noe formål, da utgangsdataen ikke kunne kobles inn noen av inngangene til «Model.Create».

Da manuell armering ikke ble gjennomførbart i GH, i henhold til opparbeidet kunnskap om GH og Fd, ble det bestemt at armerings-behovet skulle utarbeides i RC analysen. Til dette formålet, ble det benyttet ferdigkomponerte komponenter, som vil bli presentert i analysekapittelet under.

## Analyse

For å gjennomføre analysen av beregningsmodellen i GH, ble det etablert en Fd-modell i GH. Til dette ble all data koblet inn i inngangene med «Flatten»-funksjonen til «Model.Create», i Figur 6.21 og som (1) i Figur 6.27. Gjennomføringen av analysen i GH er presentert i Figur 6.27, og beskrives videre.



Figur 6.27: Komponenter benyttet for å gjennomføre RC beregninger

Målet var å gjennomføre RC beregning (4) av FdModellen (1), hvor tre analyser ble valgt i «Analysis.Define» ved å aktivere: *CalcCase*, *CalcComb* og *CalcDesign* (2), med en sann boolsk operasjon. For å kjøre analysen i Fd, ble komponenten «Application.RunDesign» brukt. Her ble inngangene *Mode*, *FdmModel*, *FilPathStruxml*, *Analysis*, *Design*, *CloseOpenWindows* og *RuneNode* spesifisert. I henhold til casen, var det RC som var gjeldende i inngangen *Mode*.

For automatisk modellering av armering i analysen, ble gjennomføringen av RC beregningen (4), utstyrt med valget om å kjøre en *AutoDesign* eller *Check* (3) i «Design.Define». Disse valgene ble utstyrt med en boolsk operasjon, for å gi muligheten til et manuelt valg.

## 7 Resultat

Resultatene som presenteres i dette kapitlet, er innhentet i overenstemmelse til metodene presenter i kapittel 6. Dette innebærer resultater fra litteraturstudiet til formål to, erfaringer fra grenseindustrier og parametriske prosjektering.

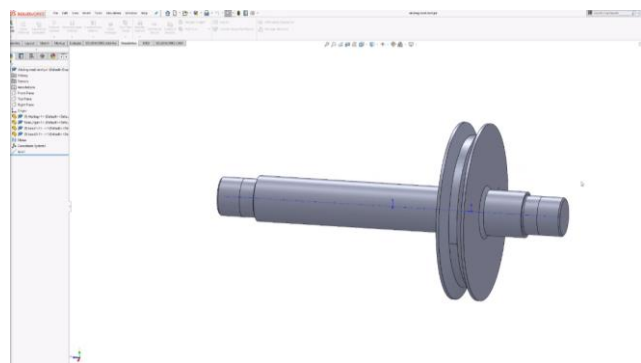
### 7.1 Grenseindustrien mekatronikk

Mekatronikk ingeniører prosjekterer blant annet mindre komponenter, mekanismer og anordninger. Her er det hensiktsmessig å styre unna rammestrukturen i et prosjekt. Rammestruktur brukes av bygg-ingeniører, da det undervises mer om last og grensebetingelser i henhold til standardiserte regelverk. For mekatronikingeniører er det nødvendig å arbeide med mer omfattende FEM analyser. Det kommer blant annet av at elementene består av kompleks- og dynamisk geometri.

For å prosjektere elementene i et prosjekt, benyttes arbeidsflyten hvor CAD-modellen importeres til eksternt FEM-program. CAD programmet Solidworks og FEM programmet Impetur AFEA kan benyttes til denne samhandlingen. Solidworks er en programvare som samhandler med FEM programmer, hvor IMPETUS AFEA har et grensesnitt mellom CAD og FEM. Andre FEM-program som kan benyttes til det samme formålet, er Abaqus og Ansys.

Komponentene som prosjekteres, modelleres først i CAD programmet. Deretter preppes CAD-filen, før den lagres som en step-fil<sup>7</sup>. Å preppe en CAD-fil, innebærer at unødvendige detaljer tas bort, og at koblinger (sveis), riktig overføringspunkt, bestemmelse av grensebetingelser og kontakt-betingelser, spesifiseres. Her inngår også inndeling av overflatene til komponenter, i henhold til lastpåføring. Preppingen gjøres i CAD-programmet, da grensesnittet i FEM er begrenset. Dette arbeidet er tidkrevende, men ytterst nødvendig.

Et eksempel på en type komponenter som modelleres i CAD-programmet Solidworks og importeres i FEM-programmet, er illustrert i Figur 7.1. Denne sammenstillingen består av tre elementer: en aksel, en reimskive, samt to sveiser. CAD-modellen er ferdig preppet, og klar til ekstern FEM- analyse. Preppingen som er involvert her, innebærer detaljert modellering av sveisene, oppdeling av overflatene og definering av kontaktbetingelsene, mellom sveisene og reimskiven. Oppdelingen av overflaten utbedres, for å få riktig påkjennings-simulering på komponenten.



Figur 7.1: Eksempel på preppet sammenstilling, modellert i Solidwork

For å oppnå en optimalisert prosjekteringsprosess, kan en bedre løsning på dette, være at alt arbeid skjer i én programvare. At produksjonen av både CAD- og FEM-produkter, er gjennomførbart i samme program. Solidworks er integrert med en FEM-analysemodul. Sammenliknet med andre

<sup>7</sup> .step er et filformat som kan leses av både CAD og FEM programmer.

eksterne FEM-program, mangler denne modulen en mengde muligheter. Modulen prøver å være brukervennlig, men oppleves ikke slik. I tillegg inneholder FEM-programmet Abaqus enkle CAD-verktøy. Disse verktøyene er på lik linje som med FEM-modulen i Solidworks, lite brukervennlig.

## 7.2 Litteraturstudie formål to

Resultatene til formål to av litteratursøket, presenteres i dette delkapittelet. Resultatene representerer innholdet til masteroppgavene som ble inkludert i Tabell 6.2, i metodekapittelet. Her presenteres masteroppgavens forfattere, tittel, formål, brutype, benyttede programvarer, arbeidsflyten til programvarene, metode og konklusjon i Tabell 7.1 og Tabell 7.2. Utformingen av tabellene, er utarbeidet med inspirasjon fra [79].

I Tabell 7.1 fremkommer det at det karakteristiske med oppgavene, er at har tatt for seg en arbeidsflyt, for å effektivisere eller forbedre prosjekteringsprosessen av brukonstruksjoner. Til dette har de benyttet parametriske verktøy i en samhandling med å utarbeide beregningsprodukter internt og eksternt i eksternt program (Master 2 og 3), og kun i eksternt program (Master 1 og 3). Master 1 har også benyttet parametriske verktøy for å produsere geometrisk modell i det eksterne programmet Rvt.

Tabell 7.1: Mål, programmer og arbeidsflyt til inkluderte masteroppgaver

Maste	1	2	3	4
Forfatter	2018 Daniel Nilden og Øystein Urdal NTNU	2017 Christian Due Lind Og Kasper Hagestande NTNU	2019 Espen Rønnes NTNU	2017 Herman Enger Aas og Krustian Mathias Eick NTNU
Tittel	Bruprosjektering med bruk av parametriske design	Konseptuell design av gangbruer	Optimalisering av nettverksbuebru «Parametriske design som optimaliseringsverktøy»	Conceptual structural design of footbridges – A case study of asymmetrically suspended bridges using parametric modelling
Formål	Effektivisere tegnings- og beregning- produksjon	Benytte parametriske verktøy i den konseptuelle fasen av brudedesign for beregningprodukter	I hvilken grad Dynamo og RSA kan benyttes som et optimaliseringsverktøy som en automatisert prosess	Utarbeide konseptuelle gangbrudedesign

Brutype	Etteroppspent veibru	Gangbru som dobbelbue-bru	Nettverksbuebru med radiell utforming	Gangbruer som består av to dekker koblet sammen med trapper. Et av dekkene er kurvet.
Program	Revit Dynamo SOFiSTiK	Grasshopper Karamba Robot Structural Analysis (RSA)	Dynamo RSA Excel	Grasshopper Karamba Galapagos RSA
Arbeidsflyt	Produsere tegnings og beregningsmodell i Revit med Dynamo, som ble brukt i SOFiSTiK	Produsere brumodell i GH, som analyseres med Karamba inne i GH. Gjennomførte formgenerering i henhold til spesifikke grenseverdier med Galapagos. RSA benyttes for sammenlikning av FE-analysen hos Karamba.	Produsere parametrisk modell i Dynamo, som sendes til RSA for analyse. Analysen sendes til Excel. Her defineres oppdaterte parametere som sendes til Dynamo	Tar i bruk Grasshopper for parametrisk tegning, og Karamba inne i Grasshopper for å utføre hypotesetesting av FE analyse. Her brukes Galapagos i kombinasjon med Karamba, til optimalisering. Robot ble brukt for å sammenlikne FE-analyser og generere lastkombinasjoner.

Det ble tatt i bruk programvarer som hadde som formål å gjennomføre effektiviserende beregninger, i den konseptuelle fasen, og som optimaliseringsverktøy. Formålene var forbeholdt etteroppspente vegbruer, gangbru med dobbel bue, nettverksbuebru med radiell utforming, og gangbru med to dekker koblet sammen med trapper. I sistnevnt brutype, har det ene dekket buet utforming. I master 1, var formålet i tillegg til beregningsundersøkelser, å effektivisere tegningsproduktene i Rvt.

Master 1 og 2 tok i bruk det parametriske verktøyet Dynamo, mens master 2 og 4 benyttet GH. Her ble GH benyttet til utforming av konstruksjon, samt gjennomføre FE-analyse i Karamba. Resultatene ble så sammenliknet med FE-analysen i RSA. Beregningene ble benyttet til å bestemme designet av brutypene. Master 1 benyttet Dynamo, både til tegningsproduksjon i Rvt, samt gjennomføring av beregninger i FE-programmet SOFiSTiK. I Master 2, ble Dynamo kun benyttet til å produsere FE-analyser i RSA.

I Tabell 7.2 presenteres metodene som ble benyttet, til å foreta endringer på utforming og beregningsanalysene. Endringene ble gjennomført med ulike metoder. Master 1, gjennomførte fire arbeidsflyter, med utvelgelse av den beste. Denne ble brukt til å produsere effektivisert tegnings- og beregningsprodukter. Master 2, utvidet et utvalg analyser med bruk av parametrisk design. Master 3, gjennomførte tre ulike analyser, som ble sammenliknet før optimaliseringen startet. Mens Master 4, benyttet hypotesetesting med analyseforbedringer, i henhold til design.

Tabell 7.2: Metodene og konklusjonene til de inkluderte masteroppgavene

	1	2	3	4
Metode	Undersøke 4 arbeidsflyter  Sammenliknet analyser  Endret geometri for å identifisere nytteverdien	Utvidet analysen.  Gjennomførte endringer gjort av arkitekturisk eller strukturelle årsaker.  Innhente erfaringer med parametrisk design	Gjennomførte 3 ulike analyser  Sammenliknet analyser  Optimaliserer en av analysene med å foreta utformingsendringer	Gjennomførte hypotesetester  Gjennomførte analyseforbedringer i henhold til design
Konklusjon	Oppnår god beregningsmodell  Begrensninger for målsetting og god tegningsmodell  Krever analytisk modell i Revit for arbeidsflyten  Arbeidsflyten er best egnet for konseptuell fase	Kan utforske et bredt spekter av utforminger  Effektivisering av å identifisere utformingspåvirkninger på bæresystemet.  Form-generering gir positive reduksjoner av effekter i analysen. Finner enkelt verste lastplassering.  Ikke mulig å gjennomføre dynamisk analyse	Får en effektiv og nøyaktig modellering og redigering  Får automatisert prosessen, som medfører tilleggsberegninger av lastpåføring, lastkombinasjoner og analysetype.	Får gjennomført en rask utforsking av konseptuelle designideer.  Geometrien tilpasses enkelt, og gir øyeblikkelige resultater. Dette er med på å få frem indikatorene som reagerer på endringer.

Metodene resulterte i konklusjonene, i nederste rad i Tabell 7.2. Her fremkommer det at Master 1 konkluderer med at parametrisk design, er best egnet for konseptuell fase. Dette er fasen, som de tre andre masterne hadde brukt som begrensning.

Videre konkluderer Master 1, med at beregningsmodellen i SOFFiSTiK er god. Slutningen ble tatt etter at parametrisk og tradisjonell beregningsmodell var sammenliknet. I Master 2, hvor form-generering var brukt, gav dette en positive reduksjoner av effektene nedbøyning, utnyttelsesgrad, lastkombinasjoner og kjøretid i analysen. Her var det også mulig å enkelt finne verste lastplassering. Master 3, konkluderte med at parametrisk beregninger kunne gjennomføres, som en automatisert prosess med designet produsert i GH. Her var det imidlertid nødvendig å gjennomføre tilleggsberegninger av lastpåføring, lastkombinasjoner og analysetype. I Master 4 var det klart at det er mulig å gjennomføre en rask utforsking, som samtidig gir muligheter for nøyaktig og effektiv modellering, samt redigering av modellen.

Master 4, konkluderte også med en sammenheng mellom samhandlingen av geometrien og resultatene som parametrisk utforming. Tilpasset geometri gir øyeblikkelige resultater. Disse brukes som indikatorer til å få frem geometriske endringer i henhold til reaksjonen i analysen. Dette inngår også i Master 2, ved at parametriseringen gir en effektiv mulighet til å utforske forskjellige utformingspåvirkninger på bæresystemet. For beregning av dynamiske effekter i Karamba, konkluderes det med at dette ikke er mulig i master 2.

Master 1 er den eneste som har spesifikt fokusert på tegningsproduktene, i tillegg til beregningsproduktene. Her ble det tatt en slutning med at arbeidsflyten fra Dynamo til Rvt og fra Rvt til SOFiSTiK, krever utarbeidelse av analytisk modell i Rvt. Her oppstår det begrensninger for målsetting og produksjon av et optimalt tegningsprodukt. For utarbeidelse av analytisk modell, krever en oppdeling av buede elementer, til korte rette elementer.

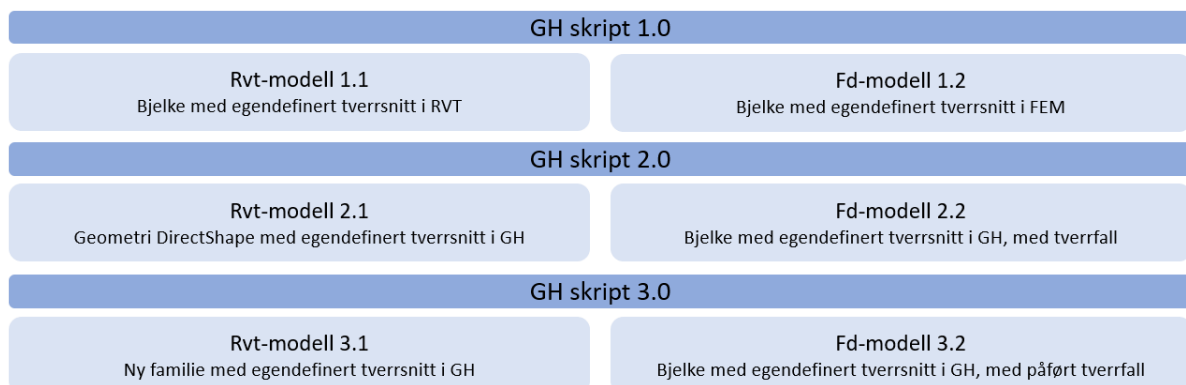
### 7.3 Parametrisk prosjektering

Målet med parametrisk prosjektering, innebar benyttelsen av arbeidsflyten i Figur 6.2, fra metodekapittelet. Her var fokuset, å oppnå effektive revisjonsmuligheter, av automatiserte tegnings- og beregningsmodeller. Hvordan denne arbeidsflyten er gjennomførbar til å produsere og revidere prosjekteringselementene i Rvt og Fd, inkluderes i resultatene. Resultatene presenteres med følgende overskrifter og rekkefølge:

1. Grasshopper-skript: til produksjon og revisjoner av prosjekteringselementer i Rvt og Fd
2. Arbeidsflyten for produksjon: til prosjekteringselementene i geometrisk- og beregningsmodell
3. Grasshopper produktene: resulterende produkter i Rvt og Fd, produsert med GH
4. Parametriske revisjonsmuligheter: mulig revisjoner og brukervennlighet

#### 7.3.1 Grasshopper skript

Resulterende GH-skript er inndelt i tre individuelle filer. Disse er presentert i Figur 7.2, hvorav hvert skript består av produksjon av prosjekteringselementene i Rvt og Fd, samt revisjonsmuligheter. Filene til hvert skript, er vedlagt i Vedlegg G, hvorav et oversiktsbilde av hvert skript er plassert i Vedlegg H. Videre presentasjon av resulterende parametrisk prosjektering, skjer i overensstemmelse til disse tre GH-skriptene, med tilhørende Rvt- og Fd-modeller.

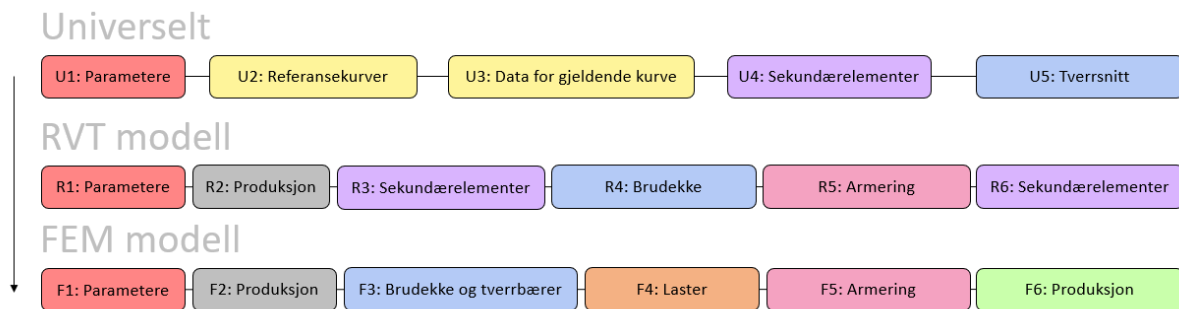


Figur 7.2: Resulterende GH-skript



Inndelingen av skriptene indikerer ulik brudekkeproduksjonen. Dette er hovedelementet i denne oppgaven, og danner derfor grunnlaget til de resterende prosjekteringselementer, foruten prosjektinnstillingene i Rvt. Fd-modellene over, inneholder også ulik utformingen av tverrbærerne.

Hvert av GH-skriptene inneholder samme oppbygning og logikk. Skriptinnholdet er delt inn i de tre hoveddelene: universelt, Rvt-modell og Fd-modell, og presenteres i Figur 7.3. Denne fremvisningen representerer fargegruppene inneholder, til hver hoveddel. Nummereringene i figuren er kun inkludert, for å få en intuitiv referering.



Figur 7.3: Hoveddelene til hvert av GH-skriptene, med tilhørende fargegrupper

Her fremkommer det at parameterne U1, R1 og F1, og produksjonen R2 og F2 er plassert først, i hver hoveddel. Dette forenkler gjennomføring av revisjoner, ved at det ikke er nødvendig å manøvrere seg gjennom hele skriptet. De variable parameterne og produksjonen styrer skriptene. Videre blir resulterende skript, for disse to gruppene presentert i resultatkapittelet. De er avgjørende for hvordan revisjoner, på geometrisk- og beregningsmodellene, kan genereres.

Resulterende skript i de resterende fargegruppene, er å finne i GH-filene som er vedlagt i G. For beskrivelse av innhold her, er dette generelt beskrevet i metodekapitlene 6.4.3, 6.4.4 og 6.4.5. Å benytte GH-filene og beskrivelsene sammen, er dette tiltenkt tilstrekkelig informasjon, til å forstå skriptinnholdet. Disse resultatene er ekskludert i denne rapporten, da revisjon og produksjon er hovedfokuset.

### 7.3.2 Arbeidsflyten for produksjon

Resulterende arbeidsflyt, er presentert i Figur 7.4. Her fremkommer det at det ikke gjennomføres utveksling mellom CAD og FEM modeller, verken fra Rvt til Fd eller motsatt. Det produseres et utvalg av prosjekteringselementene i GH, som fremkommer i hver av programvarene. Disse elementene er presentert i de grønne boksene, som viser til hvilke prosjekteringselementer som produseres i GH. For prosjekteringselementene som må produseres i hver av programvarene, er disse inkludert i de blå boksene. Disse blir utarbeidet, etter at elementene i grønn boks er produsert.



Figur 7.4: Arbeidsflyt til den parametriske prosjekteringen

Brudekket er et av primærelementene, i både Rvt- og Fd-modellen; til hvert av GH-skriptene. Selv om dette elementet er plassert i grønn boks over, er det ikke alle Rvt-modellene som har arbeidsflyten kun i GH. Hvilken arbeidsflyt som brukes for å produsere hvert av brudekkene i de ulike modellene, er vist i Tabell 7.3. Her representerer blå farge at utarbeidelsen skjer i Rvt eller Fd, mens grønn farge indikerer utarbeidelse i GH. Her tydeliggjøres ulikhetene mellom GH-skriptene, da det er brudekket som skiller skriptene fra hverandre. Hvert skript inneholder en ny måte å definere brudekket på, både som Rvt- og Fd element. I Tabellen er det GH-skript 2.0 og 3.0 som står for produksjon av flest prosjekterings-elementer i GH, både for Rvt- og Fd-modell.

Tabell 7.3: Arbeidsflyt for produksjon av brudekker i hvert GH-skript

	1.0 GH		2.0 GH		3.0 GH	
	Rvt 1.1	Fd 1.2	Rvt 2.1	Fd 2.2	Rvt 3.1	Fd 3.2
Tverrsnittsprofil	Ny familie	Ny profil	Ny familie	Ny profil uten tverrfall	Omriss med tverrfall	Ny profil med tverrfall
Element-type	Bjelke				Direkte form	

For å ta i bruk elementer og familier i GH generelt, må disse være lastet inn i Rvt-prosjektet, som GH kjøres i. Det samme gjelder for Fd elementer, i form av at disse må være lastet inn i standard-databasen eller til en spesefik struxml-fil som er utarbeidet fra Fd.

### Produksjonen av prosjekterings-elementer i GH til Rvt og Fd

Det er mulig å holde kontroll på produksjonen av elementer i Rvt og Fd, ved at skriptene er installert med knapper. Knappene brukes til å manuelt produsere Rvt- og Fd modellene. De sender signal til «Stream Filter» komponenter i skriptene nedstrøms, som forteller at prosjekterings-elementene skal modelleres i Rvt og Fd. Knappene består både av «Button» og «Boolean Toggle» komponenter.

Hvilke komponenter som produserer Rvt-modeller, inngår i gruppene presentert i Figur 7.5. Det er blant annet mulig å produsere prosjekterings-elementene, og også muligheten til å slette allerede produserte elementer. Disse endringene skjer i sanntid i Rvt.



Figur 7.5: GH-skript for produksjon av Rvt-modellene

**R2.1 Pinne elementene:** operatøren spesifiserer at alle elementene enten skal produseres som pinnet eller ikke. False indikerer at elementene ikke er pinnet, og motsatt for true.

**R2.2 Sekundærelementer:** slette og produsere nivåene, aksene og kurvene til snittene.

**R2.3: Slette alt som er synlig i visningen:** operatøren sletter alt som er synlig i spesifikt Rvt-visning.

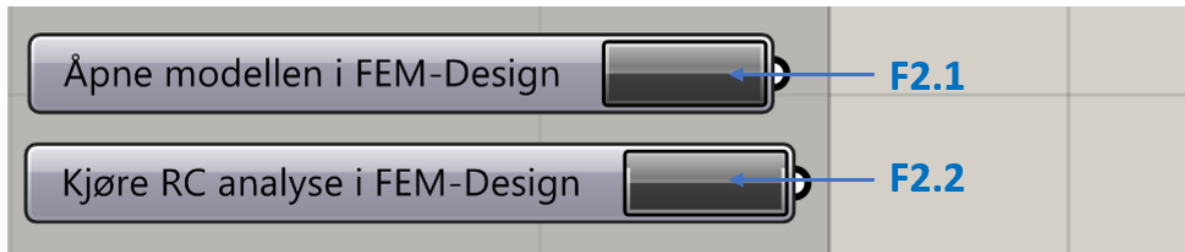
**R2.4 Brudekke som bjelke-element:** produserer brudekket som bjelke-element, samt muligheten til å slette bjelke-element/er. Skjer i henhold til organiseringsnivået type, i kategorien «Structural Framing».

**R2.5 Brudekke som direkte form:** produserer brudekker som direkte form-geometri i Rvt.

**R2.6 Armering som geometriske lukkede rør:** produsere lengde- og tverrarmring som to individuelle armeringgelementer i Rvt.

Det er ikke alle gruppene i Figur 7.5 som brukes i hver Rvt-modell. Gruppene R2.1, R2.2, R2.3, R2.4 og R2.6 gjelder for Rvt-modell 1.1. Her er rekkefølgen på produksjonen av brudekket (R2.4) avhengig av at nivåene er produsert først (R2.3). Resterende elementer er ikke avhengig av produksjonsrekkefølge. Gruppene R2.1 - R2.4 er installert i Rvt-modell 2.1, her er R2.6 neglisjert fordi denne modellen ikke produserer et gyldig brudekke og har ingen nytteverdi av å produsere armering. For Rvt-modell 3.1 inngår gruppene R2.1, R2.2, R2.3, R2.5 og R2.6, hvorav rekkefølgen på produksjonen er irrelevant.

Skriptet for produksjon av Fd-modellene, utføres med å ta i bruk de to knappene i Figur 7.6. Dette kommer av at GH ikke produserer elementene til Fd-modellene i sanntid, men i en samlet komponent; hvor elementene implementeres for så å produseres som en str-fil. Denne filen kan åpnes (F2.1) og/eller analyseres (F2.2) med operatørene i figuren under.

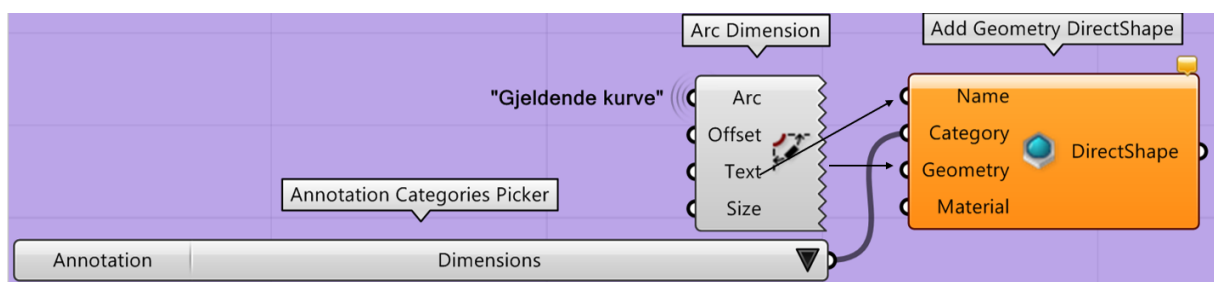


Figur 7.6: GH-skript for produksjon av Fd-modellene

### 7.3.3 Grasshopper produktene

Resulterende produkter produsert med GH, til programvarene Rvt og Fd, presenteres videre. Dette gjelder både primær- og sekundærelementene. Disse vil bli presentert i henhold til Rvt- og Fd-modellene, i de tre individuelle GH-skriptene. Resultatene henviser til innholdet av Rvt- og str-filene i Vedlegg G. Her blir alle produktene presentert i henhold til konstante variable parametere. Hvilke verdier dette innebærer, er vedlagt i Vedlegg I. Alle brudekkene består av både horisontal- og vertikalkrumning, en høydeforskjell på start og endepunkt, samt et tverrfall på 8 %. Et tverrfall på 8%, innebærer en vinkel på 4.57 grader, fra uttrykket  $\tan^{-1}(8/100)$ .

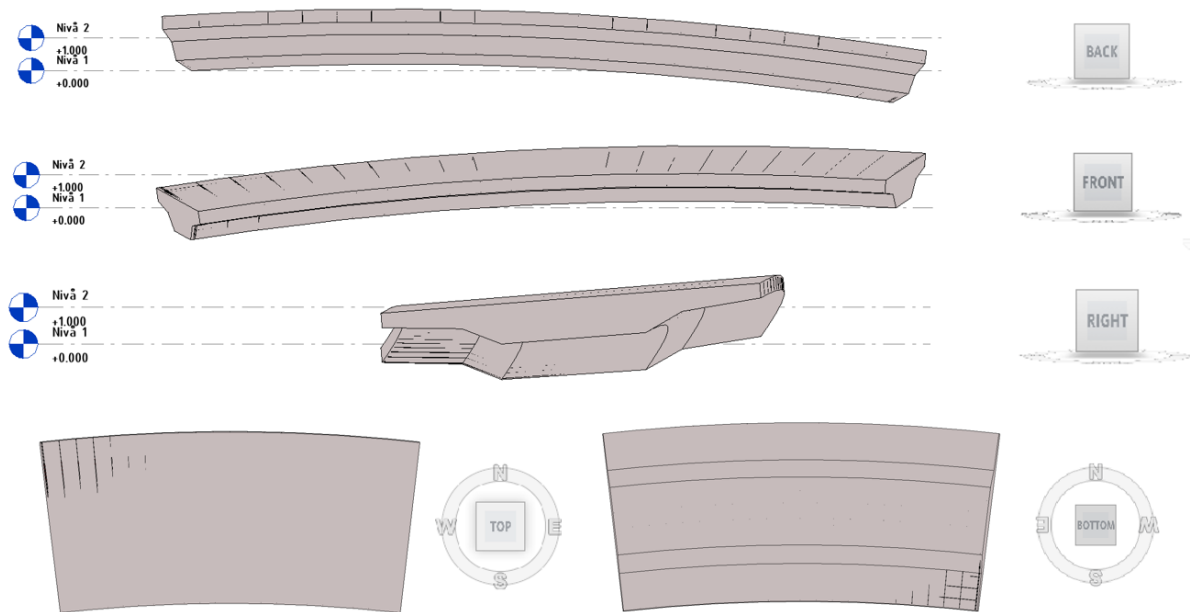
Målsetting av brudekket i både Rvt og Fd, ble ikke produsert i GH. Det er foreløpig ikke ferdigkomponerte komponenter, for produksjon av målsetting i verken Rvt eller Fd. Målsettingene må derfor modelleres i hver av programvarene, etter at produktene er produsert fra GH, dette er presentert i Figur 7.4. Metoden som er vist i Figur 7.7 under, ble ikke gjennomførbart for produksjon av målsetting. Dette var ett forsøk på å ta i bruk parametrisk målsetting i GH, med komponenten «Arc Dimension». Komponent er spesifisert til produksjon av målsetting i Rhino, som ikke har utgangsparemetere. Det skulle vært utgangsparemetere her, som skulle blitt koblet inn i *geometri* inngangen, mens teksten skulle kobles inn i *navn* inngangen til «Add Geometry DirectShape». Uten utgnagsparameterne, var det ikke mulig å finne ut om denne metoden kunne blitt brukt for å utarbeide parametrisk målsetting i Rvt-modellene.



Figur 7.7: Forsøk på å produsere Rvt-målsetting i GH, bildet er tatt fra 3.0 skriptet



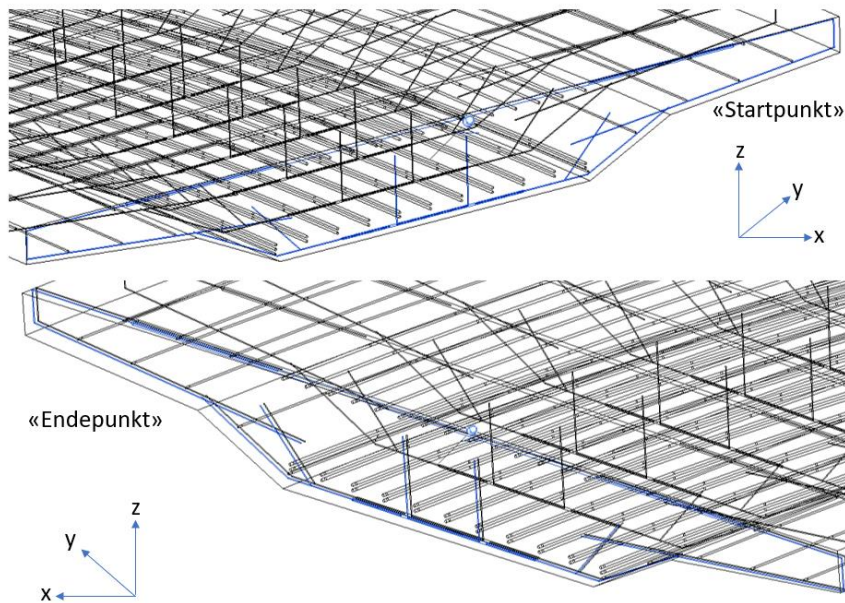
Til venstre i figuren, der tverrsnittet slutter, fremstår armeringen som feilplassert ut ifra formen på brudekket. Armeringen er vinklet feil i retningen av tverrsnittet, og i retning til høyden av det. Det betyr at bjelke-elementet og armeringen ikke samhandler, med logikken benyttet i skriptet. En bedre presentasjon på dette problemet, visualiseres i Figur 7.10. Her er spesifikasjonen på visuell stil, endret til «Consistent Colors». Her følger ikke tverrarmeringen i endepunktet tverrsnittsformen til brudekket. Deler av armeringen passert på utsiden av brudekket, som fremkommer i visningene «TOP», «RIGHT» og «FRONT». Dette er armering i nærheten av endepunktet til brudekket.



Figur 7.10: Produktene av prosjekteringselementene fra GH, RVT-modell 1.1, 3D visning

Lengdearmeringen derimot, fremstår i figuren over, at den får korrekt utstrekning. Den følger utstrekningen på brudekket, ved at ingen av armeringsjernene er synlig i start- og endepunktet. I tillegg er den ikke synlig i verken overkant- og underkant av elementet. Et annet element som fremkommer i figuren over, er at overkanten av dekket i start- og endepunkt, ikke plasseres i Nivå 1 og 2.

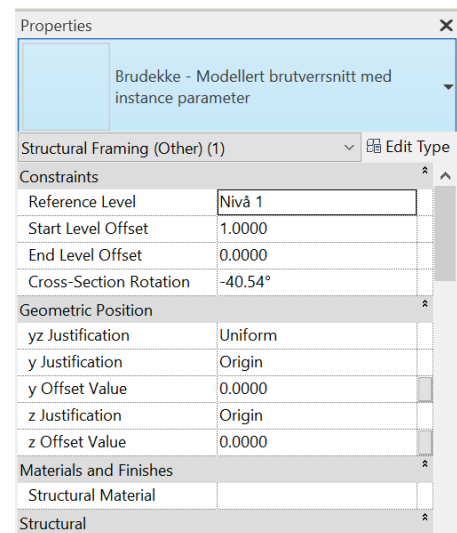
Tverrarmeringen i startpunktet er riktig i Rvt, men ved endepunktet får armeringen feil tverrfall, samt feil vinkel inn i planet til tverrsnittet. Dette fremkommer tydelig i nærbildene av start- og endepunktet til brudekket, i Figur 7.11.



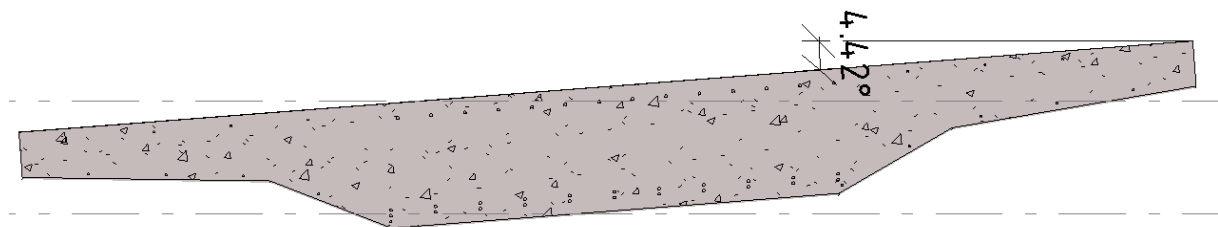
Figur 7.11: Armeringsdetalj i start- og endepunkt til brudekket

Bjelke-elementet får en analytisk kraftlinje, og denne linjen følger høydeforskjellen som er definert med nivåene. Dette fremkommer i «Egenskap paletten» i Figur 7.12, hvor «Start Level Offset» har en verdi på 1m, mens «End Level Offset» har en verdi på 0m. Dette tilsvarer høydeforskjellen definert i GH. Se Vedlegg I, for verdi på høydeforskjellen.

Også tverrsnittsrotasjonen er inkludert i «egenskap paletten». Verdien her er på  $-40.54$  grader, som består av naturlig tverrfall addert med påført tverrfall på  $4.57$  grader. Hvilket tverrfall, elementet faktisk har i Rvt, er vist i Snitt 1 i Figur 7.13. Det faktiske tverrfallet er på  $4.42$  grader i Rvt. Dette innebærer at verdien spesifisert med skriptet i GH, har en nøyaktighet på 97%.

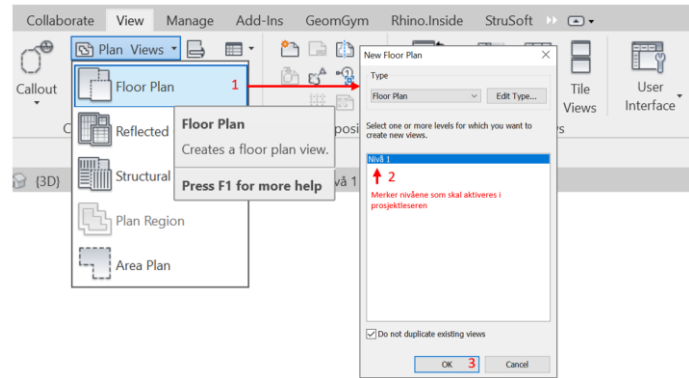


Figur 7.12: «Egenskap paletten» til bjelke-elementet



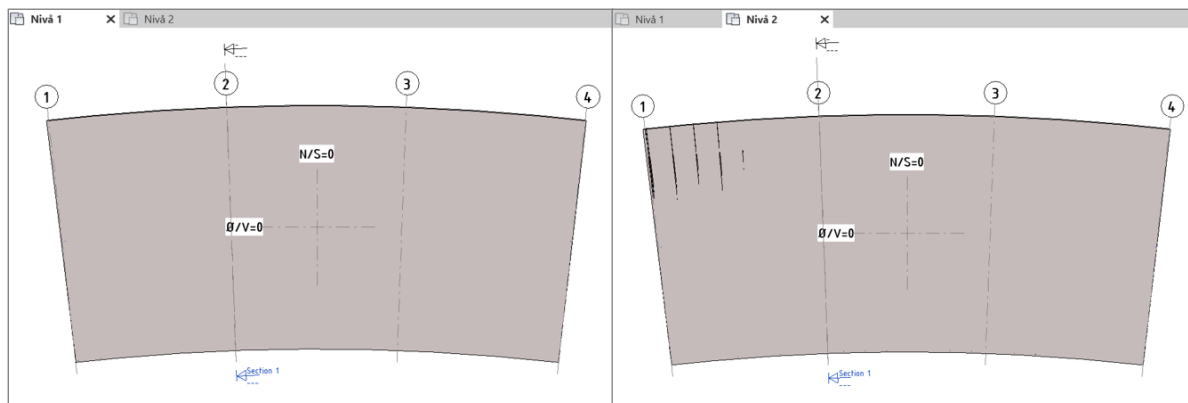
Figur 7.13: Snitt 1 i RVT-modell 1.1, med detaljnivå Fine og Visuell stil «Consistent Colors»

Produktet av nivåene, er lik i alle Rvt-modellene. Produksjonen aktiverer ikke nivåene som visninger i prosjektleseren. Dette må gjennomføres manuelt i Rvt, etter at nivåene er produsert i GH. Aktivering er nødvendig, for å kunne arbeide videre i planvisningene. Det manuelle arbeidet er beskrevet i Figur 7.14, og må gjøres for hver gang nye nivåer produseres i GH.



Figur 7.14: Aktivere nivåene som plan-visninger i Rvt

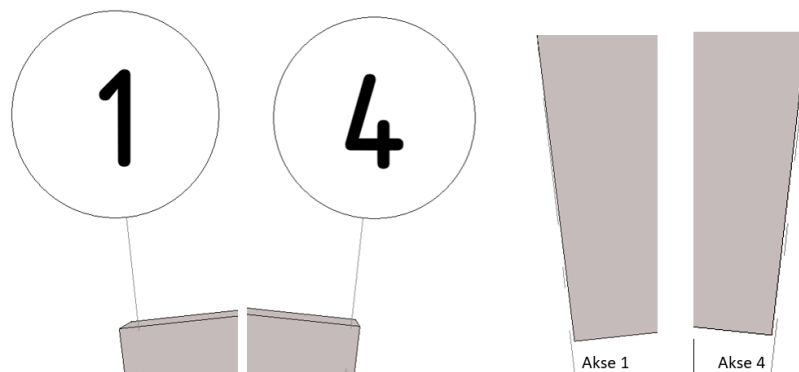
Når nivåene er definert som planvisninger i Rvt, blir resulterende planvisning til hvert av nivåene slik som vist i Figur 7.15. Her plasseres produktene av aksene fra GH, og benyttes til å spesifisere hvordan snittet skal tas. Planvisningene har visningsdybdene, som er vedlagt i Vedlegg J.



Figur 7.15: Planvisning av Nivå 1 og 2, RVT-modell 1.1, detaljnivå «Fine» og Visuell stil «Consistent Colors»

I planvisningen blir ikke snittene produsert i Rvt, med utgangspunkt i linjene produsert i GH. Når snitt-funksjonen er valgt, «snapper» ikke funksjonen seg på linjene. Derfor blir snittene utarbeidet i henhold til aksene produsert i GH. Disse får et snitt som er vinkelrett på tverrsnittet i xy-planet. Snittet er på denne måten ikke vinkelrett på tverrsnittet i yz-planet, som betyr at de ikke er vinkelrett på tverrsnittet til brudekket.

Akse 1 og 4, i Figur 7.15, følger ikke formen på brudekket i tverretning. Dette er vist på nærbildet i Figur 7.16, hvor aksene i xy-planet ikke er plassert i ytterkant av tverrsnittet i start- eller endepunkt til brudekket.

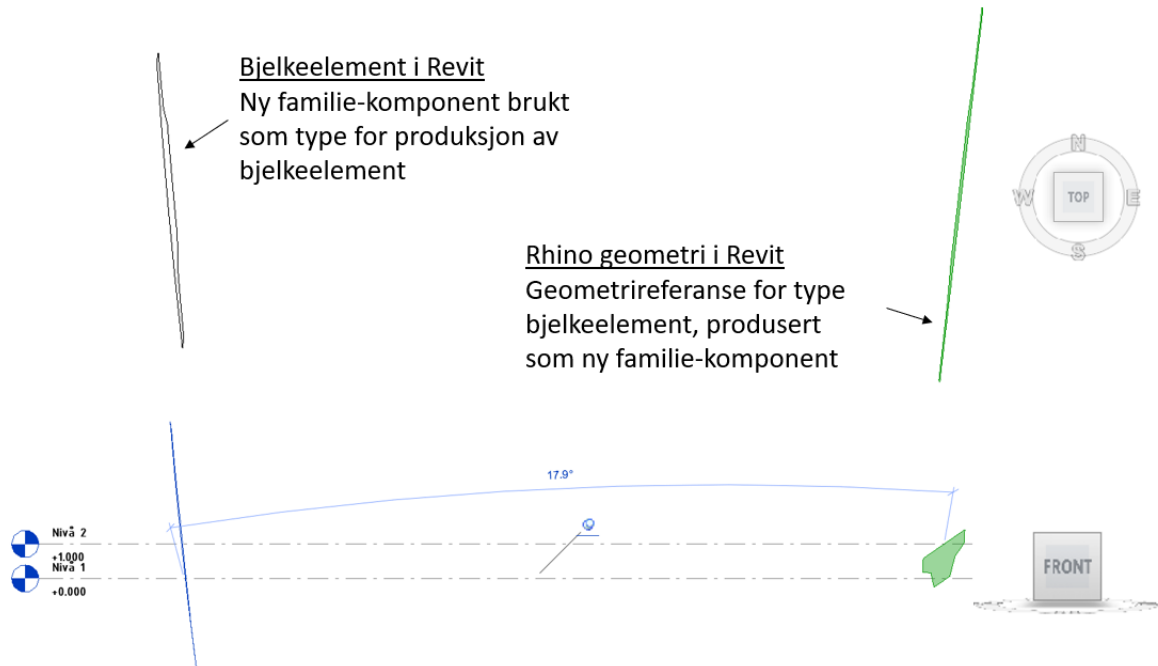


Figur 7.16: Akse 1 og 4 i henhold til brudekket

Rvt-modell 2.1 produserer brudekker som bjelke-element, hvor type brudekke er hentet fra en familie produsert i GH. Familien plasseres som en type i kategorien til bjelker. På den måten er den



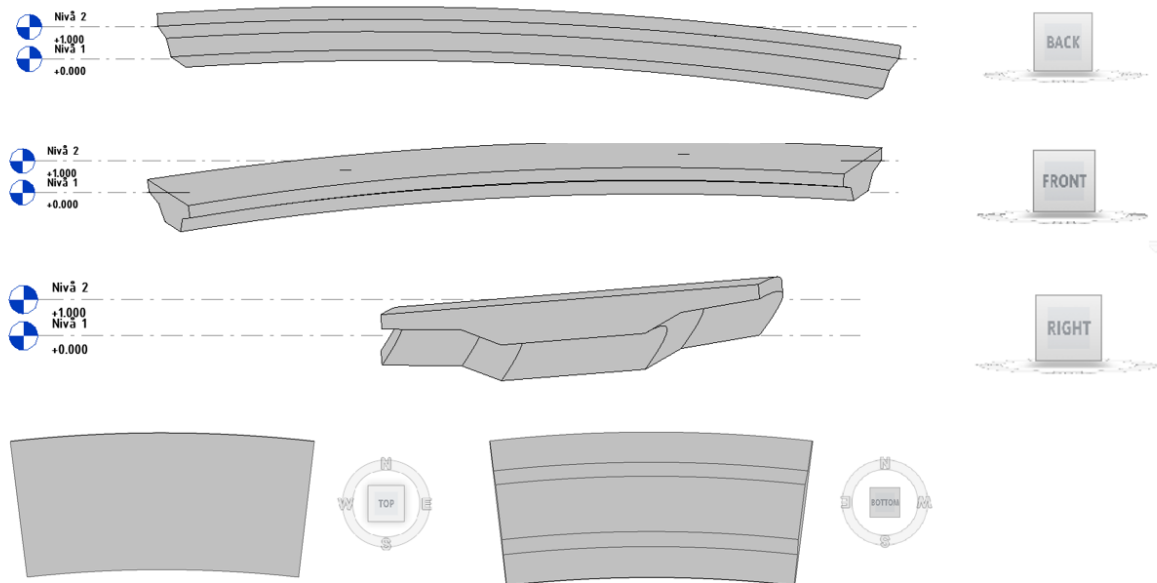
tilgjengelig til bruk i komponenten «Add Beam». Produktet av elementet er presentert i Figur 7.17, sammen med referansegeometrien som brukes til å produsere familien i GH. Produktet er hentet fra 3D visningen, med spesifikasjonene «Fine» og «Consist Colour».



Figur 7.17: Egendefinert familiekomponent i GH, til produksjon av bjelke

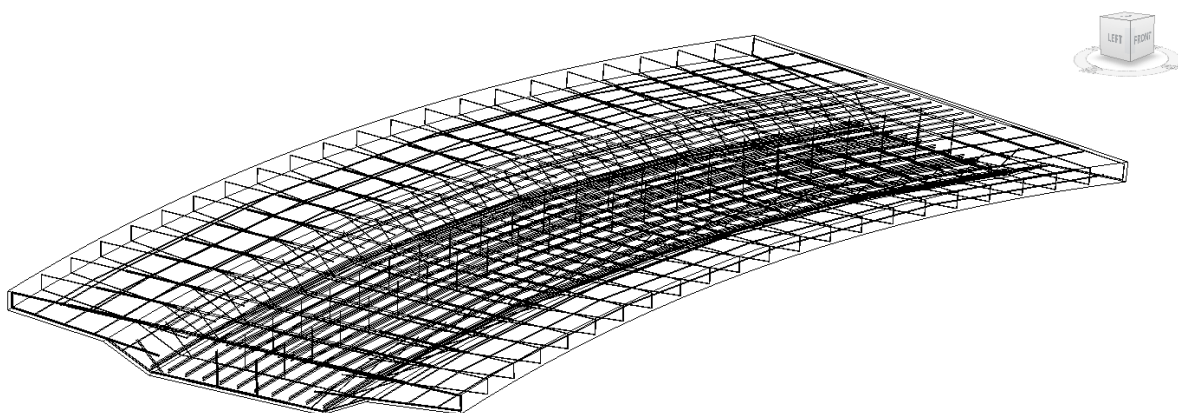
Det fremkommer i «FRONT»-visningen at bjelken består av familien, men fungerer ikke som en bjelke. Utover hva som er spesifisert i GH-skriptet, både flyttes og roteres familien brukt til bjelkene, sammenliknet med lokasjonen i Rhino. Komponentene «New Component Family» fungerer ikke for utarbeidelsen av en ny type familie, til produksjon av bjelker. Familien oppfører seg som en individuell komponent, og uavhengig kurve på utstrekning. Utstrekningen på det som definerer 17.9 i «Front» visningen, indikerer utstrekningen som familien skulle fulgt. Resulterende brudekke i Rvt-modell 2.1, er dermed et ubrukbart bjelke-element, som ikke armeres.

Rvt-modell 3.1 er det siste skriptet som produserer brudekker. Her produseres brudekker i Rvt som geometri utarbeidet fra en direkte form. I dette skriptet kan det produseres brudekker, med fire forskjellige komponentutvalg. Hvert utvalg resulterer i hvert sitt brudekke-produkt i Rvt. Alle produktene får lik avbildning i Rvt, og resultatet er Figur 7.18. I visning «FRONT», fremkommer det at midtpunktet til overkanten av tverrsnittet, i start- og endepunkt, plasseres i hvert sitt nivå.



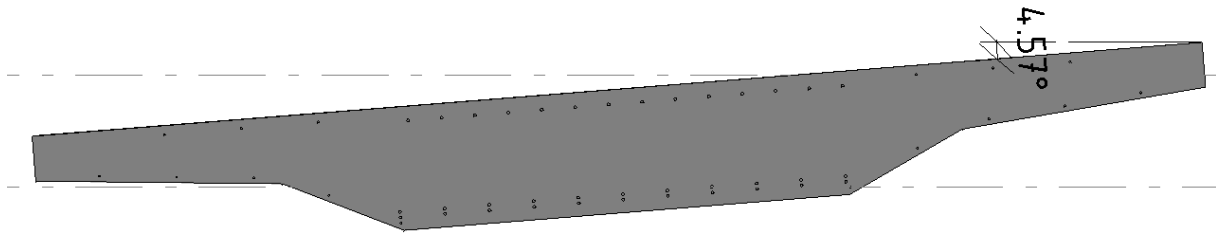
Figur 7.18: Produktene av prosjekteringselementene fra GH, RVT-modell 3.1, 3D visning

Resulterende armering i alle de fire brudekkene, er vist i Figur 7.18. Her er armeringen plassert inne i elementet, både i lengde- og tverretning. Det betyr at den både følger krumningene, tverrfallet, høydeforskjellen og riktig plassering i henhold til plasseringen av brudekket. Fremvisningene under representerer et detaljnivå som «Fine» og visuell stil som «WireFrame».



Figur 7.19: Produktene brudekke og armering i RVT-modell 3.1

Ingen av brudekkene har parameteren for tverrsnittsrotasjonen i «Egenskap paletten». Men i snitt 1 til brudekkene, fremkommer det at tverrfallet i Rvt kan målsettes til 4.57. Dette innebærer at verdien på tverrfallet i skriptet i GH, har en unøyaktighet på 100%. Her er både naturlig tverrfall og påført tverrfall inkludert i skriptet.

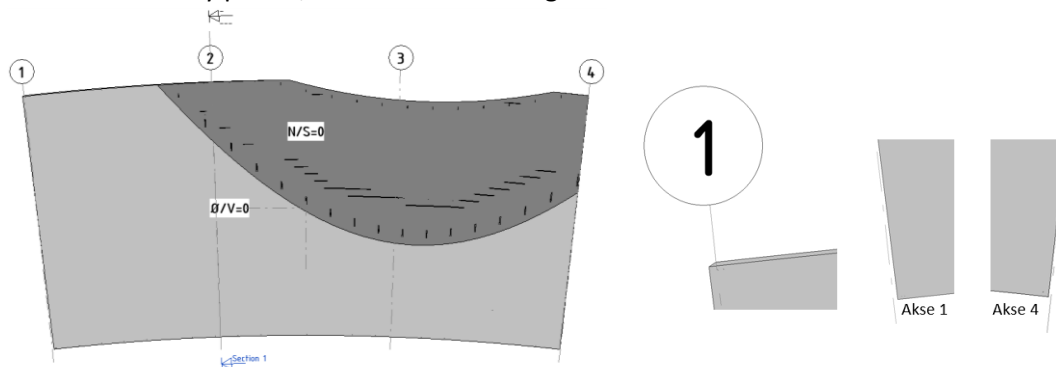


Figur 7.20: Snitt 1 i RVT-modell 3.1, med detaljnivå Fine og Visuell stil «Consistent Colors»

Snittet over, er presentert med et detaljnivå som «Fine» og visuell stil som Consist Colour». At snittet blir farget mørkegrått, betyr at produktet er et solid element.

Planvisningene må også her aktiveres i Rvt, etter at nivåene er produsert med skriptet. Hvert av komponentutvalgene får lik planvisning til Nivå 1 og 2, hvorav Nivå 2 er visuelt presentert i Figur 7.21. Her er visningsdybdene til planvisningene definert likt som for brudekket i Rvt-modell 1.1. Det gjelder for de fire produktene, i Rvt-modell 3.1. Denne informasjonen er vedlagt i Vedlegg J.

I planvisningen til Nivå 2, i Figur 7.21, fremkommer det brudekket i sin helhet ikke inkluderes. Delen av elementet som er over Nivå 2 kuttes bort. Også for aksene 1 og 4 i denne modellen, er ikke vinkelen til disse i overensstemmelse med vinkelen til brudekket. De følger ikke ytterkant av tverrsnittet, i verken start- eller endepunkt til brudekkene. Derfor har ikke avslutningen på brudekke samme vinkel i xy planet, som aksenes retning.



Figur 7.21: Unøyaktighet mellom akse 1 og 4 i henhold til avslutningen på brudekkene

Ulikhetene til de fire produktene i Rvt-modell 3.1, er å finne i de globale-parameterne, som er integrert i «Egenskap paletten». Dette er presentert i Tabell 7.4, hvor det trekkes frem hvilke typer parametere som har fått tildelt definisjoner. Produktene blir her inndelt i komponentutvalgene 3.1.1 - 3.1.4. Resulterende innhold i hver av «Egenskap palettene», er vedlagt i Vedlegg J. Tabellen inkluderer også hvilke komponenter som står bak produksjonen av elementene.

Tabell 7.4: Egenskapene til hvert av brudekkene i RVT-modell 3.1, i form av parametere og deres innhold

Egenskaper	3.1.1 «Add Geometry DirectShape»	3.1.2 «Add Brep DirectShape»	3.1.3 «Add DirectShapeType» «Add DirectShape»	3.1.4 «Component Family Form» «Add Brep DirectShape»
Kategori	Structural framing	Generic models	Structural framing	Generic models

<i>Type</i>	Ingen	ingen	Brudekke – Add Direct Shape	Ingen
<i>Referansenivå</i>	Ingen	-	Ingen	-
<i>Strukturelt materiale</i>	Ingen	-	Ingen	-
<i>Overdekning</i>	35 mm	35 mm	35 mm	35 mm
<i>Fase skapt</i>	Ny	Ny	Ny	Ny
<i>Phase demolished</i>	Ingen	Ingen	Ingen	Ingen
<i>Ekkluder fra arealberegning</i>		Avhukket		Avhukket

Komponentutvalgene plasseres i to ulike kategorier i Rvt-hierarkiet. 3.1.1 og 3.1.3, og får tildelt kategoritypen sin i GH, mens de to andre ikke har mulighet til dette. Dette resulterer i at 3.1.1 og 3.1.3 plasseres i Rvt-hierarkiet, i henhold til den tildelte kategorien «Structural framing». Det som gjelder 3.1.2 og 3.1.4, er at disse automatisk tildeles kategorien «Generic models».

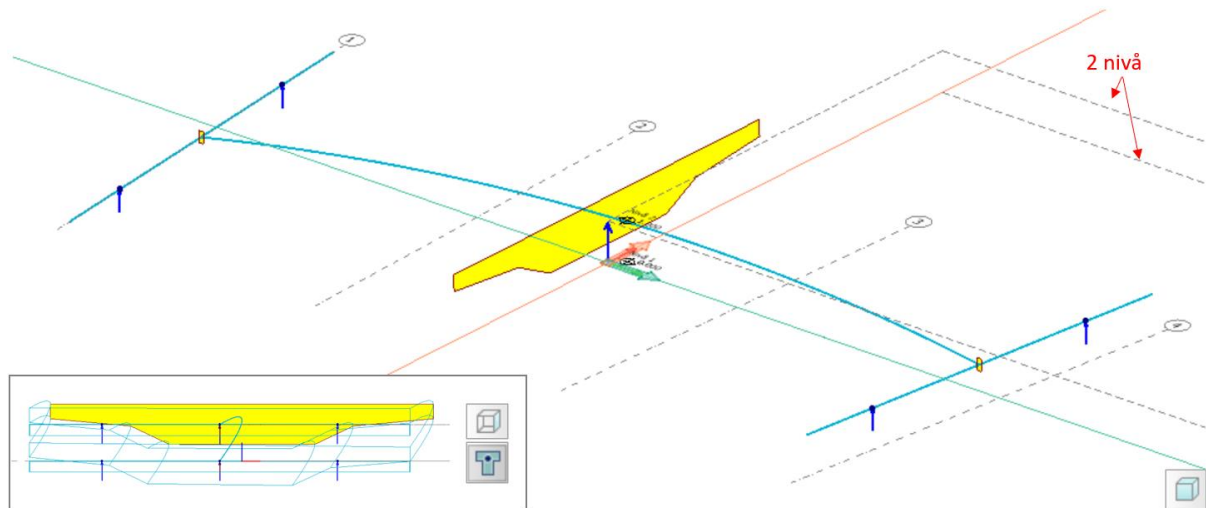
Både 3.1.1 og 3.1.3 tildeles navnet «Brudekke – Add Direct Shape», i GH. Det er kun 3.1.3 som får dette navnet inkludert i elementet, i organiseringsnivået type. Forskjellen på navngivningen i komponentutvalgene, er at 3.1.3 produserer en spesifikk type direkte form, før produksjon av selve geometrien i Rvt. Men for 3.1.1 tildeles geometrien et navn, direkte i komponenten som produserer geometri i Rvt.

For 3.1.2 og 3.1.4 har produktet til hver av komponentutvalgene identiske parameterverdier i «Egenskap paletten». Disse to produktene tilhører kategorien «Generic model», og gir ikke mulighet til å tildele produktene referansenivå eller strukturelt materiale. Både 3.1.1 og 3.1.3 har disse mulighetene.

### **FEM-Design produktene**

GH-skript produserer både brudekket og tverrbærerne som bjelke-elementer. Hvordan tverrsnittsformen og utstrekningen på tverrbæreren er spesifisert i GH, resulterer i tre ulike produkter av disse to elementene. Disse Fd-produktene, presenteres videre i samsvar med sekundærelementene: akser, nivåer, og opplegg.

GH-skript 1.0 produserer Fd-modell 1.2. Her utarbeides brudekke i henhold til egendefinert tverrsnittsprofil, produsert i Fd og hentet inn i GH. Tverrbærerne derimot er definert fra en standardisert profil i databasen til Fd. Resulterende Fd-modell i 3D visning og i fysisk vest-visning, er presentert Figur 7.22.

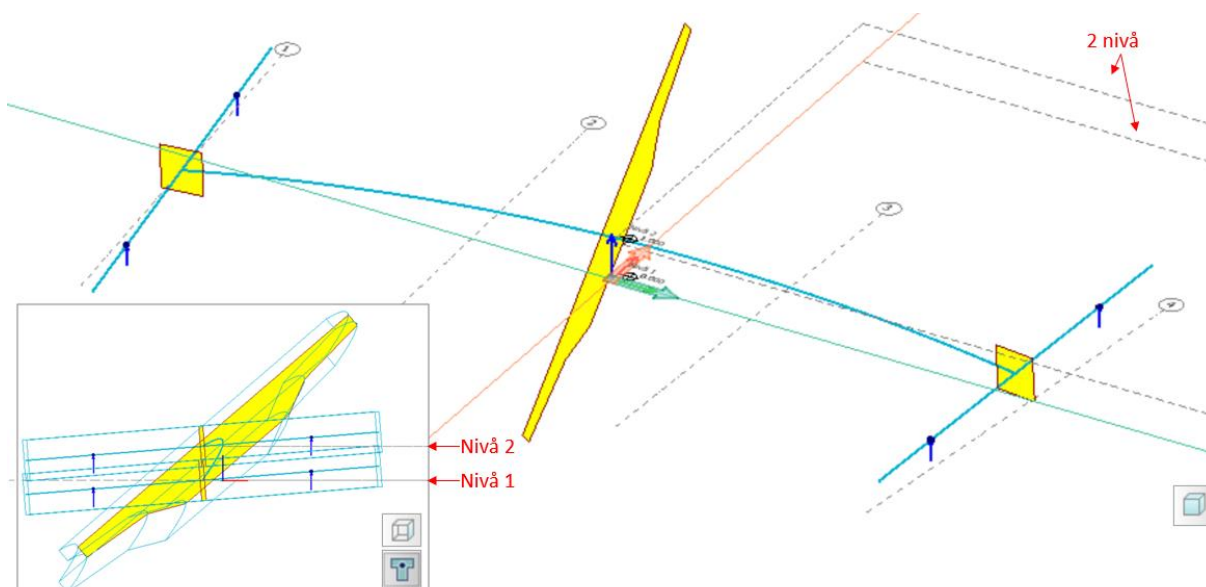


Figur 7.22: Fd-modell 1.2, 3D visning og fysisk vest-visning

Det fremkommer her at brudekket består av horisontal- og vertikal krumning, samt en høydeforskjell mellom start og endepunkt. Brudekket produseres dermed i henhold til GH skriptet. Her er ikke tverrfall tilført verken brudekket eller tverrbærerne i GH. Den fysiske vest-visningen, illustrerer at brudekket ikke får et naturlig tverrfall. Det kommer også frem at oppleggene til tverrbærerne, er plassert i kraftlinjen, vinkelrett på elementene.

Nivåene i Figur 7.22, plasseres i origo til koordinatsystemet UCS, med utstrekningen definert i GH i x- og y-retning. Her plasseres aksene i xy-planet og har lik retning som for tverrbærerne. Både nivåene og aksene, får samme resulterende produkt i de to andre Fd-modellene, som presenteres nedstrøms.

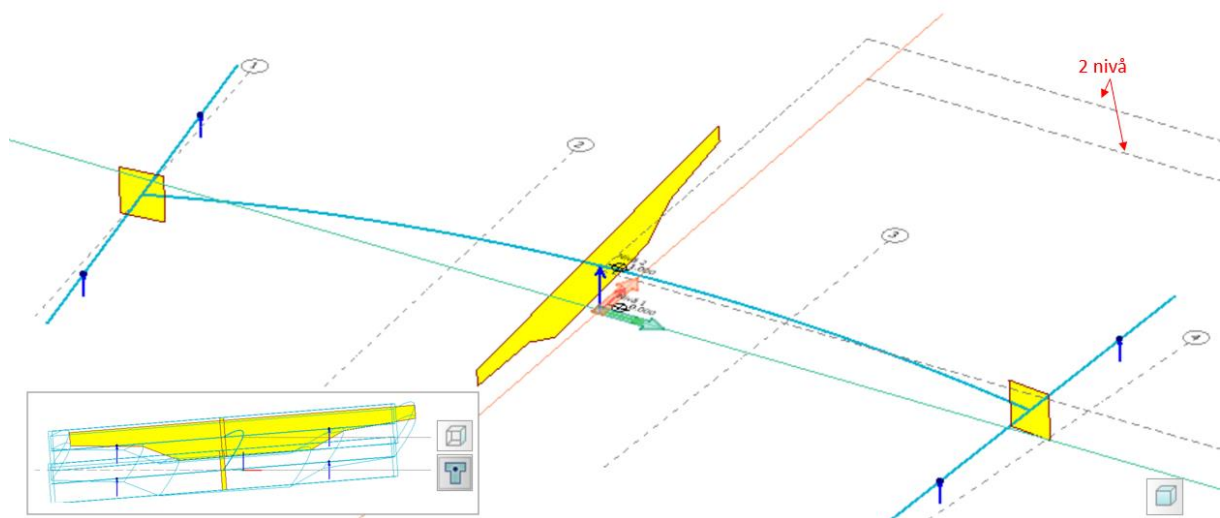
GH-skript 2.0 produserer Fd-modell 2.2. Her utarbeides brudekke i henhold til egendefinert tverrsnittprofil utarbeidet i GH, i xy-planet. Tverrsnittet roteres med påført tverrfall samt kompensering for naturlig tverrfall. Her er også tverrbærerne rotert, men kun med påført tverrfall. Tverrsnittprofilen til tverrbærerne er egendefinerte i GH. Resulterende Fd-modell av disse endringene, er presentert i 3D visning og fysisk vest-visning i Figur 7.23



Figur 7.23: Fd-modell 2.2, 3D visning og fysisk vest-visning

Figuren viser til at brudekket ikke skal modelleres i henhold til naturlig tverrfall. Dette kommer av at brudekket og tverrbærerne ikke har samme vinkel, i start- og endepunkt til brudekket. Videre modelleres oppleggene i positiv z-retning, uten av de følger rotasjonen på tverrbærerne. De står dermed ikke vinkelrett på kraftlinjen til disse.

Det siste Fd- produktet er hentet fra GH-skript 3.0, som produserer Fd-modell 3.2. Her utarbeides brudekke på samme måte som for Fd-modell 2.2, med unntak av tverrfallverdien. Brudekket roteres kun med påført tverrfall. Tverrbæreren er utarbeidet identisk som for Fd-modell 2.2. Resulterende Fd-modell i 3D visning og fysisk vest-visning, er vist i Figur 7.24.

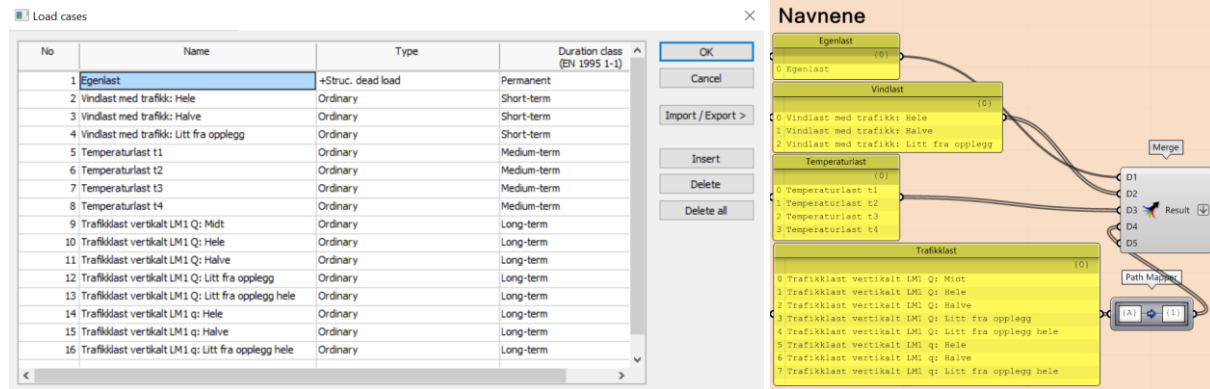


Figur 7.24: Fd-modell 3.2, 3D visning og fysisk vest-visning

Her fremkommer det at retningen på brudekkets tverrsnitt, i start- og endepunkt, er lik som for tverrbærerne. Dette resulterer i at tverrfall til et element i Fd utarbeides i henhold til påført tverrfall, uten å kompensere for naturlig tverrfall.

Hver av Fd-produktene som er presentert oppstrøms, er implementert med produksjon av laster, lasttilfeller og lastkombinasjoner. Dette er skript fra fargegruppene F4. Produktet av lastenes plassering, retningen og størrelsen påføres kraftlinjen til brudekkene. Retningen på lastene får samme retning som er spesifisert i skriptet. De utformes vinkelrett på kraftlinjen, og ikke i z-retning. Resulterende laster er vedlagt i Vedlegg K.

Både lasttilfellene og lastkombinasjonene definert i GH-skriptet overføres til Fd, etter logikken utarbeidet i skriptet. Lasttilfellene presenteres i Figur 7.25, og viser til at definerte lasttilfeller i GH, er overført i Fd etter samme rekkefølge som er utarbeidet her. Hvert lasttilfelle tildeles også riktig type og varighetsklasse.



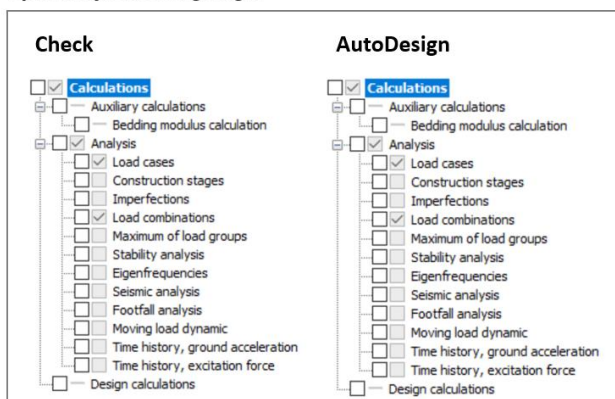
Figur 7.25: Resulterende lasttilfeller i Fd

Lastkombinasjonene er produsert med numerisk tallverdier, etter definisjonstypen til lastkombinasjonene fra v668. Resulterende lastkombinasjoner i Fd, får ikke tiltenkt rekkefølge på data i GH. Dette fremkommer i dialogen hvor alle lastkombinasjonene er listet opp, og er vedlagt i Vedlegg K. De fremvises ikke her, da det totalt er 55 lastkombinasjoner som produseres i Fd fra GH-skriptet. Utenom dette, produseres lastkombinasjonene i Fd med riktige typer, lastfaktorer og lasttilfeller.

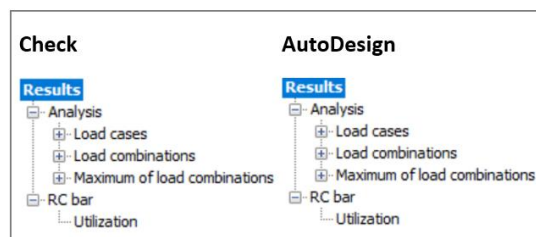
I GH er det gjennomførbart å prosjektere armering i bjelke-elementene, ved å spesifisere dette i RC analysen. Manuell armering kan ikke modelleres i GH, fordi Fd verktøyet i GH ikke inneholder ferdigkomponerte komponenter til dette formålet. I RC analysen er det mulig å velge ut beregningstypene som skal analyseres, på samme måte som gjennomføringen inne i Fd selv. De utvalgte beregningene spesifisert i GH, er «Load cases», «Load combinations» og «Design calculations». Disse analysene gjennomføres i Fd, fra GH skriptet.

Resulterende analyser og deres resultater i Fd, hvor analysen kjøres med «Check» og «AutoDesign», er presentert i Figur 7.26. Disse resultatene er hentet fra Fd-modell 3.1, men berører også Fd-modell 1.2 og 2.2. Videre blir Fd-modell 3.1 benyttet for å vise til resulterende analyse, resultat og armering, da dette produktet inkluderer flest revisjonsmuligheter.

Gjennomførte beregninger



Resultater



Figur 7.26: Resulterende beregninger og resultater for designvalgene «Check» og «AutoDesign»

Selv om ikke «Design calculation» er «hukket» av i fremvisningen av gjennomførte analyser i Fd, er denne også analysert. I begge design analysene blir elementene sjekket designmessig eller at armeringsmengden beregnes. Dette avhenger av type designanalyse som velges.

Analysene som gjennomføres med «Check» og «AutoDesign», får ulik beregningstid i de fire tilfellene, og er presentert i Tabell 7.5. Beregningstiden og merknader for uoverenstemmelser i beregningene (Warning messages), fremkommer ikke i Fd; med å gjennomføre analysene i GH. Resultatene er derfor hentet fra manuell gjennomføring av analysene i Fd. Alle analysene får opp merknader med uoverenstemmelser, i alle fire tilfellene.

Tabell 7.5 Beregningstid til analysene, med «Check» og «AutoDesign», for de 4 tilfellene

Tilfelle	1	2	3	4
AutoDesign	0,11 min	3,37 min	4,42 min	4,13 min
Check	0,12 min	0,45 min	0,48 min	0,11 min

Det er dermed mulig å hente ut resultatene til analysene av lasttilfellene, lastkombinasjonene, og design beregningen, etter at analysene er kjørt i komponentene «Application.RunDesign» i GH. For lasttilfellene og lastkombinasjonene omhandler dette brudekker bestående av horisontal- og/eller vertikalkrumning, tverrfall og høydeforskjell, samt alle utformingene av tverrbærerne. Dette inkluderer alle kombinasjons-tilfellene presentert i Tabell 7.6.

Designet til tverrbærerne i Fd-modellene 1.2-3.2 kan analyseres som både «AutoDesign» og «Check». Dermed kan de også armeres som auto- og manuell design etter at analysen er kjørt.

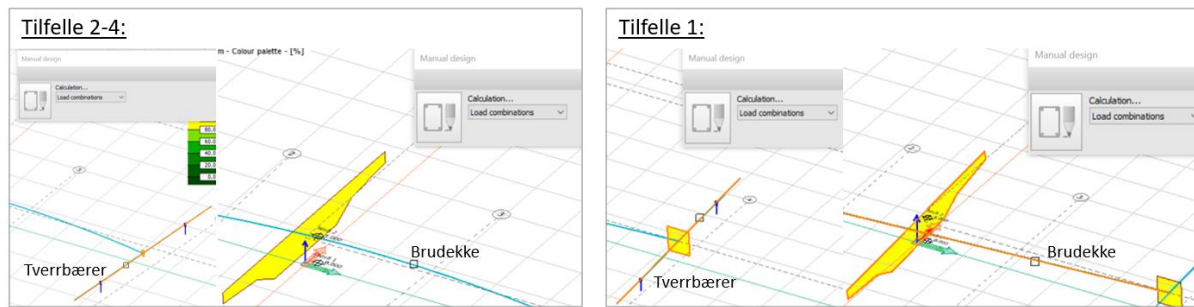
Mens for designanalysen av brudekkene, er ikke dette tilfellet. Her er den avgjørende faktoren horisontal- og/eller vertikalkrumning. Brudekkene kan kun bestå av høydeforskjell mellom start og endepunkt og tverrfall, for at elementet skal kunne analyseres når «AutoDesign» og «Check» er involvert. Dette utfallet fremkommer, når armeringsmengden skal sjekkes etter at designanalyse er kjørt. Disse er vedlagt i Vedlegg K, og presenteres i Tabell 7.6. Tabellen inkluderer både auto design som resultat av analysen, samt muligheten til å prosjektere manuell armering i bjelke-elementet.

Tabell 7.6: Utforminger på brudekket i henhold til mulighet for armering

Tilfelle	Utformingsparametere på brudekket	Auto design	Manuell design
1	Høydeforskjell + tverrfall	Ja	Ja
2	Høydeforskjell + tverrfall + horisontalkurve	Nei	Nei
3	Høydeforskjell + tverrfall + vertikalkurve	Nei	Nei
4	Høydeforskjell + tverrfall + horisontal- og vertikalkurve	Nei	Nei

Resultatene for prosjektering av manuell armering, etter at analysen er kjørt, er hentet fra metoden benyttet i Figur 7.27. Årsaken til at tilfelle 2-4 ikke kan armeres, fremkommer ved at elementene ikke er mulig å merkes, når verken auto- eller manuell design skal utarbeides i Fd. Her indikerer oransje farge på kraftlinjene, at elementet kan armeres, mens blå farge, det motsatte. For tilfelle 2-4 viser figuren til at tverrbæreren kan armeres, mens brudekket ikke kan. Mens for tilfelle 1, kan begge armeres. Dette representerer resultatene i Tabell 7.6.





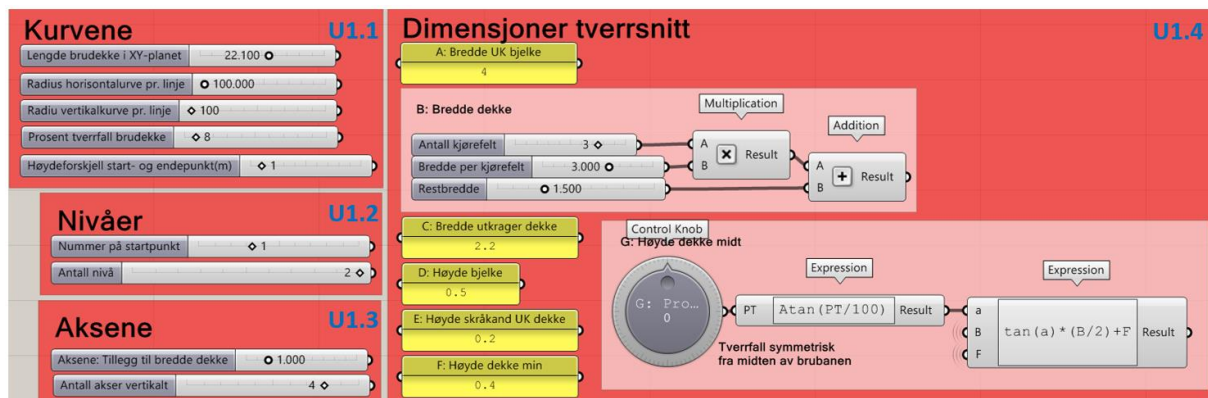
Figur 7.27: Manuell armering av tverrbærer og brudekke

### 7.3.4 Parametriske revisjonsprosess

Resulterende revisjonsprosessen inkluderer type revisjoner som er gjennomførbare, samt hvordan revisjonene utføres. Her benyttes de universelle variable parameterne (U1) for både Rvt- og Fd-modellene, samt parameterne fra gruppene R1 og F1. Siden Rvt-modell 2.1 ikke produserer et fungerende brudekke, blir denne modellen ekskludert i dette kapittelet. Og for Fd-modellene, inkluderes Fd-modell 1.2 og 3.2. Det kommer av at Fd-modell 2.2 produserte brudekket i Fd, med uoverenstemmelse med tiltenkt logikk i GH-skriptet.

### De variable parameterne

Revisjonene er styrt av de variable parameterne i skriptene. De universelle parameterne påvirker både Rvt- og Fd-modellene, både for primær og sekundærelementene. De innebærer endringsmuligheter for produksjon av kurvene, nivåene og aksene, samt dimensjonene på tverrsnittet til brudekket. Parameterne som benyttes til disse endringene, er presentert i Figur 7.28, og punktvis forklart under. Alle parameterne må tilføre verdier, med meter som enhet.



Figur 7.28: Universelle variable parameterne

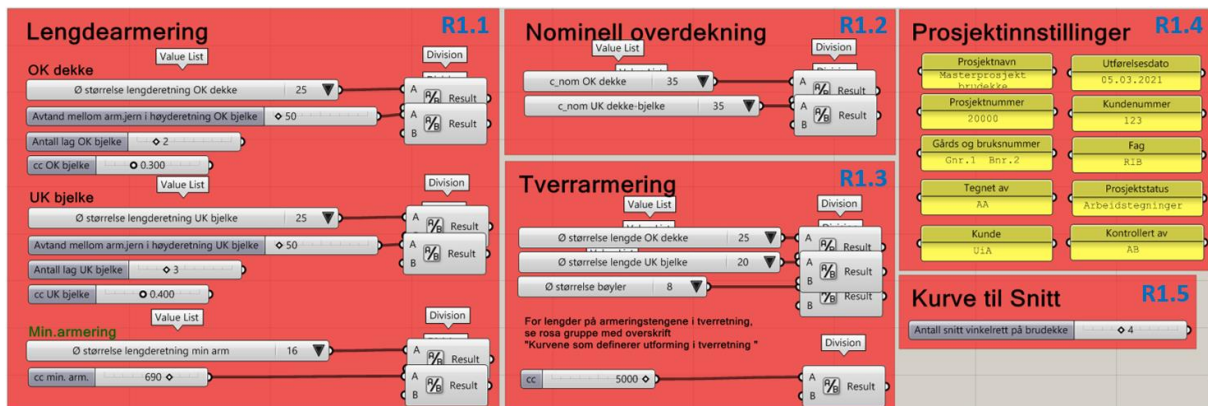
**U1.1 Kurvene:** definerer brudekkets veilinje, i henhold til lengden i xy-planet, horisontal- og vertikalkurveradius, tverrfallprosent og høydeforskjell mellom start- og endepunkt.

**U1.2 Nivåer:** spesifiserer nummeret på startnivået, og antall nivåer (med en begrensning på å bestå av én eller to nivåer).

**U1.3 Aksene:** definerer en tilleggslengde til aksens utstrekning, og antall akser vertikalt, med begrensningen på intervallmuligheten 1-6.

**U1.4 Dimensjoner tverrsnitt:** bestemmer dimensjonene på tverrsnittsprofilen til brudekket.

En samlet oversikt av R1 parameterne, er presentert i Figur 7.29 . Rvt-modell 1.1 og 1.3 inneholder parametere for produksjon av armering i tverr- og lengderetning, nominell overdekning, prosjektinnstillinger og en kurve til utarbeidelse av snitt i Rvt. Her fremkommer det at verdiene som er relatert til armeringen, divideres. Divisjonen transformere verdiene i parameterne, fra millimeter til meter. Denne operasjonen må gjøres, fordi meter er den universelle enheten som programvarene må bruke, for å få til arbeidsflyten i Figur 7.4 (kapittel 7.3.2).



The image shows a screenshot of the Rvt software interface with five main sections, each with a red header and a 'Value List' dropdown menu. Each section contains input fields and result indicators (A, B, Result).

- R1.1 Lengdearmring:** Contains three sub-sections: 'OK dekke' (Ø størrelse lengderetning OK dekke: 25, Avtand mellom arm,jern i høyderetning OK bjelke: 50, Antall lag OK bjelke: 2, cc OK bjelke: 0.300), 'UK bjelke' (Ø størrelse lengderetning UK bjelke: 25, Avtand mellom arm,jern i høyderetning UK bjelke: 50, Antall lag UK bjelke: 3, cc UK bjelke: 0.400), and 'Min. armering' (Ø størrelse lengderetning min arm: 16, cc min. arm: 690).
- R1.2 Nominell overdekning:** Contains two sub-sections: 'OK dekke' (c\_nom OK dekke: 35) and 'UK dekke' (c\_nom UK dekke-bjelke: 35).
- R1.3 Tverrarmring:** Contains three sub-sections: 'Ø størrelse lengde OK dekke' (25), 'Ø størrelse lengde UK bjelke' (20), and 'Ø størrelse bøyler' (8). A note below states: 'For lengder på armeringsstengene i tverretning, se rosa gruppe med overskrift "Kurvene som definerer utforming i tverretning"'. A 'cc' field is set to 5000.
- R1.4 Prosjektinnstillinger:** A grid of 10 yellow buttons for project settings: Prosjektnavn, Masterprosjekt, Prosjektnummer, Gårds og bruksnummer, Tegnet av, Kunde, Utførelsesdato, Kundennummer, Fag, Prosjektstatus, and Kontrollert av.
- R1.5 Kurve til snitt:** Contains one sub-section: 'Antall snitt vinkelrett på brudekke' (4).

Figur 7.29: Rvt-spesifikke variable parametere

**R1.1 Lengdearmring:** definerer  $\emptyset$ -størrelsen på armeringsstengene, avstand mellom stengene i høyderetning, antall lag og senteravstand mellom stengene (for både OK dekke og UK bjelke). For minimumsarmering, er parameterne minste  $\emptyset$ -størrelse og senteravstand parametriske.

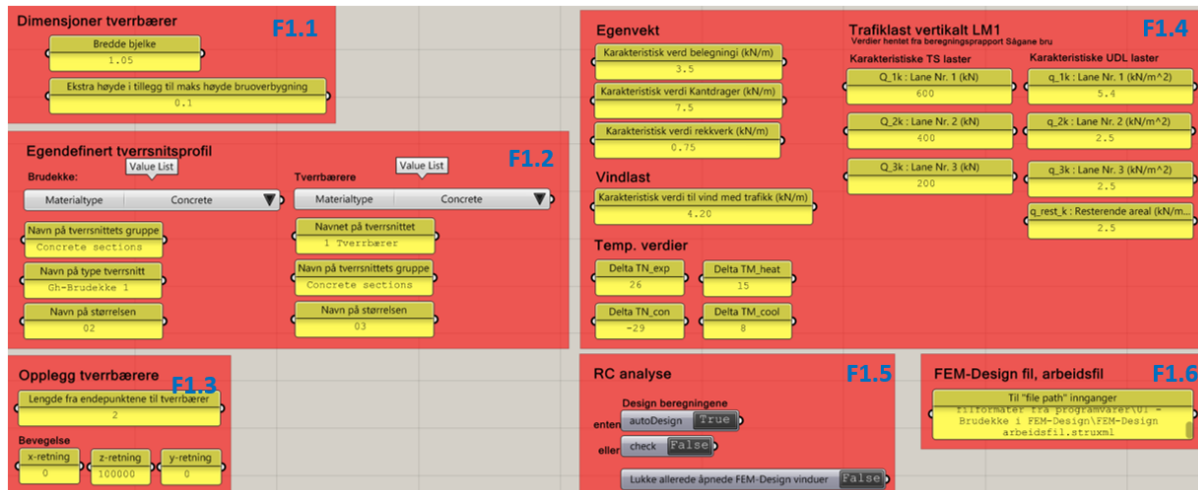
**R1.2 Nominell overdekning:** både til OK dekke og UK dekke og bjelke.

**R1.3 Tverrarmring:** definerer tre ulike  $\emptyset$ -størrelser, samt senteravstanden til tverrarmeringen.

**R1.4 Prosjektinnstillinger:** inneholder ti parametere, som kan fylles ut i henhold til prosjektet spesifikasjoner. Brukes til åpningsvinduet og arkene i Rvt.

**R1.5 Kurve til snitt:** spesifiserer antall snitt, som skal stå vinkelrett på brudekrets lengderetning. Avstanden mellom snittene er uniform.

I Fd-modellene, inngår de variable parametere både som én samlet enhet i starten av skriptet, samt enkelte variable parametere inni gruppene F3 og F4, fra Figur 7.3. Parametere som er plassert i starten av skriptet, er presentert i Figur 7.30. Hvilke parametere som inngår i hver gruppe, er listet opp under figuren. Også her, må alle verdiene som settes inn, uttrykkes i meter.



The screenshot displays the following parameter panels:

- F1.1 Dimensjoner tverrbærer:** Bredder bjelke (1.05), Ekstra høyde i tillegg til maks høyde bruoverbygning (0.1).
- F1.2 Egendefinert tverrsnittprofil:** Brudekke (Concrete), Tverrbærere (Concrete), Navn på tverrsnittets gruppe (Concrete sections), Navn på tverrsnitt (GH-Brudekke 1), Navn på størrelsen (02).
- F1.3 Opplegg tverrbærere:** Lengde fra endepunktene til tverrbærer (2), Bevegelse (x-retning: 0, z-retning: 100000, y-retning: 0).
- F1.4 Egenvekt, Trafikklast vertikalt LM1, Vindlast, Temp. verdier:** Karakteristisk verd beleggning (3.5), Karakteristisk verdi Kantrager (7.5), Karakteristisk verdi rekkverk (0.75), Karakteristisk TS laster (Q1k: 600, Q2k: 400, Q3k: 200), Karakteristiske UDL laster (q1k: 5.4, q2k: 2.5, q3k: 2.5, q\_rest\_k: 2.5), Karakteristisk verdi til vind med trafikk (4.20), Delta TN\_exp (26), Delta TN\_con (-29), Delta TM\_heat (15), Delta TM\_cool (8).
- F1.5 RC analyse:** Design beregningene (autoDesign: true), eller check (False), Lukke allerede åpne FEM-Design vinduer (False).
- F1.6 FEM-Design fil, arbeidsfil:** Til 'file path' innganger (C:\Programmer\revit\programdata\UiA - Brudekke 1\FEM-Design\FEM-Design arbeidsfil.struxml).

Figur 7.30: FEM-spesifikke variable parametere

**F1.1 Dimensjoner tverrbærer:** dimensjonene til en rektangulær tverrsnittprofil, som brukes til tverrbæreren. Denne profilen defineres med en spesifikk bredde, og en høyde som består av en tilleggsverdi til maks høyde på brudekket.

**F1.2 Egendefinert tverrsnittprofil:** til både brudekke og tverrbærerne. Her er det mulig å velge materialtype, i nedtrekksmenyer. Resterende parametere må egendefineres skriftlig.

**F1.3 Opplegg tverrbærere:** hvor oppleggene til hver av tverrbærerne skal plasseres, i henhold til lengden fra ytterpunktene. Dette skjer ut ifra begrensningen om at hver tverrbærer er opplagret på to punkter. Disse er også symmetrisk plassert, i lengderetning.

**F1.4 Lastverdier:** karakteristiske verdier for lasttilfellene egenvekt, vindlast, temperaturlast og trafikklast vertikalt for LM1.

**F1.5 RC analyse:** definerer om design beregningen skal beregnes som auto design eller som en sjekk av alle elementene. I tillegg er det her en parameter som gir brukeren mulighet til å bestemme om allerede åpne vinduer i Fd skal lukkes, når analysen gjennomføres.

**F1.6 Arbeidsfil:** skriftlig valg av stien på arbeidsfilen, som Fd-modellen og analysen skal utarbeides i. Her defineres stien på intern server, og må også inneholde navnet på filtypen. Dette er et krav for å konvertere data fra GH til en beregningsmodell i Fd.

Parameterne som ikke er plassert i starten av hovedgruppen Fd-modell, er gruppert i samme rødfarge og plassert hos tilhørende fargegruppe. Dette gjelder for parametere som er avhengig av å stå sammen med sin spesifikke arbeidsoppgave. Dette gjør det enklere å forstå hvordan parameteren skal endres. Dette gjelder lastplasseringen, lasttilfellene og lastkombinasjonene (i F4), fordi disse er sammensatte og ofte særegne for bruprosjektering. I tillegg er spesifikasjonene for kryp- og svinnverdi, samt navn på bjelke-element variable parametere plassert i F3.

## Revit modellene

Resulterende revisjonsmuligheter i GH-skript 1.0 og 3.0 til RVT-modellering, er presentert i Tabell 7.7. Her er linjene som produseres for å modellere snitt direkte i Rvt neglisjert, da denne metoden ikke var gjennomførbar etter tiltenkt logikk (presentert i kapittel 7.3.3). Ut ifra tabellen er det mulig å gjennomføre åtte typer revisjoner, hvorav den niende revisjonstypen, materialtype, ikke er mulig.

Tabell 7.7: Revisjonsmuligheter i GH-skript for RVT-modellene

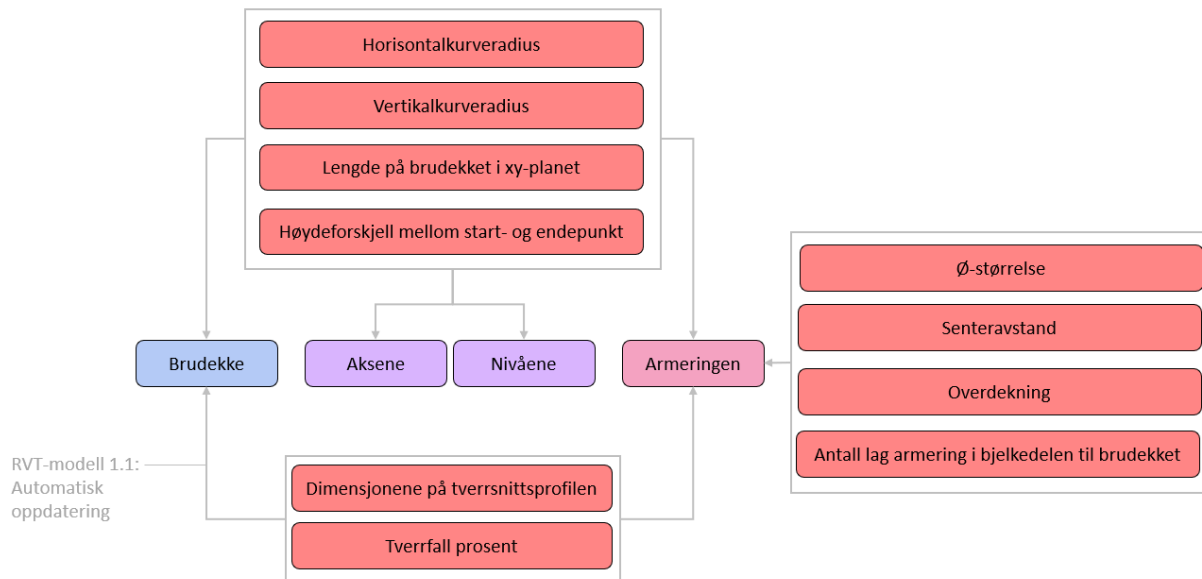
Revisjonsmuligheter	RVT-modell 1.1	RVT-modell 3.1
Lengde på brudekket i xy planet	Ja	Ja
Horisontalkurve	Ja	Ja
Vertikalkurve	Ja	Ja
Tverrfall	Ja	Ja
Dimensjonene til tverrsnittprofil	ja	Ja
Høyden på nivåene	Ja*	
Antall akser	Ja	
Prosjektinnstillinger	Ja	
Materialtype	Nei	Nei
Armering i brudekket - $\phi$ -størrelse - senteravstand - overdekning - antall lag armering i bjelkedelen av brudekket	Ja	

\* Antall parametriske nivåer er begrenset til maks to.

Begge Rvt-modellene får de samme resulterende revisjonsmuligheter. Men hvordan disse revisjonene gjennomføres, er ulik. Ett av de resulterende utfallene, er at elementer som allerede er produsert; oppdateres i henhold til nytt innhold. Det andre utfallet er, at allerede produserte elementer må slettes, for at nytt innhold skal kunne benyttes til produksjon.

Det første utfallet som er nevnt over, gjelder både for data i parameterne til prosjektinnstillingene, og bjelke-elementet som representerer brudekket i Rvt-modell 1.1. Det kommer av at forekomstparameterne til elementene overstyrer i GH, og resulterer da i automatisk oppdatering i sanntid. Forekomstparameterne til bjelke-elementet gir mulighet for at tverrfall og dimensjoner til tverrsnittprofil overstyrer, som en automatisk oppdatering, se Figur 7.31. Denne metoden er også brukt til å overstyre materialtypen, men er ikke gjennomførbart. Det betyr at revisjon av materialtype, må skje direkte i Rvt.

Når de resterende parameterne i GH-skript 1.0 for Rvt-modell 1.1 tildeles nytt datainnhold, må prosjekteringselementene som påvirkes av endringen, først slettes, og deretter produseres på nytt. Parameterne som forårsaker dette utfallet, er illustrert i Figur 7.31. Pilene uten merknad, viser til hvilke revisjoner som medfører at elementene må slettes, før import av nye reviderte elementer. For sekundærelementene nivå og akser, er det mulig å slette disse direkte i GH-skriptet. Men for brudekket og armeringen, må disse slettes i Rvt.



Figur 7.31: Hvordan revisjonene til RVT-modell 1.1 og 1.3 skal gjennomføres

Utfall to, innebærer alle elementene i Rvt-modell 3.1, foruten prosjektinnstillingene. Dette er det eneste elementet som oppdateres automatisk i sanntid. Det betyr at de fire prosjekteringselementene i Figur 7.31, må slettes før produksjonen av reviderte elementer. GH-skript 3.0 produserer brudekket i sin helhet, før det omgjøres til et Rvt element. Dette elementet får ikke dimensjonsparametere i Rvt, hvilket betyr at forekomstparametere ikke kan overstyres her. Også her kan nivå og akser, slettes i GH-skriptet, mens brudekket og armeringen slettes i Rvt.

### FEM-Design modellene

Resulterende revisjonsmuligheter i GH-skript 1.0 og 3.0 til Fd-modellering, er presentert i Tabell 7.7. Her neglisjeres presentasjonen av resulterende revisjonsmuligheter til Fd-modell 2.2, da brudekket i dette skriptet får et ukorrekt tverrfall. Ut ifra tabellen er det mulig å gjennomføre fjorten typer revisjoner i Fd-modell 1.2, mens tolv revisjoner med GH i FD-modell 3.2.

Tabell 7.8: Revisjonsmuligheter i GH-skript for Fd-modellene

Revisjonsmuligheter	Fd-modell 1.2	Fd-modell 3.2
Horisontalkurve	Ja	Ja
Vertikalkurve	Ja	Ja
Tverrfall	Nei	Ja
Dimensjonene til tverrsnittprofil	Nei	Ja
Høyden på nivåene	Ja*	
Antall akser	Ja	
Materialtype	Ja, i overenstemmelse med standard-databasen til Fd.	
Kryp og svinn verdier	Ja	
Statiske laster - Antall punktlaster på kraftlinjen - Lastverdi	Ja, med utstrekningsbegrensningene i lengderetning	
	Ja	

Lasttilfeller i hver lasttilfellegruppe	Ja, med at det kan tilføres lasttilfeller i gruppene egenlast, temperaturlast, vindlast og trafikklast vertikalt LM1
Lastkombinasjonene - Lastfaktorene - Type	Ja Ja
Type design beregning	Ja, enten «Check» eller «AutoDesign»

\* Antall parametriske nivåer er begrenset til maks to.

Revisjonsmulighetene som fremkommer i tabellen, viser til endringer som medfører produksjon av en ny Fd-modell, hver gang datainnholdet overføres til Fd. Dette er beskrevet i kapittel 7.3.3. Modellene har like revisjonsmuligheter, foruten tverrfall og dimensjonene til tverrsnittprofilen. Dette omhandler Fd-modell 1.2, hvor revisjoner av tverrsnittsformen utarbeides i Fd, mens tverrsnittsfallet ikke er mulig å revidere. Det er ikke er funksjoner i Fd, som gjennomfører denne utformingsmuligheten.

## 8 Diskusjon

I dette kapitlet blir resultatene for de tre metodene i kapittel 6 diskutert. Her diskuteres både resultatene fra litteraturstudiet og erfaringer fra grenseindustrien mekatronikk, under metoden parametrisk prosjektering. Det legges det vekt på arbeidsflyt, produkter, revisjonsmuligheter, og svakheter med valgt metode og case.

### 8.1 Litteraturstudiet

Resultatene fra litteraturstudiet vil ikke diskuteres innad, men i samsvar med resultatene av den parametriske prosjekteringen. Dette fremkommer i kapitlet 8.2.

#### 8.1.1 Svakheter ved valgt metode

I litteraturstudiet ble det tatt et valg om utelukkende å sette søkelys på masteroppgaver. Slike oppgaver inneholder arbeid utarbeidet av studenter. Å utelate andre litteraturtyper, kan ha hatt konsekvenser på at relevant litteratur ikke ble inkludert. Dette kan derfor ha en innvirkning på påliteligheten av resultatene, som er presentert i dette prosjektet. Grunnen til at søkeomfanget ikke ble satt bredere, kom av at det ble satt av mer tid til den parametriske prosjekteringen. Dette viste seg å være nødvendig, da det akkurat var nok tid til å ferdigstille resultatene.

Det ble det inkludert fire masteroppgaver, da disse var av relevans i henhold til begrensningene som ble satt i litteratursøkene. Forskingen som ble utført med tilhørende konklusjoner, ble det satt full tillit til. I etterpåklokskap burde det også her blitt satt søkelys på svakhetene med forskningen, og eventuelt undersøkt metodene dypere. Dette kan ha ført til at resultatene presentert fra deres oppgaver, ville fått en annerledes vinkling.

De inkluderte oppgavene, ble i tillegg ansett som pålitelige, og relevante for problemstillingen. Hvordan informasjonen ble hentet ut og presentert i resultatene, kan likevel ha vært overskygget av egen overbevisningen av pålitelighet og relevans. Dette kan ha blitt en konsekvens i fremstillingen i resultatkapitlet 7.2. Dette er en svakhet som er vanskelig å unngå alene. Sett i ettertid, kunne denne svakheten blitt redusert, hvis flere personer hadde blitt involvert i gjennomgangen av masteroppgavene, slik at disse kunne blitt diskutert for å definere en endelig forståelse av innholdet.

### 8.2 Parametrisk prosjektering

Det er erfart at det foreløpig er nødvendig å ha bedre kunnskap om tekstlig koding, for å ha mulighet til å innlemme flere revisjonsmuligheter, samt produksjon av flere av prosjekterings-elementer, i de resulterende skripene. Dette er en antydning som tas i henhold til tilgjengelige Rvt- og Fd verktøy i GH. Dette gjelder for elementene armering og målsetting, i begge modellene. Målet i oppgaven var å unngå tekstlig koding, men det viser seg at det ikke er mulig ennå.

Det ble satt av en uke til å undersøke om det var mulig å ta i bruk Python-skript, men jeg fikk kun til et enkelt skript. Dette ble ikke implementert i resulterende skript, da det ikke påvirket muligheten til å produsere Rvt-armering med GH. Årsaken til at det ikke ble satt av mer tid, kom av at dette fagfeltet var så mye mer krevende enn først antatt. Det var derfor ikke kapasitet til å bruke mer av

tiden, som var avsatt til oppgaven, på dette området. Til tross for dette, var det en positiv opplevelse å få til det ene fungerende skriptet.

På den ene siden, kan det virke som om tekstlig koding er nøkkelen til å utnytte potensialet av parametrisk prosjektering i GH. Det kommer av at det i opplæringsprosessen ble kjent at det er mulig å produsere armering i Rvt, i henhold til organiseringsnivået til strukturell armering, mens det foreløpig ikke er tilgjengelige verktøy i GH til dette formålet. Tekstlig koding ligger dermed foran utviklingen av tilgjengelige GH-verktøy.

I både forprosjektet og i denne masteroppgaven er det erfart at overgangen fra tradisjonell modellering i Rvt og Fd, til visuell parametrisk modellering i GH, er et enklere skritt som ingeniørstudent, sammenliknet med å ta i bruk tekstlig koding. Denne erfaringen kommer av det mislykkede forsøket med å utarbeide en Python-komponent, for produksjon av Rvt-armering. Dette kan komme av at ingeniører ikke er utdannet til å ta i bruk matematikken og forståelsen som trengs, for å produsere algoritmer i form av tekst. De er for så vidt ikke utdannet til å benytte visuell koding heller, men grensesnittet mellom tradisjonell- og visuell parametrisk modellering, oppleves som relativt enkelt å sette seg inn i. Det har jeg selv erfart det siste året. En annen fordel jeg erfarte ved utarbeidelsen av Python-skript, er at en slik kunnskap gir grunnlag til at brukeren arbeider mer effektivt, og intuitivt i Rvt.

På den andre siden fremkommer det at utviklingen innenfor programvarer, kontinuerlig oppdateres og forbedres. Spesifikt for verktøyet RiR, da det bare i løpet av tidsperioden av dette prosjektet, ble lansert to nye oppdateringer av verktøyet. Derfor er det et håp om at det om ikke lenge, vil komme ferdigkomponerte komponenter til produksjon av henholdsvis armering og målsetting i GH, både til Rvt- og Fd- modellene. Spesielt gunstig er det for armering i prosjekteringsprosessen, da dette er et helt nødvendig element i all RIB prosjektering. Dette ville effektivisere revisjonsprosessen ytterligere, sammenliknet med resulterende skript i dette prosjektet. Siden dette ikke er tilfellet i skriptene, vurderes den parametriske prosjekteringen i dette prosjektet, til å kun være gunstig i tidligfasen. Men hadde disse to elementene vært mulig å produsere og inkludere i revisjonsprosessen i GH, kunne parametrisk prosjektering blitt vurdert som effektiviserende for hele prosjekteringsprosessen.

Skulle dette ikke være tilfellet, så vil en opplæring av tekstkoding med hovedfokus på bygg-ingeniører, være en mulig løsning. At ingeniører burde få tilbudet om videreutdanning, innenfor dette fagfeltet. Dette kan etter min mening, utvide bruksområdet av parametrisk design, og ta det til nye høyder.

### 8.2.1 Beregningsproduktene

Ved å produsere og revidere både Rvt- og Fd-modellene fra GH, er GH en fordel i seg selv ved at det representerer bruksanvisningen til dem begge. GH innholdet er skript som til enhver tid kan være synlig. Dette skriptet er i seg selv en intuitiv faktor i en prosjekteringsprosess. Det er en fordel å ha et tilgjengelig og synlig skript, fordi det presenterer tankegangene til hvert element, samt sammenhengen mellom dem. Informasjonen er ikke skjult bak elementene, ved at bare det visuelle er presentert, slik som det er i både Rvt og Fd. Skriptene representerer på en måte bruksanvisningen til modellene.



En annen fordel, er at feilsøking i modellene blir ansett som enklere å gjennomføre når sammenkoblingene er synlig, slik som de er i GH. Feilsøking blir også enklere å gjennomføre når komponentene er systematisert i grupper, slik som de resulterende skriptene. Ligger problemet eksempelvis i brudekket-produktet, kan feilsøkingen starte i denne gruppen, og enklere kunne identifisere problemet, i tilhørende gruppe eller andre elementer som har påvirkning på produktet.

### **Buet brudekke**

Utformingsmulighetene i den parametriske prosjekteringen, har vært strukturell interessant i Fd, grunnet mulighet til å produsere et bjelke-element med irregulære former. Det kommer av at selv om utformingen er mulig og nøyaktig, gir det nødvendigvis ikke nytteverdi til å prosjektere armering i elementet. Et brudekke bestående av horisontal- og/eller vertikalkrumninger gir ikke mulighet for å armere elementet. Det ekskluderer muligheten til å innlemme et høyst nødvendig element i beregningsproduktet. Dette utfallet kommer til tross for at elementet i henhold til lasttilfellene og lastkombinasjonene blir analysert. Her har ikke de strukturelle lastene, som følger kraftlinjen på brudekket, problemer med den buede utforming.

Dette kan indikere at hovedproblemet med et buet bjelke-elementet, kommer av tilgjengelige standardiserte beregningsmetoder i Fd. Dette er fordi Fd ikke inneholder standardiserte løsninger, for prosjektering av buet armering. Hvis et bjelke-element med horisontal- og/eller vertikalkurve kan defineres som en ustandardisert statisk løsning. Da kan bjelke-element med høydeforskjell og tverrfall spesifiseres som standardisert løsning i Fd, da utformingsdesignet kan armeres både i RC analysen og i etterkant av analysen.

At et buet bjelke-elementet ikke er statisk løsbart, tydeliggjør også viktigheten av arbeidet som kreves av en rådgivende bygg-ingeniør. Beregningsmodeller i Fd har behov for utarbeidelse av idealisering, i henhold til ingeniørfaglig kunnskap og erfaringer, for å skape en abstrakt modell av virkeligheten. Grunnen til dette er at det i teorikapittelet fremkommer at bruk av FEM i beregningsprogrammer både krever idealisering og modellering. Viktigheten av idealiseringen tydeliggjøres i det buede bjelke-element i dette prosjektet. Dette er kun modellert og ikke idealisert. Ut over dette, tydeliggjøres fallgruven, som kan komme av å stole blindt på programvarens beregningsmetodikk til armering, i dette elementet.

En mulig løsning på problemet kan hentes fra konklusjonen i Master 1. I denne masteren ble en krummet veilinje oppdelt i mindre rette linjer. Årsaken til at dette ble gjort, var at det analytiske elementet i Rvt, ble importert i SOFiSTiK. For at dette skulle være gjennomførbart, måtte brudekket deles opp i flere mindre linjer, for at det skulle bli et analytisk element i Rvt. Et analytisk Rvt-element ble godkjent og fungerende i SOFiSTiK, til grunnlag for beregninger. Denne problemstillingen er den samme som for utveksling av Rvt-modell til Fd.

Denne likheten ved at utveksling av modeller fra Rvt til SOFiSTiK og Fd har samme krav, er det tiltenkt at en inndeling av et buet bjelke-element kan løses på lik linje, som i Master 1. Problemet kan ligge i at et buet elementet i GH-skriptet, ikke er utformet analytisk riktig. Her kunne «gjeldende kurve» blitt delt opp med punkter, og rette linjer kunne blitt utarbeidet mellom dem. Linjene kunne så blitt brukt i inngangene, til komponentene som produserer bjelke-elementene i Fd. Dette kunne kanskje

løst problemet med å gi mulighet til å armere et buet bjelke-element. Det er også mulig at denne endringen, ville ført til at feilmeldingene om at modellene har likevektsproblemer og for store deformasjoner, ville forsvunnet.

### **Rvt-armering og målsetting**

Den største ulempen i alle GH-skriptene, er begrensningene som følger med verktøyene til Rvt- og Fd i GH til produksjon av målsetting og armering. Derfor er det foreløpig ikke tilstrekkelig ferdigkomponerte komponenter, til å utarbeide et skript som er egnet i hele prosjekteringsprosessen. Disse elementene er nødvendig å kunne følge brudekket parametrisert, for at skriptene skal regnes som effektiviserende i alle fasene av prosjekteringsprosessen. Dette kommer av at revisjonene i skriptene, innebærer i stor grad at elementene må slettes for flere av dem.

Det at armeringen er utarbeidet med komponenten «AddDirectShape.ByGeometry», har som ulempe at den ikke inkluderer type armeringsstål (ø-størrelsen) som en globalparameter i Rvt. Heller ikke type armeringsjern, definert etter spesifikke nummer (00,12 ...), er inkludert i denne komponenten. Dette gjør at endeforankringen ikke kan spesifiseres med samme utarbeidelse som i Rvt selv. Her har heller ikke «rørene» som utgjør bøyene, «glidende» overganger mellom skjøtene. Dette kunne vært unngått om armeringen hadde vært spesifisert i henhold til type armeringsjern, da dette er forhåndsbestemt i denne parameteren.

Det som er fordelen med at armeringen ble utarbeidet på denne måten i Rvt, er at potensialet av en parametrisert armering, synliggjøres. Armering utarbeidet i GH opprettholder grensebetingelsene til brudekket, som armeringen tilhører. Dette igjen, viser til at produktet forbedrer revisjonsprosessen ved at parametere, som gjelder fysisk ø-størrelse, senteravstand og nominell overdekning, kan tildeles nye verdier som oppdaterer både lengde- og tverrarmeringen i forhold til hverandre. Grunnen til dette, er at det er utarbeidet en parametrisert avhengighet, som opprettholdes i henhold til disse parameterne.

### **Tverrfall**

Ved å sammenlikne resulterende buet brudekke fra Rvt-modell 1.1 og 3.1 med Fd-modellene, fremkommer det at definisjonen av tverrfall er ulik i Rvt og Fd. Tverrfallet til bjelke-element i Fd, modelleres ikke i henhold til naturlig tverrfall grunnet kurvaturen, slik som i Rvt. Det fremkommer i bjelke-elementene hvor tverrsnittsprofilen enten er utarbeidet i egendefinert Fd-mal, eller i GH.

I førstnevnte tverrsnittsprofil, er tverretningen til overkanten av brudekket parallell med y-retningen. Dette resultatet sammen med sistnevnte tverrsnittsprofil, både i Fd-modell 2.2 og 3.2, underbygger at slutningen er et buede betongbjelke-elementer, som ikke utformes med naturlig tverrfall fra kurvaturen på buen. I Fd-modell 2.2 inkluderer rotasjonen av tverrsnittet kompensering for naturlig tverrfall på brudekket, mens Fd-modell 3.2 inkluderer kun påført tverrfall. I begge modellene er tverrbærerne kun utformet med påført tverrfall. I henhold til resulterende brudekkeelementer sammen med tverrbærerne, viser til at brudekke i Fd-modell 3.2 får reelt tverrfall. Dette er også brukt til å underbygge slutningen.

### **Laster**

Resulterende produkter av lastplasseringen, lasttilfellene og lastkombinasjonene er produsert i henhold til en rekke avgrensninger. Disse avgrensningene innebærer at skriptet for beregningsmodellen må videreutvikles, for at skriptet skal kunne tas i bruk i en reel prosjektering av et brudekke. Hadde det vært mer tid, ville også lasttilfellene til lastmodellene hos vertikal trafikklast, samt horisontal trafikklast, blitt implementert i skriptet. Men det at det faktisk er vist til at de inkluderte lastene er mulig å produsere og revidere i GH, er nok til å kunne besvare hvilke elementer som kan produseres med GH. Oppsettet i skriptene, både i form av type komponenter og logikk, kan tas i bruk for å inkludere disse lasttilfellene. Disse må også integreres i skriptene til lastene og lastkombinasjonene.

En annen forbedring av Fd-produktet lasttilfeller, kunne vært å inkludere lastgrupper i GH, i tillegg til definisjonene av lasttilfellene. Lastgrupper er en egen funksjon i Fd, som viste seg å også være tilgjengelig i Fd-verktøyene i GH. Dette ble kjent litt for sent i prosessen av den parametriske prosjekteringen i GH, til at det var tid til å prøve å implementere denne funksjonen. Lasttilfellene ble inndelt i grupper i «Panel» komponenter, hvor lasttilfellene ble definert. Her er det tiltenkt at disse gruppene blir erstattet av lastgrupper, slik at de også ville blitt produkter i Fd. En annen måte lastgruppene kunne blitt implementert, er ved at plasseringen av lastene som ble inkludert i definisjonene til lasttilfellene, ville blitt separert med lastgrupper i stedet.

### **Familie opprettelse**

Forsøket med å utarbeide en ny familie i GH, med formål om å erstatte den egendefinerte familien produsert i Rvt og brukt i Rvt-modell 1.1, ble ikke vellykket i Rvt-modell 1.2. Målet her var å unngå å måtte utføre arbeid i Rvt, til bruk i GH-skriptet. Familien kunne dessverre ikke benyttes til å modellere et solid bjelke-element. Siden familien var utarbeidet til å plasseres i lik kategori og mal-fil, var det et håp om at dette skulle være tilstrekkelig til å få det til å fungere. Likevel var det tvil, fordi komponenten spesifiserte at den utarbeider en ny familie-komponent i Rvt. En Rvt-komponent er et eget element, og kunne dermed ikke brukes i organiseringsnivået type; for produksjon av et solid bjelke-element.

Da dette ikke var gjennomførbart etter tiltenkt ønske, blir resulterende brudekke i Rvt-modell 1.3, en løsning på å implementere dette arbeidet i GH-skriptet. Dette brudekket er fullt ut modellert i GH, som en geometrisk lukket «Sweep», og konvertert til et Rvt element med fire ulike komponentutvalgene. Produktene av disse fire, fremvises i Rvt-visningene som like, både i henhold til fargen på omrisset, snittet og plasseringen i forhold til armeringen. Dette er til tross for at det i Tabell 7.4, viser til at utvelgelse 3.1.2 og 3.1.4 verken inneholder parametere for å definere referansenivå, eller strukturelt materiale. Dette er parametere som påvirker produktets plassering og utseende. Her er det ikke funnet en logisk begrunnelse på resultatene. Men 3.1.1 og 3.1.3 inneholder disse parametere. Derfor blir disse ansett som mest egnet som produkt i prosjekteringsprosessen.

### **Målsetting og snitt**

Det oppstår et problem for både målsetting og snitt i Rvt, når et brudekke består av både horisontal- og vertikalkurve, horisontalkurve og høydeforskjell, vertikalkurve og høydeforskjell eller kun vertikalkurve. Disse parameterne kompliserer utarbeidelsen av å få riktig målsetting. Det kommer av at elementet målsettes i 2D visningen i Rvt, som ikke får med seg variasjonene av form-endringer i

rommet. Samme problemstilling gjelder for Fd modellen, da målsettingen er begrenset til å modelleres i 2D, selv om den kan utformes i 3D visning.

Samme problemområde fremkommer i resulterende snitt på brudekket i Rvt-modellene. Disse må også modelleres i 2D visningene, og får dermed ikke et snitt som står vinkelrett på brudekket, i tverretningen. Siden det ikke ble funnet en parametrisert løsning på disse to elementene i GH, var det heller ikke mulig å undersøke hvordan dette kunne blitt løst her. Men med tanke på at de geometriske sammenhengene og matematiske forholdene gav mulighet til å produsere de andre elementene i Rvt parametrisert, er det tenkt at også snitt og målsetting kunne blitt implementert om Python-komponenter kunne blitt utarbeidet, eller at det var tilgjengelige Rvt-komponenter til produksjon av disse.

### 8.2.2 Revisjonsprosessen

Revisjonsprosessen som inngår i arbeidsflyten er effektiv, ved at man unngår å manuelt endre elementene i samsvar med hverandre. Elementene som er reviderbare med GH-skriptene, har en gjensidig avhengighet, slik at elementene utarbeides i henhold til hverandre. Her opprettholdes forholdene og avhengighetene som er bestemt i skriptet. Til tross for dette, er det en stor ulempe at de fleste elementene, må slettes for å produsere reviderte modeller. Mer om disse fordelene og ulempene, diskuteres nedstrøms.

Det at strukturell armering og målsetting i Rvt og Fd må utføres i hvert sitt program, etter at produktene og eventuelle revisjoner er blitt foretatt i GH-skriptene, er nødvendig for å oppnå et tilstrekkelig innhold i geometrisk- og statisk modeller. Årsaken til dette er at elementer er nødvendige til bruk som tegnings- og beregningsprodukter, hos DS. Det at disse elementene ikke er parametrisert knyttet til produktene som utarbeides i skriptene, reduserer effektiviteten av å benytte GH-skriptene, til utførelse av revisjoner på modellene. Selv om det er mulig å produsere 3D rør som armering i Rvt, og beregne armeringsbehovet til rette bjelke-elementer i Fd, er ikke dette en optimal løsning til bruk i prosjekteringsprosessen. Derfor kan parameterne som er tilgjengelige i skriptene for revisjon av armeringens utforming,  $\emptyset$ -størrelsen, senteravstanden eller nominell overdekning, ekskluderes som nyttige endringsmuligheter i resulterende skript.

Det at en reel armering som strukturelt element i Rvt ikke var oppnåelig i skriptene, har sin negative påvirkning på at de ikke regnes som egnet til bruk i alle fasene av prosjekteringsprosessen. Siden elementene produsert i GH må slettes, for så å produsere nye elementer etter at revisjoner i parameterne er utført, innebærer dette at både armering som kan modelleres som «Free form» i Rvt og målsettingen, slettes. Dette ødelegger muligheten for å gi en effektivisert revisjonsprosess, spesielt med tanke på bruk i detalj- og slutfasen av et prosjekt. Det kommer av at tegningsproduktet i disse fasene skal bestå av detaljer, som er presentert i Figur 5.26 i casekapittelet. Disse blir ikke ivaretatt i GH-skriptene.

Det at GH-skriptene er utarbeidet med revisjonsmulighetene i form av å tildele nytt innhold i parametere, er fordelaktig med tanke på å ha muligheten til å benytte skriptene som mal-filer for andre bruprosjekter, som skal prosjektere brukonstruksjoner. Det kommer av at

revisjonsmulighetene som er mulige å foreta i skriptene, ansees som intuitive, til tross for begrensningene som fremkommer i dette diskusjonskapittelet.

### **Materiale**

Å definere materialtype i Rvt og Fd innebærer ulike grad av spesifikasjoner. I Rvt spesifiseres utseende på materialet, mens det i Fd defineres materialegenskaper. Resulterende brudekke-element i alle Rvt-modellene viste seg å verken kunne ha mulighet til å revidere materialtype, eller utseende. Men alle bjelke-elementene i Fd modellene, gir mulighet til å velge materiale i henhold til eksisterende materialer i databasen til Fd. Det er også mulig å definere kryp- og svinn-egenskaper til valgt materialtype. Dette innebærer at det bare er bjelke-elementene i Fd som oppnår målet om revisjonsmuligheter til materialet.

Grunnen til at dette var oppnåelig, var at komponentene som ble brukt til dette, fungerte. Dette var ikke tilfellet for komponentene som ble brukt til å gi revisjonsmuligheter av materialtype i Rvt. Det viste seg at selv om materialet ble spesifisert til brudekket i Rvt-modell 1.1 og 3.1 i GH, ble dette ikke oppdatert eller overført til elementet i Rvt. Det fremkommer ved å se på resulterende innhold i parameteren «Structural Material», til hvert av elementenes «Egenskap palett».

For brudekket utarbeidet som bjelke-element, var ikke dette et overaskende resultat. Det kom av at det allerede i skriptet, fremkom en feilmelding om at denne parameteren ikke kunne overstyres fra GH. Men for elementene som var produsert som direkte form, og tildelt materialtypen som inngangsdata, var resultatet overraskende. Selv om disse produktene hadde «Structural material» i «Egenskap paletten», så fikk ikke denne parameteren tildelt materialet som var spesifisert i GH.

### **Lastplassering, lasttilfeller og lastkombinasjoner**

I resultatkapittelet 7.3.4 for Fd-modellene, fremkommer det at revisjonene av lasttilfellene, lastplasseringen og lastkombinasjonene, skjer inni tilhørende gruppe. De er ikke plassert i starten av Fd-modell skriptet, og parameterne burde dermed defineres som modifikasjoner fremfor revisjoner. Årsaken til dette, er at skriptene til disse gruppene er avhengig av hverandre uten fullstendig automatisering. Det krever mer av brukeren som skal benytte skriptet, ved at det må opparbeides en forståelse av oppbygningen og logikken til disse gruppene. Dette er ikke tilfellet for parameterne plassert i starten. Disse er mer selvforklarende, og trenger ikke å settes i kontekst med skriptgruppe som muliggjør endringene.

Revisjonsmulighetene til lastplasseringene er i hovedsak kun styrt av antall punktlaster. Disse er låst i henhold til utstrekningene som er definert i skriptet, i brudekkets lengderetning. Også linjelastene er låst til en spesifikk utstrekning, og kan ikke revideres. Disse utstrekningene kunne vært hensiktsmessig å holde åpne for revisjon, da de kan variere for å oppnå mest mulig ugunstig effekt på elementet. Dette ble ikke utarbeidet, på grunn av tidsmessige årsaker. En måte som kunne vært av interesse å undersøke her, er relatert til metoden som er benyttet i Master 2. I denne oppgaven oppnådde de å bruke Karamba inne i GH, for å gjennomføre form-generering til å finne verste lastplassering. Hvis resulterende lastplasseringer fra dette, kunne blitt brukt som referanse data til produksjon av lastene i Fd, kunne kanskje dette gitt mulighet til en bedre revisjonsprosess.

Det at lastfaktorene til lastkombinasjonene kan revideres i GH, innebærer at antall operasjoner som kan brukes for å revidere alle lastkombinasjonene er redusert. Dette er sammenliknet med samme arbeid utført direkte i Fd. Hvis det skulle være slik at alle lastfaktorene i hver av lastkombinasjonstypene fra v668 endres, ville dette innebære å endre verdiene til fem parametere, i hver av de totalt elleve lastkombinasjonstypene. Skulle disse endringene derimot blitt gjennomført i Fd, innebærer dette at 55 operasjoner av endringer, manuelt måtte blitt utarbeidet (Se Vedlegg K). Dessuten er det lettere å unngå at feil verdi tildeles lasttilfellene, da antall operasjoner er redusert i GH. Her er også lastfaktorene lettere å holde oversikt på, da de er systematisert og plassert i tilhørende lastkombinasjonstype. Dette er faktorer som underbygger at det er mulig å generere en mer effektiv revisjonsprosess, ved å parametrisere lastkombinasjonene, slik det fremkommer i resulterende skript.

Det kan også tenkes at bedre kunnskap om «Tree» komponentene og de boolske operasjonene i GH, kunne medført en bedre parametrisering av lastplasseringen, lasttilfellene og lastkombinasjonene. Trærne kan endre strukturen til innholdet i komponentene. Dette gir mulighet til å få tilgang til spesifikk data, og skjer ved å endre rekkefølge, identifisere og/eller sortere måten treet er bygd opp. Kunnskapen om bruk av «Tree»-komponenter ble utviklet underveis i den parametriske prosjekteringen. Det var først etter at størsteparten av innholdet i skriptene var ferdigutviklet, at denne tanken inntraff. Da var det ikke tid nok, til å undersøke dette dypere.

Til tross for at det er forbedringspotensialet i hvordan revisjonene kan gjennomføres, blir både parametriseringen av lastplassering, lasttilfellene og lastkombinasjonene vurdert til å kunne effektivisere revisjonsprosessen. En generell faktor som påvirket denne vurderingen, er at et skript tydeliggjør sammenhengene av produksjonen og påvirkningen av elementenes innhold, sammenliknet med de separate dialogene som må benyttes i Fd.

### 8.2.3 Arbeidsflyt

I teorikapitlet presenteres fire ulike samhandlinger av CAD og FEM utveksling, prinsipielt. Dette omhandler å implementere og eksportere struXML filer mellom Rvt- og Fd programmene, ved hjelp av tilleggsprogrammet StruXML Revit Add-In. Denne arbeidsflyten rasjonalisere byggingeniørarbeidet. Dette er også tilfellet med arbeidsflyten i resulterende skript. Men her unngår arbeidsflyten utveksling mellom CAD og FEM-modellene, direkte. Modellene er derimot utarbeidet ved at store deler av elementene er samlet på en plattform, nemlig GH. Hvor elementene som ikke produseres her, skjer direkte i hver av programmene, også her uten utveksling mellom CAD og FEM. Dette er presentert i resultatkapittel 7.3.2.

GH skaper dermed en intuitiv arbeidsflyt til elementene, som faktisk kan produseres her. Mens det som må modelleres i henhold til disse produktene, i hver av programvarene, ansees ikke som intuitivt. En av ulempene med arbeidsflyten, er at Fd-modellen ikke inkluderer etterbehandlingen som er inkludert i Figur 3.7 i teorikapitlet 3.3. I henhold til figuren inkluder skriptene både CAD-geometri og analyse, og unngår at CAD-geometrien må forbehandles for å fungere som FEM-modell. Her sees det bort i fra problematikken med et buet produkt av bjelke-elementer, belyst over. Det betyr at arbeidet med forbehandling neglisjeres i denne arbeidsflyten.

Samhandlingspraksisene av CAD og FEM, beskrevet i teorikapittelet, er det som tas i bruk av mekatronikingeniørene. Det fremkommer i resultatene at de aktivt utarbeider CAD modeller som preppes og konverteres til Step-filer, før import til FEM-programmer. Denne praksisen er lik StruXML Revit Add-In tillegget, til samme arbeidsflyt for bygg-ingeniørene som importerer Rvt-modell (CAD) inn i Fd, etter at filen er konvertert til struXML-filer. Det å preppe en CAD-fil i prosjekteringen av komponenter for mekatronikingeniørene, er tidkrevende. Og det er også et ønske om å få en arbeidsflyt som unngår dette arbeidet, slik som arbeidsflyten er benyttet i dette prosjektet.

En annen måte som parametrisk design kan brukes på, som tilsvarer arbeidsflyten i avsnittet over, kommer fra arbeidsflyten benyttet i Master 1. Her var det utarbeidet en arbeidsflyt mellom Rvt- og SOFiSTiK programmet ved å benytte det parametriske verktøyet Dynamo. Her ble ikke modellene utarbeidet i Dynamo, slik som det er gjort i GH i dette prosjektet. Den tok utgangspunkt i å produsere en analytisk modell i Rvt, som ble importert til SOFiSTiK, på lik linje som med samhandlingen mellom Rvt og Fd er beskrevet i teorikapittelet.

Problemet som fremkom med denne arbeidsflyten, var når brudekket var buet. Da måtte som nevnt tidligere, elementet deles opp for å få en analytisk modell i Rvt. Denne oppdelingen resulterte i at Rvt-modellen innebar begrensinger på tegningsproduktet, som ikke var optimale. Hvis et buet element i GH-skriptene hadde blitt tilført den samme oppdelingen, ville skriptet hatt som fordel at dette ikke ville påvirket tegningsproduktene i Rvt. Arbeidsflyten separerer modellene, og gir mulighet til å få både optimale tegnings- og beregningsprodukter.

Grunnen til at resulterende produkt fra både Fd-modell 3.2 og Rvt-modell 1.1 og 3.1 kan separeres, kommer av utarbeidelsen av en referansegeometri i den universelle gruppen i GH-skriptene. Disse benyttes til begge modell-produksjonene. Dette bedrer blant annet muligheten til å eliminere risikoen for avvik mellom CAD- og FEM-modellene, når produksjonen og revisjonene utarbeides. Her sees det bort i fra at brudekke med krumninger ikke kan armeres i Fd, og at oppdeling av dette elementet løser dette problemet.

Ulempen med denne parametriske prosjekteringen, kommer av at størsteparten av innholdet i geometrisk- og beregningsmodellen, må produseres på nytt når revisjonene gjennomføres. Dette innebærer at elementer som ikke er parametrisert i GH, slik som målsetting, armering og eventuelle snitt i Rvt, også må slettes og produseres på nytt. Så lenge ikke alle elementene er parametrisert i GH, kan det være at det blir mer tidkrevende å benytte skriptene i hele prosjekteringsprosessen, fremfor tidsbesparende. Derfor er skriptene best egnet i tidligfasen av prosjekteringen, da denne fasen innebærer mindre detaljer og beregningsoptimaliseringer.

#### 8.2.4 Svakheter ved valgt metode/case

Det ble ikke startet med blanke ark, i starten av masteroppgaven. Jeg hadde en tanke om temaet som ble valgt, og hva som var av ønske å finne ut i dette prosjektet. Forkunnskaper og holdninger, kan derfor ha virket inn på tolkning og bearbeidelse av dataen. Det kommer av at jeg ble «instrumentet», som fanget opp og analyserte skriptene og produktene. Hvilke forutsetninger som ble satt, ble avgjørende for dataen som fremkom. Her er en svakhet at en verdinøytralisering kan ha

vært påvirket av målet med oppgaven. Dette har ikke vært av intensjon, men kan ha vært et underliggende instinkt, som har vært med på å forme vurderingen av skriptene og produktene.

Her kan også den objektive versus subjektive oppfatningen spilt en rolle. Målet har vært å være objektiv i prosessen av både utarbeidelsen av skriptene, og resultatene som er presentert. Men anvendelsen av den kvalitative metoden som er benyttet, innebar at jeg har vært i direkte møte med produktet som bidrar med data. Dette kan ha vært en svakhet i dette prosjektet, ved at det kan ha hatt en innvirkning på resulterende revisjonsprosess og produksjon.

Underveis i utarbeidelsen av skriptene, ble det erfart at kunnskapsnivået om Rvt og Fd, ikke var grundig nok. Hadde jeg hatt mer innsikt i hvordan API-et til relevante elementer og metoder er definert i hvert av programmene, kunne det økt sjansen til å få til å utvikle Python-komponenter til produksjon av armering og målsetting, i hver av modellene. Men verken en grundig nok forståelse av API-ene og nødvendigheten av å bruke Python-komponenter for å få gjennomført produksjonen av produkter i GH, var kjent før oppgaven startet.

Begrensningen på casen, om at resultatene ikke skulle vurderes i dette prosjektet, kan også bli ansett som en svakhet. Selv om produktene utarbeidet i GH ble modellert og definert i Fd, gav modellen ingen indikator på om dette er en hensiktsmessig måte å prosjektere et brudekke med krumninger. Her kunne oppgaven vært forbedret ved å fokusere mer på lastene og idealisering av brudekket. Det ville gitt en indikator på hvilken betydning en parametriske Fd-modell, kan gi i prosjekteringsprosessen i henhold til revisjonene. Her ble tidsrammen gitt oppgaven, grunnen til at dette ikke ble gjennomført.

Den parametriske prosjekteringen kunne funnet ut om et krummet brudekke ville blitt løst med oppdeling av rette linjer, om forskningen hadde vært kumulativ. Hadde konklusjonen fra for eksempel Master 1 blitt tatt i betraktning når egen parametriske prosjektering ble gjennomført, kunne resultatene også inneholdt utfallet av et krummet brudekke, oppdelt i korte rette linjer.



## 9 Konklusjon

Målet med masteroppgaven har vært å finne ut hvordan parametriske brudekkeprosjektering kan benyttes, til å generere en effektivisert revisjonsprosess, av geometrisk- og statisk modell. For å besvare dette, ble det gjennomført et litteraturstudie, innhentet erfaringer av mekatronikkingniøren og gjennomførte en parametriske prosjektering. Prosjekteringen satt søkelys på å benytte det parametriske verktøyet GH, til å prosjektere Rvt- og Fd-modeller. Produktene var tiltenkt å skulle brukes i hele prosjekteringsprosessen; fra tidligfasen til ferdigstilt konstruksjon. I henhold til dette, konkluderes det med at parametriske prosjekteringen av brudekker, kan få en effektivisert revisjonsprosess. Dette til bruk i tidligfasen, og at det ikke er egnet til å tas i bruk i hele prosessen.

Parametriske prosjektering i GH forbedrer ikke utvekslingen mellom CAD og FEM-modellen, den heller unngår den. Det er mulig å unngå filoverføring, ved at arbeidsflyten innebærer at store deler av prosjekteringen av både geometrisk- og statisk modell, samles i GH. Det er dermed ikke nødvendig å produsere en analytisk modell i Rvt, til bruk i Fd. Parametriske design åpner opp muligheten med å arbeide i GH, med produksjon av både CAD- og FEM-modell. Her utarbeides en felles referansegeometri, som benyttes til begge.

GH kan benyttes til å produsere parametriske prosjekteringselementer til både Rvt og Fd, som geometrisk og statisk modell. I den geometriske Rvt-modellen kan det produseres et brudekke som et bjelke- og direkte form-element, nivåer og akser. I tillegg til kurvene for tverr- og lengdearmring i brudekket, i form av lukkede rør. Armeringen fungerer ikke som en reell 3D strukturell armering. I den statiske Fd-modellen er det mulig å produsere et brudekke og tverrbærere som bjelke-element, tverrprofil, opplegg, laster, lastkombinasjoner, lasttilfeller og RC analyse. RC analysen er produsert ved at type analyser er definerbare i GH, mens analysen i seg selv kjøres direkte i Fd. Her er et brudekke med horisontal-og/eller vertikalkrumning, et fullverdig element i Rvt, men ikke i Fd. Det kommer av at elementet består av en buet veilinje, som ikke kan armeres til tross for at elementet analyseres i henhold til lasttilfellene og lastkombinasjonene. Dette betyr at armering, målsetting og snitt må utarbeides i tilhørende programmer, da disse ikke var mulig å produsere med GH.

Parametriske prosjektering muliggjør en effektiv revisjonsprosess av brudekkene i begge programmene. Her kan elementet revideres i henhold til horisontal- og vertikalkurveradius, tverrfallprosent, tverrsnittsform og høydeforskjell mellom start- og endepunkt. I tillegg kan prosjektinstilling i Rvt, antall og nummereringssystem til aksene, valg av én eller to nivåer, samt høyden mellom nivåene, effektivt revideres. Disse er parametrisert slik at nye verdier i parameterne oppdaterer elementene i en gjensidig og automatisert avhengighet. Kurvene til 3D armering i Rvt, kan også revideres parametriske, i henhold til utforming av brudekket,  $\phi$ -størrelse, senteravstand, antall lag armering i bjelkeområdet til dekket, og nominell overdekning. Også egenskaper til betongmaterialet, karakteristiske lastverdier, lastfaktorer til lastkombinasjonene, samt type lastkombinasjon inngår i en effektiv revisjonsprosess av Fd-modellen.

Parametriske prosjektering med ferdigkomponerte komponenter i GH, har mulighet til å generere en effektivisert revisjonsprosess av geometrisk- og statisk modell i tidligfasen. Dette kommer av at prosjekteringselementene både produseres og revideres i GH, med en gjensidig avhengighet og med nøyaktige koblinger. Det omhandler ikke revisjon av armerings og målsettingsproduktet, verken i Rvt

eller Fd. Både armering og målsetting til Rvt- og Fd-brudekket, må utføres direkte i hvert av programvaren, etter at revisjonsprosessen er utarbeidet i GH. Derfor er parametrisk prosjektering mest effektiv i tidligfasen, da denne fasen innebærer mindre detaljer og beregningsoptimaliseringer.

## 10 Anbefalinger

Til videre arbeid kan det være av interesse å undersøke følgende temaer:

- Undersøke utfallet av å dele opp brudekket, som et bjelke-element i FEM-Design, i mindre rette linjer når det inneholder horisontal- og/eller vertikalkurve. Det er av interesse å finne ut om brudekket da analyseres i henhold til designet, og dermed kan armeres.
- Definere en referansegeometri hvor inngangsparametere for utarbeidelse av veilinje er definert av variable start- og endepunkt, eventuelt koordinater. Eller at inngangsvinkelen til start- og endepunkt kan parametriseres og defineres med variable parametere.
- Implementere muligheten for å spesifikt definere veilinje i henhold til en variabel parameter for z-verdiene til start- og endepunkt, fremfor høydeforskjell som er brukt i denne oppgaven. Eventuelt muliggjøre parametrisk forflytning av brudekket i z-retning.
- Videreutvikle Grasshopper-skriptene, ved å undersøke hvordan komponenter i GH kan settes sammen, for å unngå at produkter produsert i Grasshopper må slettes når variable parametere endres. Og om det er mulig å unngå at elementene må slettes, slik at produktene armering og målsetting ikke forsvinner når revisjonsprosessen gjennomføres.
- Ta i bruk Python-komponenter og utarbeide innholdet slik at Revit og FEM-Design API-ene kan benyttes til å modellere og prosjektere armering. Finne ut hvordan Python-komponenten kan utarbeides for å produsere strukturell armering i Revit-brudekket. Enten i form av armering produsert i henhold til kurver i Grasshopper, eller i form av å benytte metoden «Free form». Det er også av interesse å finne eller utarbeide en komponent som kan armere bjelke-elementer til FEM-Design modellen, i Grasshopper.
- Undersøke om en oppdeling av et buet brudekke, løser problemet med at det ikke kan armeres i FEM-Design. En oppdeling av veilinjen, med korte rette linjer.
- Utnytte «Tree» komponentene, til å automatisere avhengighetene mellom lastprosjekteringen i Grasshopper-skriptet ytterligere.

## Referanser

- [1] MODELAB. 2015. *The grasshopper primer (en)* [Online]. Hentet fra: <http://grasshopperprimer.com/en/index.html>. Lastet ned: 08.02.2021.
- [2] R. M. associates. 2021. *Getting started with rhino.Inside.Revit* [Online]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/getting-started>. Lastet ned: 20.01.2021.
- [3] A. Aamodt, "Parametrisk design, implementering av visuell parametrisk design i prosjekteringsprosessen," Forprosjekt til masteroppgaven, Teknologi og realfag, UiA, 2020.
- [4] S. Vaezi, "Studier i parametrisk modellering, 3d-cad modeling applications " Eksamensarbeid Civil and environmental engineering Chalmers University of Technology Goteborg, Sverige, 45, 2007.
- [5] wikipedia. 2020. *Boolean expression* [Online]. Hentet fra: [https://en.wikipedia.org/wiki/Boolean\\_expression](https://en.wikipedia.org/wiki/Boolean_expression). Lastet ned: 11.05.2021.
- [6] E. K. Jøns Sjøgren, Lars Chr Christensen og Karoline H Olsen-Skåre "Digitalt veikart - for en heldigitalisert, konkurransedyktig og bærekraftig bae-næring," 19.02.2017 2017, <https://www.bnl.no/dokumenter/digitalt-veikart/>.
- [7] M. G. T. o. L. B. Anne Espelien, "En verdiskapende bygg-, anlegg- og eiendomsnæring (bae) " Handelshøyskolen BI1/2015 2015, [0803-2610], [https://www.bi.no/globalassets/forskning/senter-for-byggenaringen/publikasjon/2015-01-espelien\\_theie\\_bygballe.pdf](https://www.bi.no/globalassets/forskning/senter-for-byggenaringen/publikasjon/2015-01-espelien_theie_bygballe.pdf).
- [8] BNL. *Digitalt veikart for bygg-, anleggs- og eiendomsnæringen for økt bærekraft og verdiskaping*. Hentet fra: <https://www.bnl.no/siteassets/dokumenter/rapporter/digitalt-veikart-bae.pdf>. Lastet ned: 26.04.2021.
- [9] FN-Sambandet. 2021. *Fns bærekraftsmål* [Online]. Hentet fra: <https://www.fn.no/om-fn/fns-baerekraftsmaal>. Lastet ned: 14.01.2021.
- [10] United Nations. *Goal 9: Build resilient infrastructure, promote sustainable industrialization and foster innovation* [Online]. Hentet fra: <https://www.un.org/sustainabledevelopment/infrastructure-industrialization>. Lastet ned: 26.04.2021.
- [11] K. M. R. K. Lalit Narayan, M.M.M. Sarcar *Computer aided design an manufacturing*, nr. Issue, s. Pages, E-Pub, [Online]. Hentet fra:
- [12] J. Krebs, *Basics fundamentals of presentation cad* ", nr. Issue, s. Pages, E-Pub Date. doi: 10.1515/9783035612745-009 [Online]. Hentet fra:
- [13] Autodesk. 2020. *What is cad software* [Online]. Hentet fra: <https://www.autodesk.com/solutions/cad-software>.
- [14] Alfredo Andia, *Post-parametric automation in design and construction*, nr. Issue, s. Pages, E-Pub, [Online]. Hentet fra:
- [15] M. C. Jorge D. Camba, Pedro Company "Computer-aided design," *Parametric CAD modeling: an analysis of strategies for design reusability* vol. 74, nr. 18-31, s. 31, 16.01.2016. <https://reader.elsevier.com/reader/sd/pii/S0010448516000051?token=C14F882113E6F0E169BE7DA90533BF52448055A2254542CD951D02DEDB1E1B658AD0A77CC9F8CD64536D1B45D30230C5>. Hentet fra: <https://reader.elsevier.com/reader/sd/pii/S0010448516000051?token=C14F882113E6F0E169BE7DA90533BF52448055A2254542CD951D02DEDB1E1B658AD0A77CC9F8CD64536D1B45D30230C5>

- [16] Symetri. *Interer konstruksjonsprosessen i bedriftens arbeidsforløp* [Online]. Hentet fra: <https://www.symetri.no/produkt-design-plm/plm/dele-konstruksjonsdata/>. Lastet ned: 27.04.2021.
- [17] P. Barnes, *Bim in prinsiple and in practice*, 3. utg. London ICE Publishing, 2019.
- [18] O. S. Kolstad, *Revit på norsk*. 07 Oslo: Symra 2014, 2015.
- [19] Fokus software. 2021. *Dokumentasjon for autodesk revit* [Online]. Hentet fra: <https://www.fokus.no/produkter/autodesk/revit/revit-dokumentasjon/>. Lastet ned: 27.04.2021.
- [20] ParallaxTeam. *The dynamo primer* [Manual online ]. Hentet fra: <https://primer.dynamobim.org/index.html>.
- [21] Autodesk. 2020. *Funksjoner, programvare for modellering av bygningsinformasjon* [Online ]. Hentet fra: <https://www.autodesk.no/products/revit/features?plc=RVT&term=1-YEAR&support=ADVANCED&quantity=1>.
- [22] R. M. Associates. 2021. *Overview* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/guides/revit-revit>. Lastet ned: 17.02.2021.
- [23] P. Aubin. 2020. *Revit families: A step-by-step introduction* [Online Artikkel ]. Hentet fra: <https://www.autodesk.com/autodesk-university/article/Revit-Families-Step-Step-Introduction-2018>.
- [24] S. Sunnersjö, *Fem i praktiken*, 2 utg. MediaPrint, Uddevalla AB: 1992 författaren och industrilitteratur AB 1999.
- [25] Sintef. *Finite element method (endelige elementmetoder)* [Online]. Hentet fra: <https://www.sintef.no/finite-element-method-endelige-elementmetoder/>.
- [26] D. N. A. f. S. o. Litteratur. *Det norske akademis ordbok*, [Online]. Hentet fra: <https://naob.no/ordbok/diskretisering>.
- [27] Autodesk. 2020. *Hva er dwg?* [Online]. Hentet fra: <https://www.autodesk.no/products/dwg>.
- [28] Strusoft. 2010. *User manual fem-design* [Manual online]. Hentet fra: <http://download.strusoft.com/FEM-Design/inst110x/manual.pdf>.
- [29] FEM-Design Wiki. 2020. *Fem-design api* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/femdesignwiki/view/FEM-Design%20API/Introduction/>. Lastet ned: 09.03.2021.
- [30] F. Lagerström. 2020. *Structure definition* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/Manuals/User%20Manual/Structure%20definition/>. Lastet ned: 13.04.2021.
- [31] I. B. Bjergø. 2021. *Struxml revit add-in* [Online]. Hentet fra: StruXML Revit Add-In - FEM-Design Wiki (strusoft.com) Lastet ned: 27.04.2021.
- [32] I. B. Bjergø. 2021. *Getting started* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/BIM%20Integration/StruXML%20Revit%20%20Add-In/Getting%20started/#Download>. Lastet ned: 23.03.2021.
- [33] I. B. Bjergø. 2020. *Import struxml to revit* [Online]. Hentet fra: Import struxml to Revit - FEM-Design Wiki (strusoft.com). Lastet ned: 27.04.2021.
- [34] I. B. Bjergø. 2020. *Opening struxml in fem-design* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/BIM%20Integration/StruXML%20Revit%20%20Add->

- [In/Export%20StruXML/Export%20struxml%20manual/Opening%20struxml%20in%20FEM-Design/](#). Lastet ned: 27.04.2021.
- [35] I. B. Bjergø. 2020. *Reinforcement* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/BIM%20Integratoin/StruXML%20Revit%20%20Add-In/Import%20StruXml/Import%20scope/Reinforcement/>. Lastet ned: 27.04.2021.
- [36] Arup, "Parametric design for better buildings," 2021.
- [37] Graphisoft. 2019. *Parametrisk design/visuell programmering inntar byggebransjen* [Online]. Hentet fra: <https://graphisoft.no/archicad/parametrisk-design-visuell-programmering-inntar-byggebransjen/>. Lastet ned: 28.01.
- [38] Graphisoft. 2020. *Parametrisk designseminar oslo 11. I februar 2020* [Online ]. Hentet fra: <https://graphisoft.no/kurs-kat/parametric-design-seminar-oslo/>.
- [39] M. Ganz, "Parametrisk design," 23.10, 2019.
- [40] Structures Parametrics, "Grasshopper vs dynamo parametric pros and cons for engineers. Revit vs rhino. Which is better? ," utg.: Youtube, 2020.
- [41] Structures Parametrics. 2019. *Beskrivelse* [Online]. Hentet fra: <https://www.youtube.com/channel/UCqgD1Dm9u5dCoi0f5dPcXIA/about>. Lastet ned: 27.04.2021.
- [42] S. Davidson. 2021. *Grasshopper* [Online]. Hentet fra: <https://www.grasshopper3d.com/>. Lastet ned: 08.02.2021.
- [43] S. Davidson. 2021. *Download* [Online]. Hentet fra: <https://www.grasshopper3d.com/page/download-1>. Lastet ned: 27.04.2021.
- [44] Forfatter, "Eget arbeid," 2021
- [45] Rhino Grasshopper, "Rhino grasshopper tutorial (beginner)," utg.: Youtube, 2018.
- [46] R. M. Associates. 2021. *Getting started with rhino.Inside.Revit* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/getting-started#what-is-rhinoinsiderevit>. Lastet ned: 27.04.2021.
- [47] Simply Rhino Limited. *Rhino.Inside.Revit essentials* [Online ]. Hentet fra: <https://simplyrhino.co.uk/files/Simply-Rhino-RhinoInsideRevit-Essentials-Course-Outline.pdf>. Lastet ned: 13.04.2021.
- [48] K. Garcia. *Rhino.Inside for autodesk revit* [Online]. Hentet fra: <https://www.food4rhino.com/app/rhinoinside-autodesk-revit> Lastet ned: 27.04.2021.
- [49] R. McNeel. 2021. *Release notes* [Online]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/reference/release-notes>. Lastet ned: 24.03.2021.
- [50] R. McNeel. 2021. *Download rhino.Inside revit for windows wip* [Online]. Hentet fra: <https://www.rhino3d.com/download/rhino.inside-revit/7/wip>. Lastet ned: 24.03.2021.
- [51] R. M. Associates. 2021. *Grasshopper in revit* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/guides/rir-grasshopper>. Lastet ned: 17.02.2021.
- [52] I. Bjorhag. 2020. *Grashopper toolbox* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/femdesignwiki/view/FEM-Design%20API/Grasshopper%20API/>. Lastet ned: 19.01.2021.
- [53] food4Rhino. 2021. *Fem-design api toolbox (by strusoft)* [Online]. Hentet fra: [https://www.food4rhino.com/app/fem-design-api-toolbox#downloads\\_list](https://www.food4rhino.com/app/fem-design-api-toolbox#downloads_list). Lastet ned: 19.01.2021.

- [54] github. 2021. *Fem-design api* [Online]. Hentet fra: <https://github.com/strusoft/femdesign-api/tree/grasshopper>. Lastet ned: 27.04.2021.
- [55] R. M. Associates. 2019-2021. *New in rhino 7* [Online]. Hentet fra: <https://www.rhino3d.com/features/>. Lastet ned: 10.02.2021.
- [56] Autodesk Help. 2021. *Rebar shape constraints and cover* [Online]. Hentet fra: <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/Revit-Model/files/GUID-C9757E0B-5ADB-4BEB-B6D4-28AC730FBE5E-htm.html>. Lastet ned: 28.04.2021.
- [57] Autodesk Help. 2021. *About free form reinforcement* [Online]. Hentet fra: <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/Revit-Model/files/GUID-E1FD47EA-F9CB-4A77-A8BC-17A12672FF40-htm.html>. Lastet ned: 28.04.2021.
- [58] Autodesk Help. 2021. *Place rebar* [Online]. Hentet fra: <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/Revit-Model/files/GUID-D61AF551-AD15-4FEC-966F-C048E0885212-htm.html>. Lastet ned: 28.04.2021.
- [59] Autodesk Help. 2021. *Sketch free form rebar* [Online]. Hentet fra: <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/Revit-Model/files/GUID-B740C331-72B1-4B50-9898-F588B2513CCB-htm.html>. Lastet ned: 28.04.2021.
- [60] D. BD, "Curved bridge deck detailing in revit," utg.: Youtube, 2019.
- [61] F. Lagerström. 2020. *Loads* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/Manuals/User%20Manual/Loads/>. Lastet ned: 13.04.2021.
- [62] I. Björhag. 2020. *Analysis* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/Manuals/User%20Manual/Analysis/>. Lastet ned: 13.04.2021.
- [63] F. Lagerström. 2020. *Rc design* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/femdesignwiki/view/Manuals/User%20Manual/RC%20Design/>. Lastet ned: 29.04.2021.
- [64] F. Lagerström. 2020. *Design* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/Manuals/User%20Manual/Design/>. Lastet ned: 13.04.2021.
- [65] S. Davidson. 2019. *Your first python script in grasshopper* [Online]. Hentet fra: <https://developer.rhino3d.com/guides/rhinopython/your-first-python-script-in-grasshopper/>. Lastet ned: 01.03.2021.
- [66] R. M. Associates. 2021. *Python component in revit* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/guides/rir-ghpython>. Lastet ned: 19.03.2021.
- [67] R. M. Associates. 2021. *Revit: Elements instances* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/guides/revit-elements>. Lastet ned: 17.02.2021.
- [68] R. M. Associates. 2021. *Revit: Parameters* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/guides/revit-params>. Lastet ned: 17.02.2021.

- [69] R. M. Associates. 2021. *Revit: Types and families* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/guides/revit-types>. Lastet ned: 17.02.2021.
- [70] S. vegvesen. 2014. *Håndbok v440 bruregistrering* Hentet fra: <https://www.vegvesen.no/attachment/111840/binary/964064>.
- [71] Statens vegvesen. 2021. *Rv.9 besteland-helle - fakta* [Online]. Hentet fra: <https://www.vegvesen.no/Riksveg/rv9bestelandhelle/fakta>. Lastet ned: 27.01.2021.
- [72] H. Johansen, "Beregningsveiledning for etteroppspente betongbruer " Statens Vegvesen10.01.2017 2017, [v668], <https://vegvesen.brage.unit.no/vegvesen-xmlui/bitstream/handle/11250/2569768/SVV%20rapport%20%20668%20Beregningsveiledning%20for%20etteroppspente%20betongbruer.pdf?sequence=1&isAllowed=y>.
- [73] Wikipedia. 2013. *Cad* [Online]. Hentet fra: <https://no.wikipedia.org/wiki/CAD> Lastet ned: 07.03.2021.
- [74] Wikipedia. 2013. *Parametric design* [Online]. Hentet fra: [https://en.wikipedia.org/wiki/Parametric\\_design](https://en.wikipedia.org/wiki/Parametric_design). Lastet ned: 07.03.2021.
- [75] Granruds. 2016. *Joomla nyheter* [Online]. Hentet fra: <https://granruds.no/nyheter/14-joomla-video-med-tips-og-triks-pa-youtube>. Lastet ned: 07.05.21.
- [76] UiA. *Litteratursøk(digitalt kurs)* [Online]. Hentet fra: <https://uia.pameldingssystem.no/litteratursok-digitalt-kurs> Lastet ned: 03.02.2021.
- [77] I.-J. Thidemann, *Bachelor-oppgaven for sykepleier-studenter*, 2. utg. 0105 Oslo: Universitetsforlaget, 2019.
- [78] O. Dalland, *Metode og oppgaveskriving*, 6. utg. 0130 Oslo: Gyldendal Norge 2019.
- [79] G. A. Pedersen, "Litteraturstudie som metode," utg.: Youtube, 2018.
- [80] Rhino Grasshopper. *Info* [Online]. Hentet fra: <https://www.youtube.com/channel/UCjLDKM9EzNdASaNdjBhTqug/about>. Lastet ned: 10.04.2021.
- [81] J. E. Peter Wentworth, Allen B. Downey and Chris Meyers. 2012. *Some tips, tricks, and common errors* [Online]. Hentet fra: [https://openbookproject.net/thinkcs/python/english3e/app\\_e.html](https://openbookproject.net/thinkcs/python/english3e/app_e.html). Lastet ned: 15.04.2021.
- [82] V. Pavlenko. 2012-2018. *Lesson 1, input, print and numbers* [Online]. Hentet fra: [https://snakify.org/en/lessons/print\\_input\\_numbers/](https://snakify.org/en/lessons/print_input_numbers/). Lastet ned: 16.04.2021.
- [83] apidocs. *Namespaces, revit 2021.1 api* [Online]. Hentet fra: <https://apidocs.co/apps/revit/2021.1/d586b341-f687-9d90-e96d-255806b7d4fc.htm>. Lastet ned: 20.04.2021.
- [84] R. M. Associates. 2021. *Rhino.Inside.Revit beta* [Online]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/>. Lastet ned: 17.02.2021.
- [85] Aussie BIM Gurr, "Rhino/grasshopper tutorials," utg.: Youtube, 2019.
- [86] A. B. Guru. 2019. *Info aussie bim guru* [Online]. Hentet fra: <https://www.youtube.com/c/AussieBIMGuru/about>. Lastet ned: 28.04.2021.
- [87] XWikiGuest. 2020. *Hello world* [Online]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/wiki.fem-design.strusoft.com/view/FEM-Design%20API/Grasshopper%20API/Getting%20started/>. Lastet ned: 28.04.2021.



- [88] I. Björhag. 2020. *Grid on freeform surface* [Online film]. Hentet fra: <https://wiki.fem-design.strusoft.com/xwiki/wiki/femdesignwiki/view/FEM-Design%20API/Grasshopper%20API/Surface/>. Lastet ned: 27.02.2021.
- [89] food4Rhino. 2017. *Bifocals* [Online]. Hentet fra: <https://www.food4rhino.com/app/bifocals>. Lastet ned: 03.02.2021.
- [90] *Veg- og gateutforming*, Håndbok N100, 2019.
- [91] *Bruprosjektering* Håndbok N400 2015.
- [92] Apidocs. *Apidocs* [Online ]. Hentet fra: <https://apidocs.co/apps/revit/2021.1/d4648875-d41a-783b-d5f4-638df39ee413.htm#>. Lastet ned: 19.03.2021.
- [93] R. M. Associates. 2021. *Project settings* [Online veileder]. Hentet fra: <https://www.rhino3d.com/inside/revit/beta/guides/revit-psettings>. Lastet ned: 05.03.2021.

## Vedlegg

Vedlegg A – Sagåne bru\_Beregningsrapport

Vedlegg B – Sagåne bru\_RIB-teninger

Vedlegg C – Birkelid bru\_Beregningsrapport

Vedlegg D – Inndata til lastdimensjoneringen av Sagåne bru

Vedlegg E – Utarbeidelsen av ny familie i Rvt

Vedlegg F – Filer brukt til den parametriske prosjekteringen

Vedlegg G – Resulterende GH-, Rvt- og Fd-filer

Vedlegg H – Grasshopper-skript produktene

Vedlegg I – Parameterne benyttet til resultatene i kapittel 7.3.3

Vedlegg J – Revit produktene

Vedlegg K – FEM-Design produktene

Vedlegg L – A3 poster

# Vedlegg D

Inndata til lastdimensjoneringen, Sagåne bru

All informasjon presenteres videre er hentet fra beregningsrapporten til Sagåne bru [Vedlegg A] og RIB-tegningene [Vedlegg B]. Det som gjelder idealiseringen av vindlastplasseringen som statisk system, er å finne i beregningsrapporten til Birkelid bru i Vedlegg C.

### 1. Egenlast

Egenlastene innebærer vekten av betongdekket, samt lastene som er presentert i tabellen under.

Last	Verdi	Enhet
Egenvekt belegning	3,5	kN/m <sup>2</sup>
Egenvekt kantdrager	7,5	kN/m
Egenvekt rekkverk	0,75	kN/m

### 3. Trafikklaster vertikale

Bruen er dimensjonert for 3 kjørefelt, hvor kjørefeltsbredden er på 3 meter og restbredden på 1,5 meter (Tabell 4.1 – NS-EN 1991-2). Bruen er dimensjonert for lastmodellene LM1, LM2 og LM3. Her er det blitt foretatt en begrensning i masteroppgaven, å kun inkludere LM1. det betyr at LM2 og LM3 ikke utdypes noe ytterligere.

Lastmodell 1 består av en jevnfordelt last ( $q_{ik}$ ) og boggilaster ( $Q_{ik}$ ). Her er  $q_{ik}$  definert feltvis med tre mulige plasseringer for lastfelt 1, som er på 5,4 kN/m<sup>2</sup>, og er påsatt på brudekket som linjelast. For  $Q_{ik}$  ble det kun tatt med et sett med boggilast. Dette kan bestå av 12 eller 6 punktlaster. Lastene fra LM1 superponeres.

### 4. Horisontale trafikklaster

Disse lastene består av bremse- og akselerasjonslaster, i tillegg til tverrlaster og sentrifugalkrefter. De påføres i nivå med bruens overflate. Bremselasten ( $Q_{ik}$ ) regnes ut som en andel av maksimal vertikallast, fra LM1. Lasten plasseres på brudekket som en linjelast med eksentrisitet til bjelkeelementet. Her er akselerasjonslasten ekvivalent med bremselasten, bare i motsatt retning. Tverrlasten fra bremsing og akselerasjonen ( $Q_{trk}$ ) er definert som 25% av totalen til bremse – eller akselerasjonslasten. Den påføres videre som jevnt fordelt flatelast ( $q_{trk}$ ) og opptrer samtidig som  $q_{ik}$ .

### 5. Vindlast

Rambøll har verifisert at konstruksjonen ligger i vindklasse I, ved å identifisere at laveste svingeperiode for bruene er mindre enn 2 sekunder. Vindlaster tas med i situasjoner med og uten trafikklast, og er beregnet i henhold til NS-EN 1991-1-4. Vindlasten her, beregnes med et vindfelt hvor kastvindhastigheten ved kjørebans høyeste punkt er lik 35 m/s, eller med et vindfelt med returperiode på 50 år, dersom det gir en lavere verdi.

Her ble den største krafteffekten, vindlast med trafikklast. Denne lastkombinasjonen ble derfor benyttet som vindbelastning i alle tilfeller. Denne ble påført globalt på modellen, med en eksentrisitet. En tabell over lasttilfellene, med og uten trafikklast, er presentert i **Feil! Fant ikke referanse-kilden..**

Vindretning	Tverretning [kN/m]	Lengderetning [kPa]	Vertikalt [kPa]	Kommentar
Vind uten trafikk	2,01	0,04	0,51	Neglisjeres
Vind med trafikk	4,20	0,09	0,51	Benyttes
Vind med trafikk (N400)	5,71	0,12	0,69	Øvre grense

## 6. Temperaturlaster

Temperaturlastene er beregnet etter NS-EN 1991-1-5 for type 3 betongtverrsnitt. Her er temperatur på byggetidspunktet antatt å være  $T_0 = +7^\circ\text{C}$ . Denne verdien er hentet fra isotermkartet, for maksimum og minimumstemperatur i skyggen, med en returperiode på 50 år.

Den termiske påvirkningen på bruen, regnes som lineært varierende temperaturdifferanse over brutverrsnittet, i henhold til NS-EN 1991-1-5 pkt. NA.6.1.2. Her er den varierende temperaturdifferanseandelen for betongbjelken gitt i tabell NA.6.1, som er beregningsgrunnlaget for temperaturlastene som er presentert i **Feil! Fant ikke referanseskilden..**

Temperatur	Verdi	Enhet
$\Delta T_{N,exp}$	26	$^\circ\text{C}$
$\Delta T_{N,con}$	-29	$^\circ\text{C}$
$\Delta T_{M,heat}$	15	$^\circ\text{C}$
$\Delta T_{M,cool}$	8	$^\circ\text{C}$
$\Delta T_N$	43	$^\circ\text{C}$

Jevnt fordelt temperaturfordeling ( $\Delta T_{N,exp}$  og  $\Delta T_{N,con}$ ), samt lineær varierende temperaturdifferanse ( $\Delta T_{M,heat}$  og  $\Delta T_{M,cool}$ ), kombineres i henhold til formel 6.1.5 og NA.1.5 i NS-EN 1991-1-5 med faktorene

$$\omega_N = 0,35$$

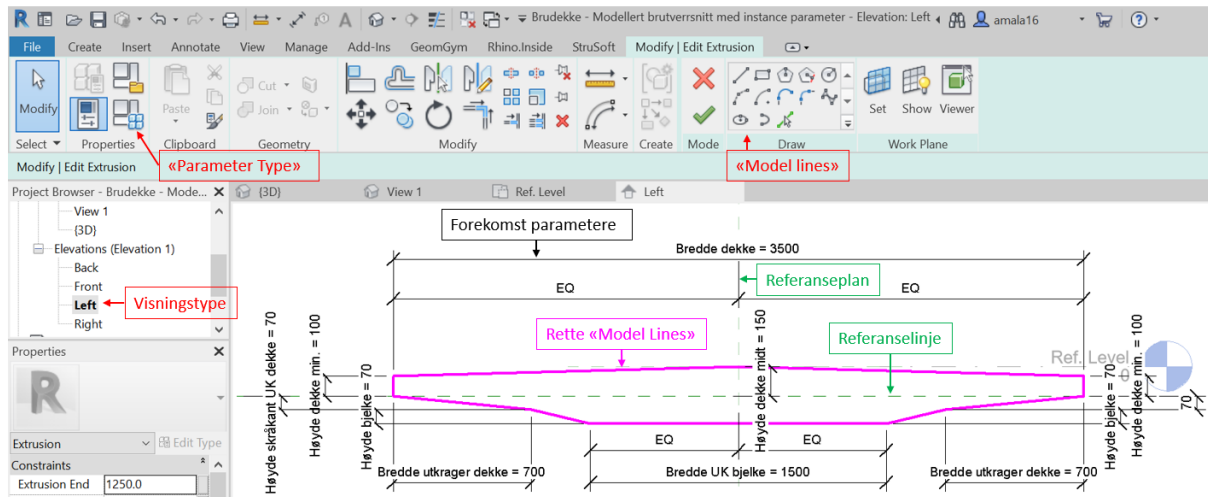
$$\omega_M = 0,75$$

# Vedlegg E

Utarbeidelsen av ny familie i Revit

Tverrsnittprofilen produsert i Rvt, ble utarbeidet som en egendefinert type, i kategorien «Structural Framing», i Revit. Dette ble gjennomført ved å produsere en ny familie, i malen «Metric Structural Framing – Beams and Braces».

I malen ble geometrien av brutverrsnittet tegnet opp, med «Model lines». Dette ble gjort i visning «Left». Tverrsnittet ble låst til å være symmetrisk i tverretningen, da dette var tilfellet i casen. Dette forenklet kompleksiteten av å produsere et parametrisk tverrsnitt. Formen på tverrsnittet er presentert i figuren under.



Deretter ble referanselinjen (grønn tekstbokser) tegnet opp. For referanseplanet, var dette allerede ferdigdefinert i malen, så den forble uendret. Det var videre nødvendig å utarbeide geometrien med parametriske variabler, for mulighet til endring av dimensjoner på tverrsnittet i ettertid.

Referanselinjen ble plassert i horisontalplanet, slik at målsettingen var mulig å definere med gjensidig avhengighet. Her ble målsettingstypen «Alignment» benyttet. For å opprettholde symmetrien i tverrsnittet, ble bredden av brudekket og UK bjelke, målsatt med en tilleggsdimensjon: EQ (betyr lik).

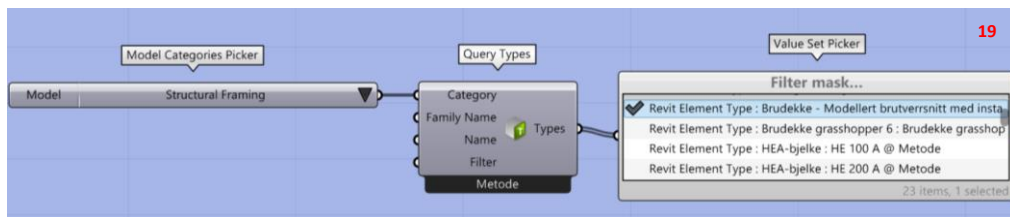
For å få tilgang på målne, som definerer tverrsnittet, i form av parametere i Rvt, ble det utarbeidet nye «Labels». Disse ble definert som forekomst-, og ikke type- parametere. Denne beslutningen var avgjørende for at det skulle være mulig å endre på dimensjonene av tverrsnittet, når familien ble importert i Rvt-dokumentet. Navnene som ble benyttet som «Labels» på målsettingene av tverrsnittet, er listet opp under, vist i figuren over.

- Brekke dekke
- Brekke UK bjelke
- Brekke utkrager dekke
- Høyde bjelke
- Høyde skrånkant UK dekke
- Høyde dekke min.
- Høyde dekke midt

At merknadene (Labels) ble gyldige forekomstparametere i familien, ble kontrollert i fanen «Parameter Type», som er den røde tekstboksen øverst til venstre i figuren. På denne måten ble

dimensjonene som definerte tverrsnittet tilgjengelige i «properties panelet», altså synlige parametere.

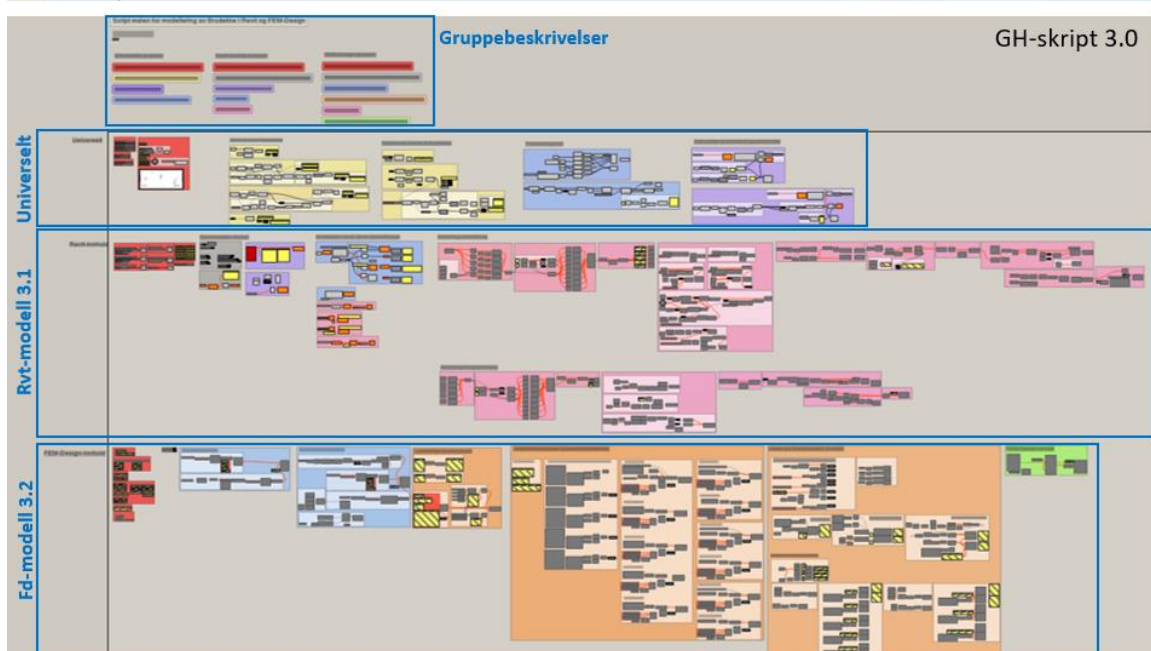
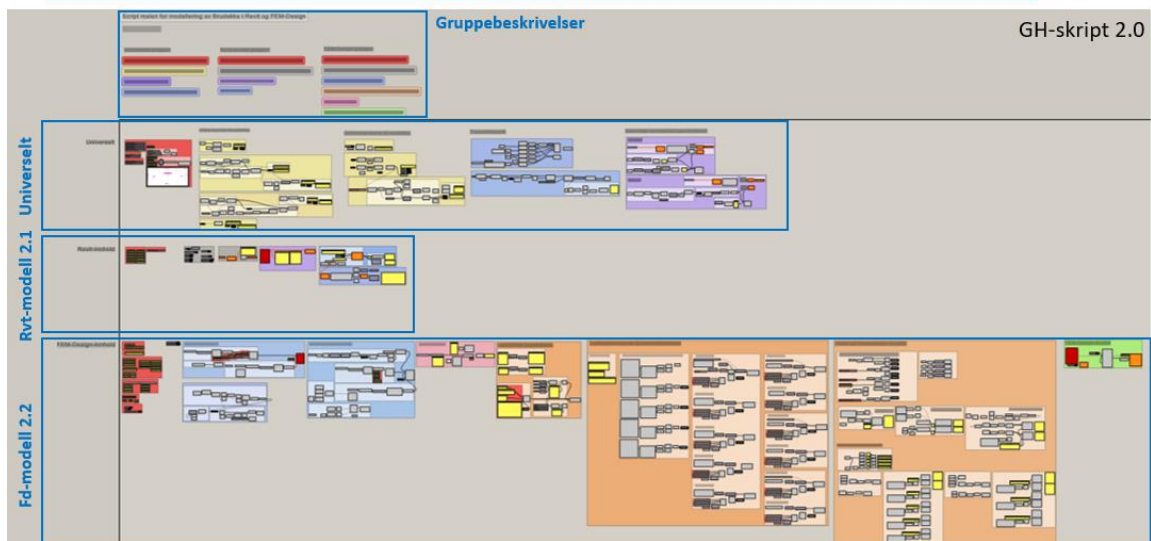
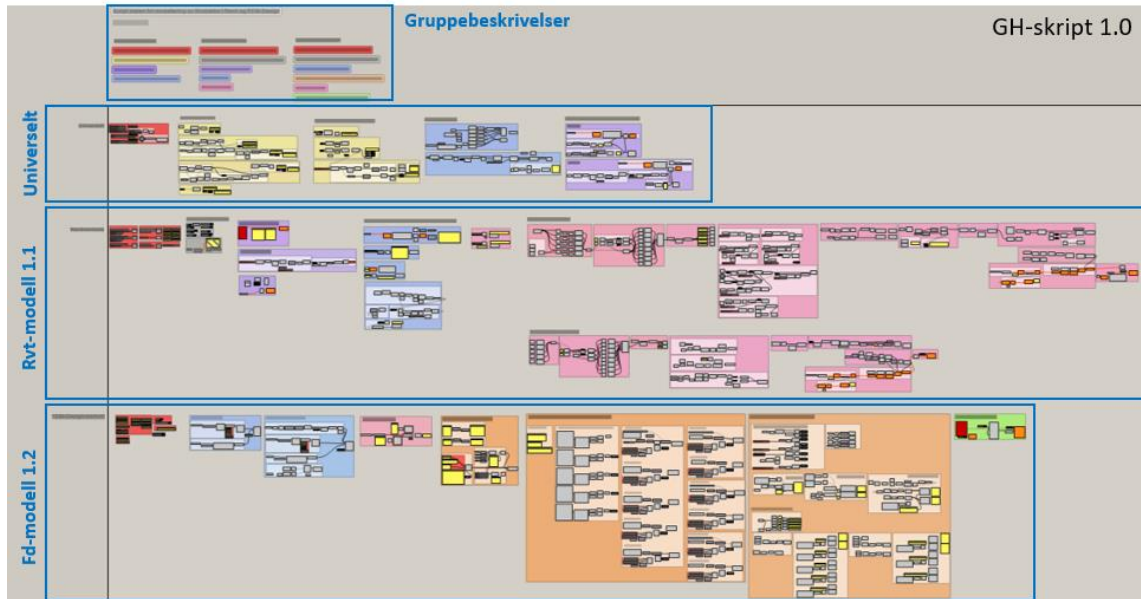
For at familien skulle kunne hentes ut i GH som en type i kategorien «Structural Framing», ble familien først lagret på intern server på egen datamaskin. Deretter ble den lastet inn i Rvt-dokumentet, som ble benyttet for utforming av parametrisk brudekke. For å få tilgang til denne familien i Grasshopper, definert som en type, ble komponentene som er presentert i figuren under benyttet. Datainnholdet i utgangsparameteren til komponenten «Value Set Picker» ble koblet inn i inngangen «Type» i «Add Beam».





# Vedlegg H

Grasshopper-skript produktene



# Vedlegg I

Parameterne benyttet til resultatene i kapittel 7.3.3

Parameterverdierne til resultatene presentert for prosjekteringselementene produsert med GH.

**De universelle parameterne:**

**Kurvener**

- Lengde brudekke i XY-planet: 22.100
- Radius horisontalurve pr. linje: 100.000
- Radius vertikalkurve pr. linje: 100
- Prosent tverrfall brudekke: 8
- Høydeforskjell start- og endepunkt(m): 1

**Dimensjoner tverrsnitt**

- A: Bredde UK bjelke: 4
- B: Bredde dekke: 3.000 (Antall kjørefelt: 3, Restbredde: 1.500)
- C: Bredde utrager dekke: 2.2
- D: Høyde bjelke: 0.5
- E: Høyde skrårand UK dekke: 0.2
- F: Høyde dekke min: 0.4

**Nivåer**

- Nummer på startpunkt: 1
- Antall nivå: 2

**Aksene**

- Aksene, Tillegg til bredde dekke: 1.000
- Antall akser vertikalt: 4

**Control Knob: G: Høyde dekke midt**

**Expression:**  $\tan(a) * (B/2) + F$

**Parameterne til Rvt-modellene:**

**Lengdearmering**

- OK dekke: Ø størrelse lengdearmering OK dekke: 25
- UK bjelke: Ø størrelse lengdearmering UK bjelke: 25
- Min. armering: Ø størrelse lengdearmering min arm: 16

**Nominell overdekning**

- c\_nom OK dekke: 35
- c\_nom UK dekke-bjelke: 35

**Tverrarmering**

- Ø størrelse lengde OK dekke: 25
- Ø størrelse lengde UK bjelke: 20
- Ø størrelse bøyer: 8

**Prosjektinnstillinger**

- Prosjekt navn: Masterprosjekt: brudekke
- Utførelsesdato: 05.03.2021
- Prosjektnummer: 20000
- Kundenummer: 123
- Gårds og bruksnummer: Gnr.1 Bnr.2
- Fag: RIB
- Tegnet av: AA
- Prosjektstatus: Arbeidsbegynnelser
- Kunde: UIA
- Kontrollert av: AB

**Kurve til Snitt**

- Antall snitt vinkelrett på brudekke: 4

**Parametere til Fd-modellene:**

**Egenvekt**

- Karakteristisk verd beleggningi (kN/m): 3.5
- Karakteristisk verdi Kantdrager (kN/m): 7.5
- Karakteristisk verdi rekkverk (kN/m): 0.75

**Vindlast**

- Karakteristisk verdi til vind med trafikk (kN/m): 4.20

**Temp. verdier**

- Delta TN\_exp: 26
- Delta TM\_heat: 15
- Delta TN\_con: -29
- Delta TM\_cool: 8

**Trafiklast vertikalt LM1**

Verdier hentet fra beregningsrapport Sågane bru

Karakteristiske TS laster	Karakteristiske UDL laster
Q_1k : Lane Nr. 1 (kN): 600	q_1k : Lane Nr. 1 (kN/m^2): 5.4
Q_2k : Lane Nr. 2 (kN): 400	q_2k : Lane Nr. 2 (kN/m^2): 2.5
Q_3k : Lane Nr. 3 (kN): 200	q_3k : Lane Nr. 3 (kN/m^2): 2.5
	q_rest_k : Resterende areal (kN/m...): 2.5

**Opplegg tverrbærere**

- Lengde fra endepunktene til tverrbærer: 2
- Bevegelse: x-retning: 0, y-retning: 0, z-retning: 100000

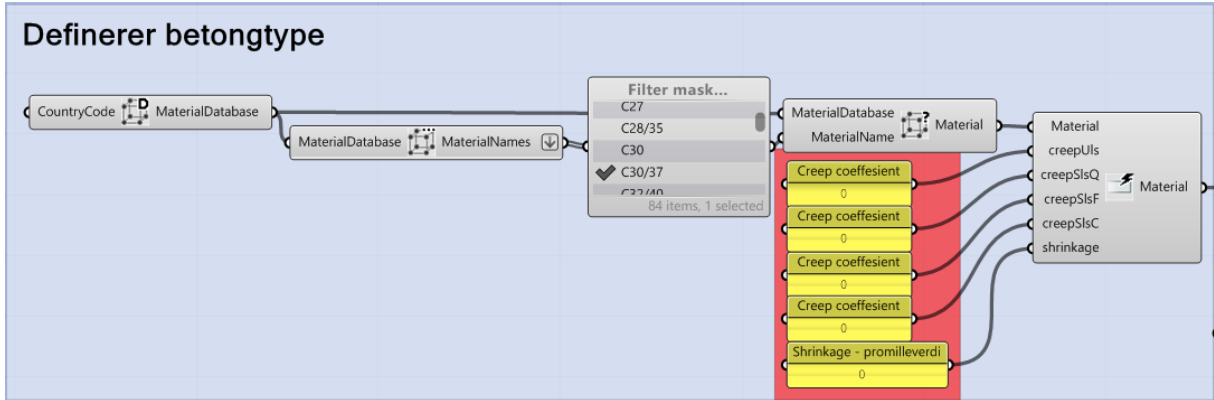
**FEM-Design fil, som produktene produseres i**

Til "file path" inganger

C:\Users\Jobb\OneDrive - Dagfin Skaar AS\02 - Masteroppgave\06 - Mine filiformater fra prosjektering\06 - Brudekke 1 - FEM-Design\FEM

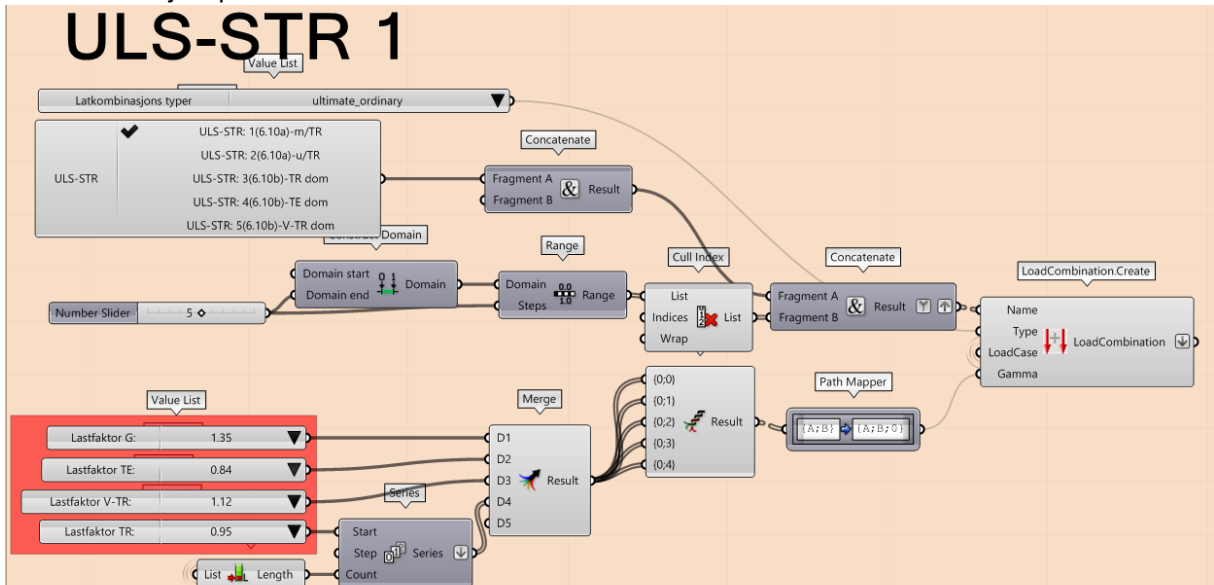
Materialtype til brudekke og tverrbærere:

### Definerer betongtype

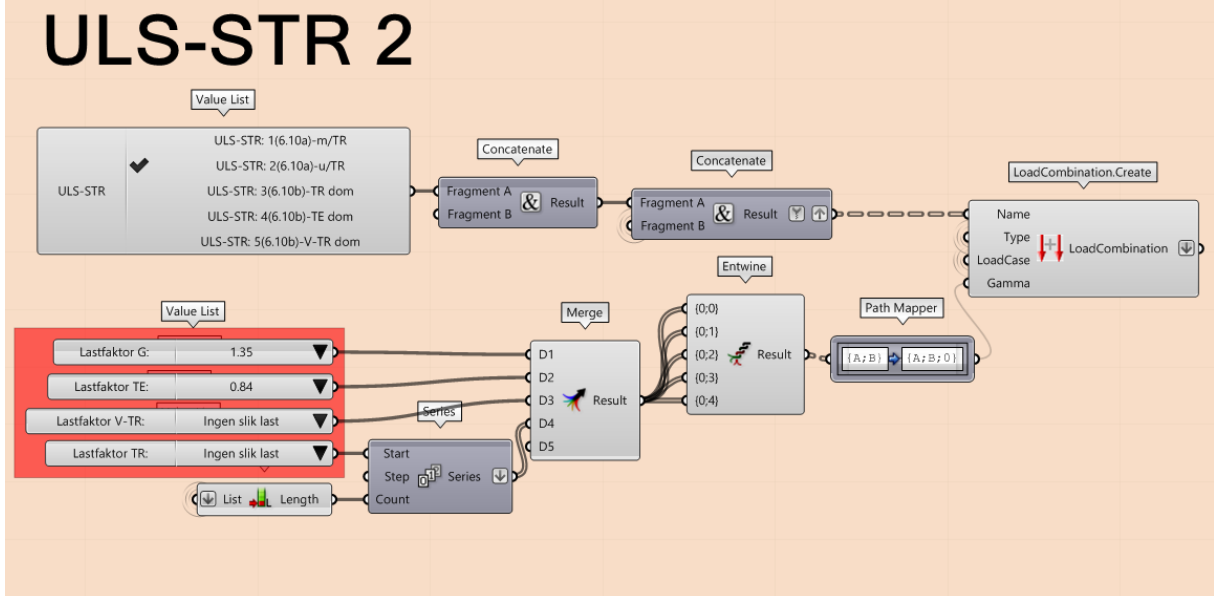


### Lastkombinasjonsparameterne:

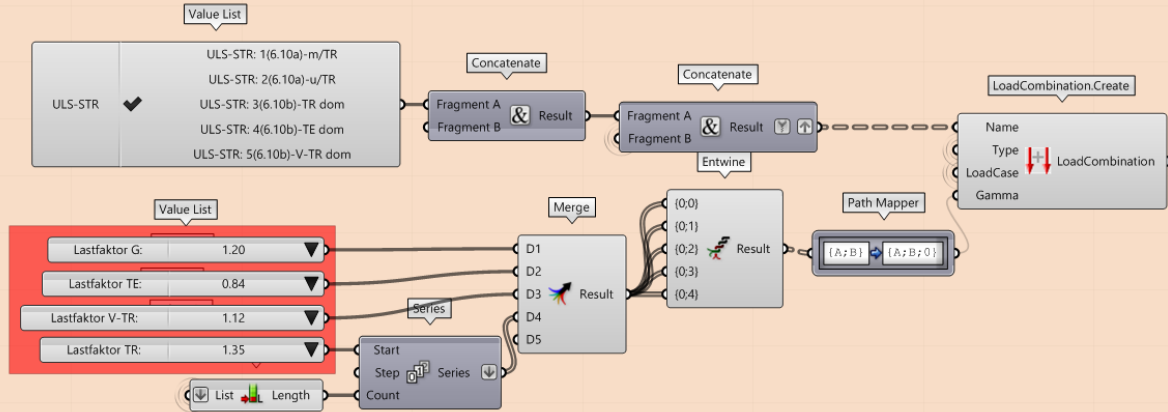
## ULS-STR 1



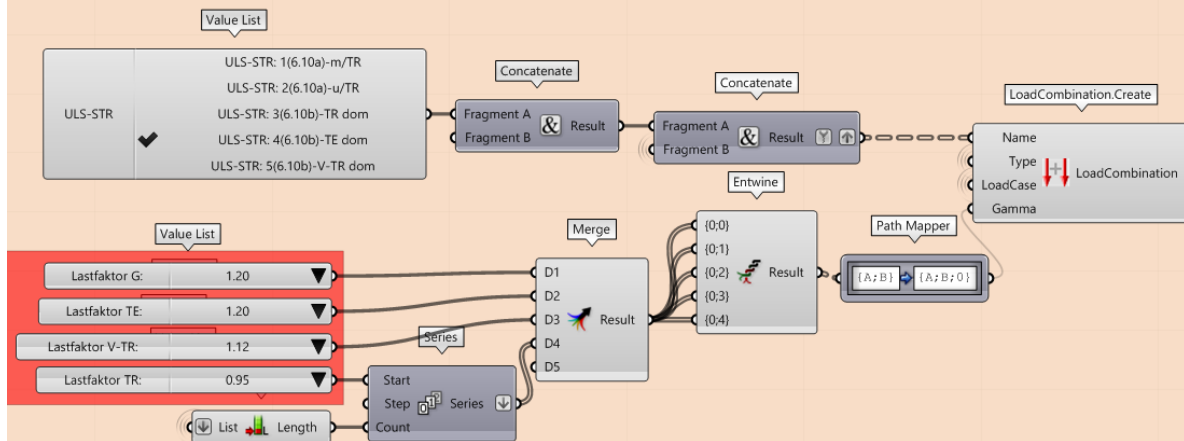
## ULS-STR 2



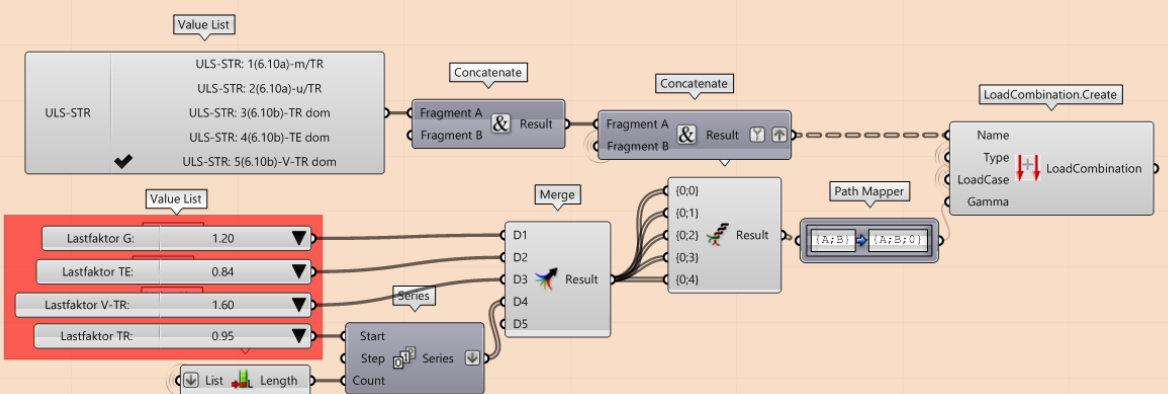
# ULS-STR 3



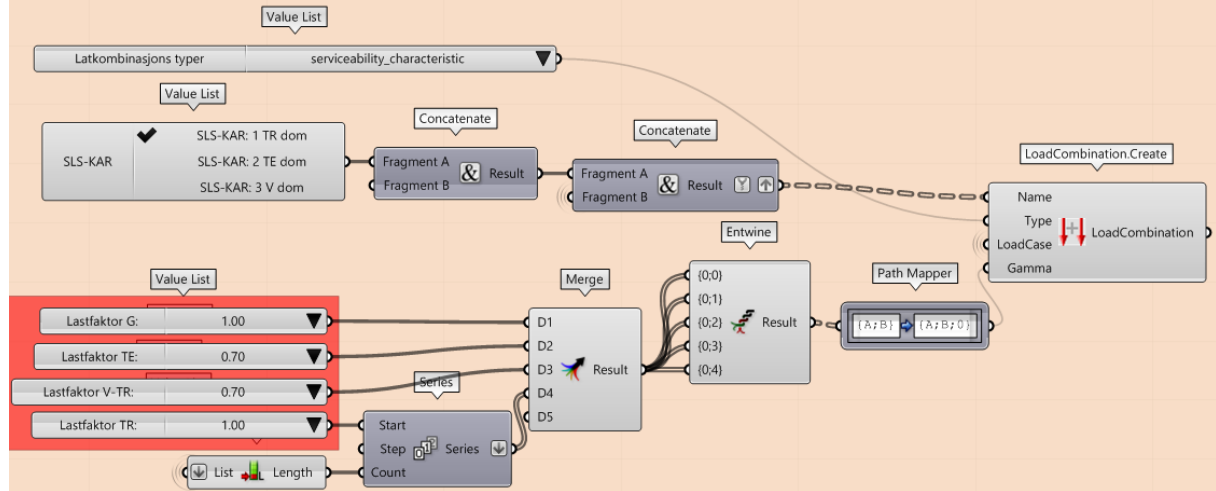
# ULS-STR 4



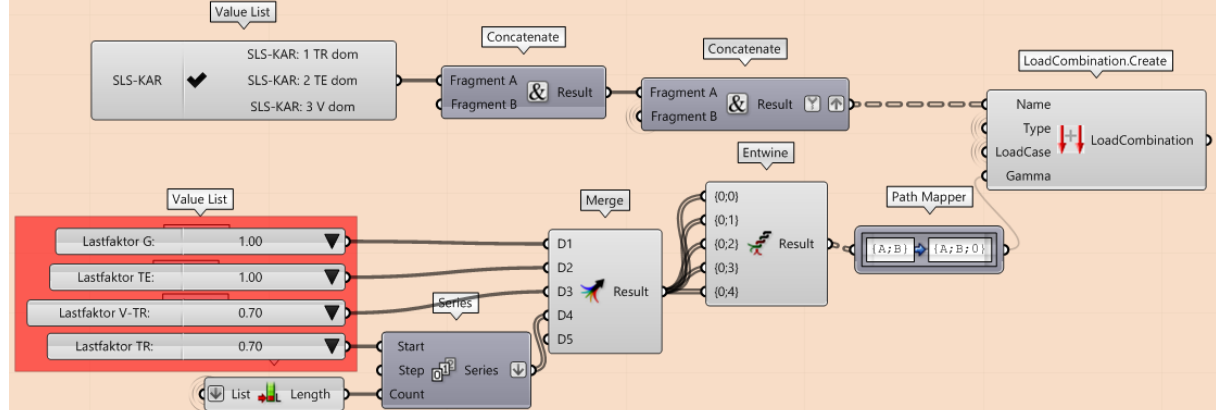
# ULS-STR 5



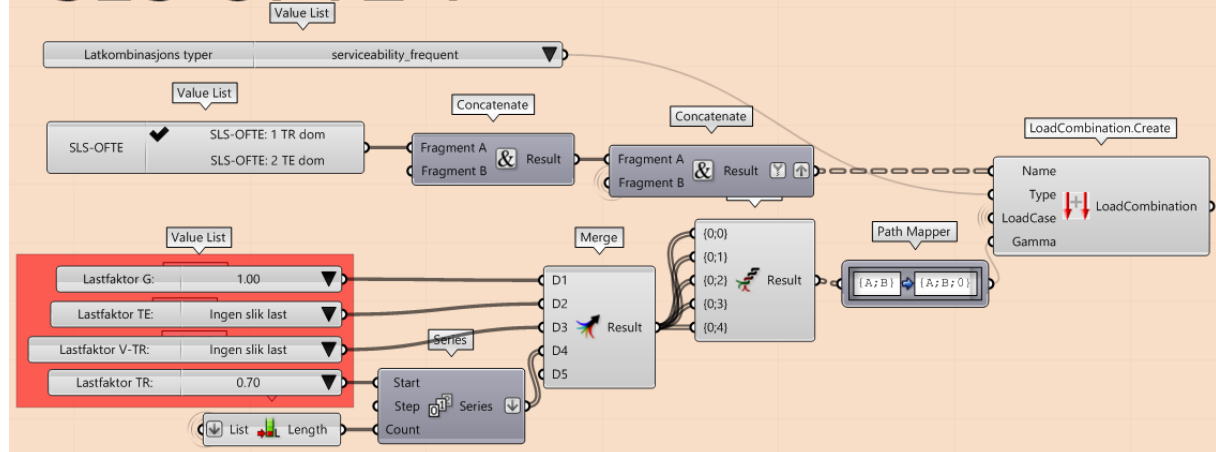
# SLS-KAR 1



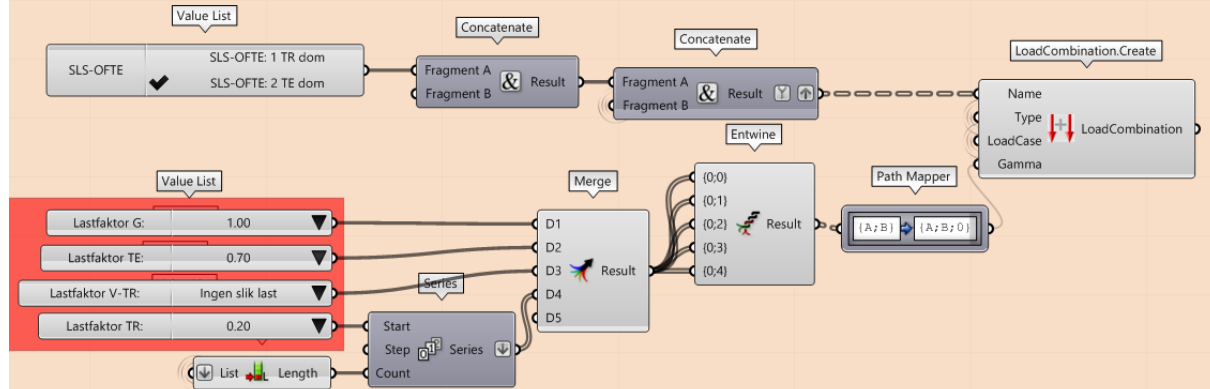
# SLS-KAR 2



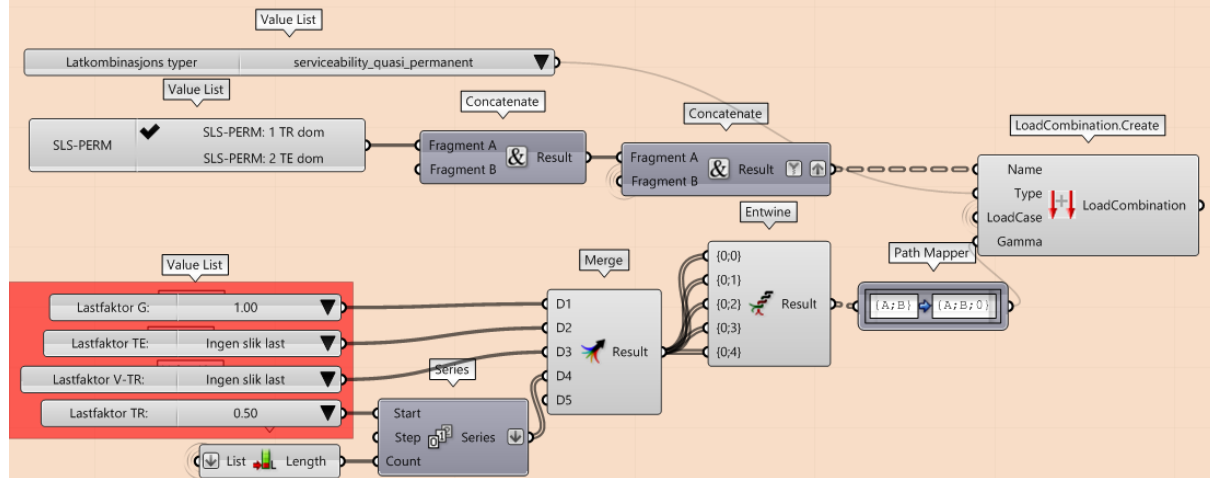
# SLS-OFTE 1



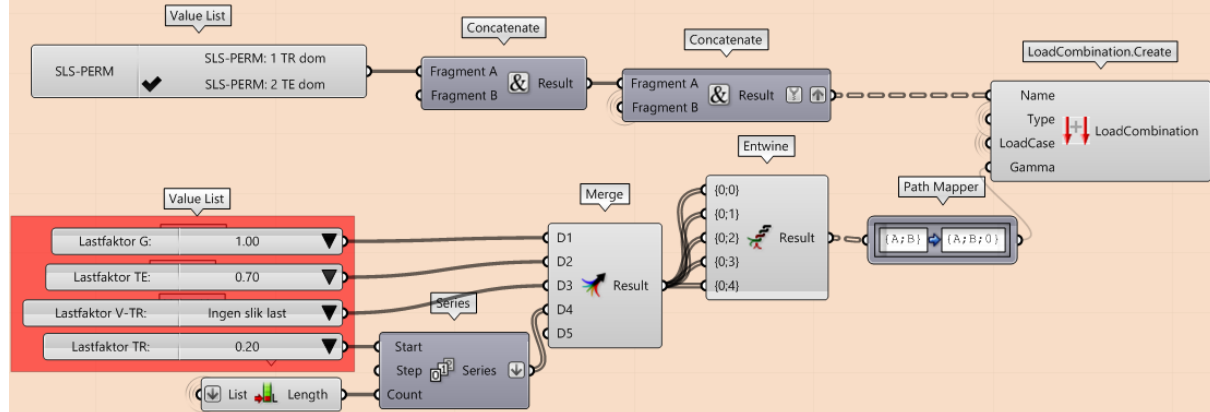
# SLS-OFFTE 2



# SLS-PERM 1



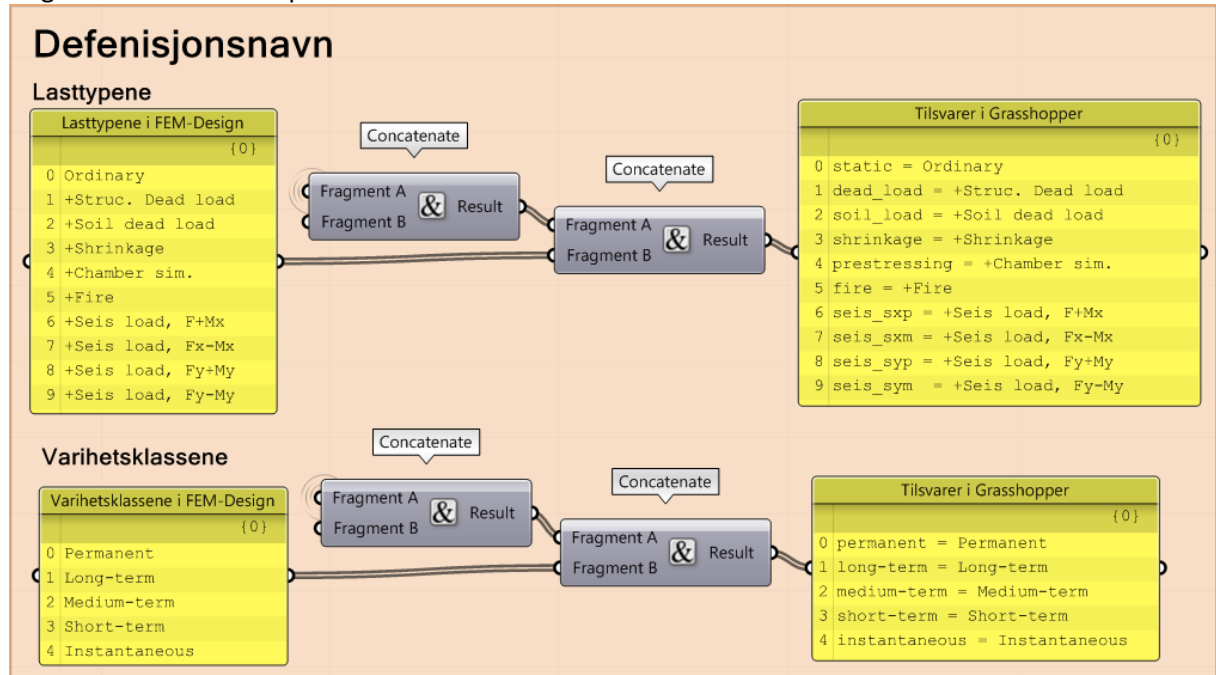
# SLS-PERM 2







Indeks-nummereringene for lasttypene og varighetsklassene til lasttilfellene, tilsvarer innholdet fra følgende lister i GH-skriptet:



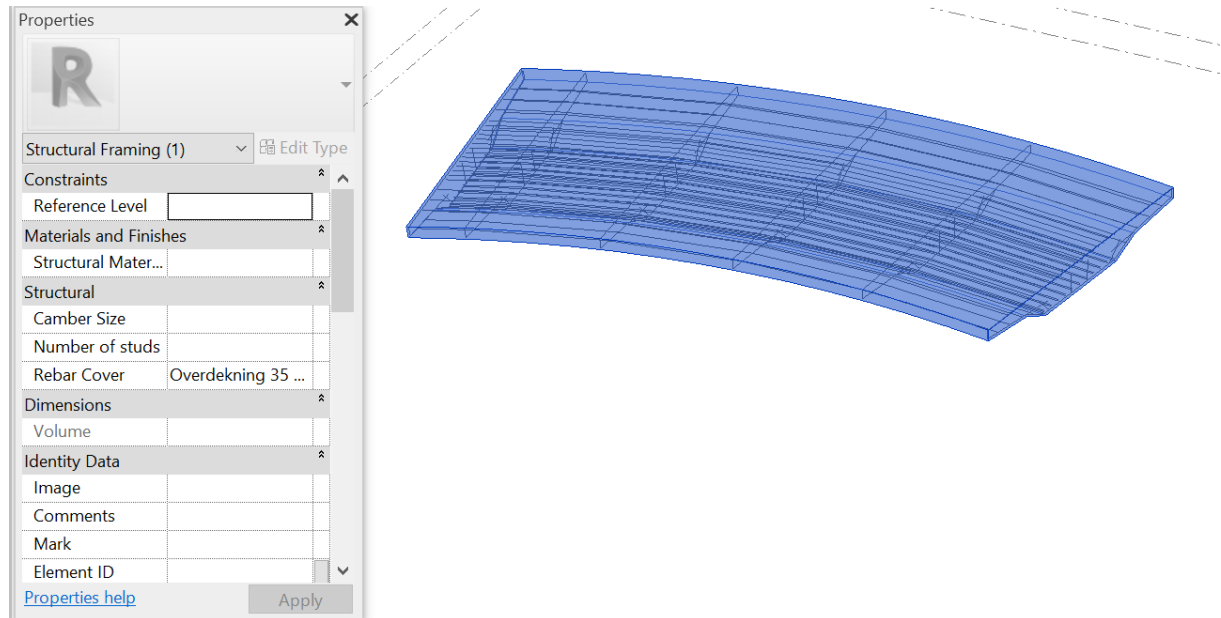
# Vedlegg J

Revit produktene

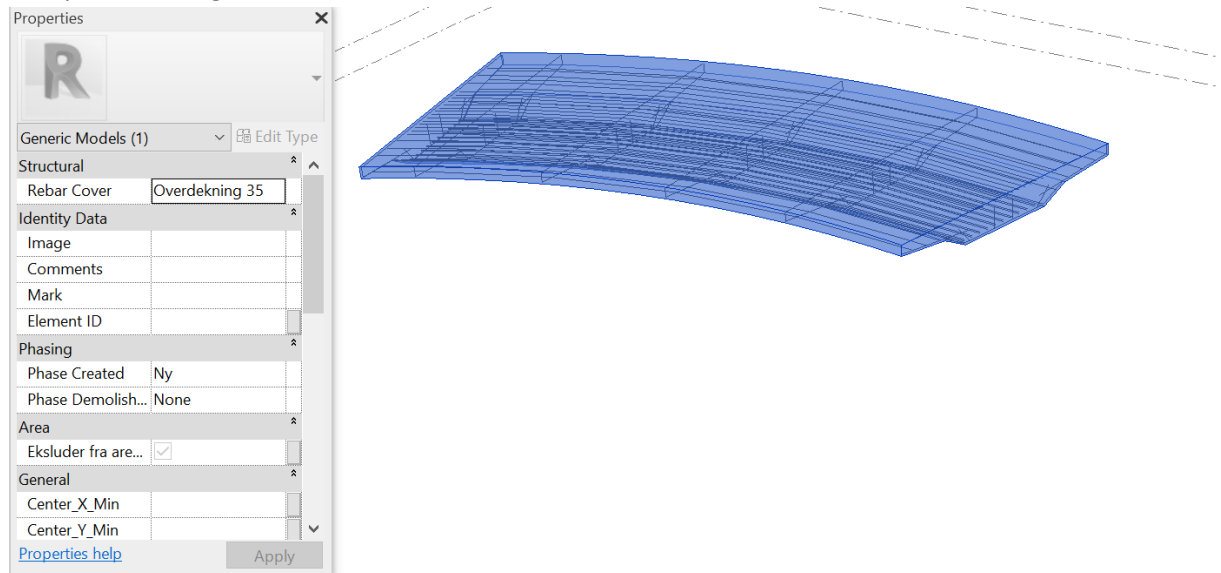
### «Egenskap palettene» til komponentutvalgene for produksjon av brudekker i RVT-modell 3.1.

Dette er referansene til parameterne, som er presentert i Tabell 7.4. Parameteren type fremkommer med «R» ikonet, mens kategori fremkommer ved siden av «Edit type» i bildefremvisningene under.

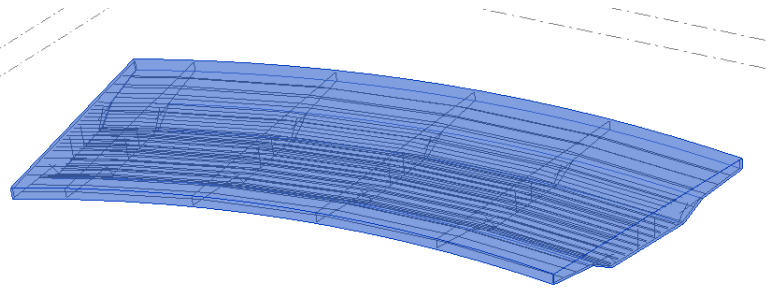
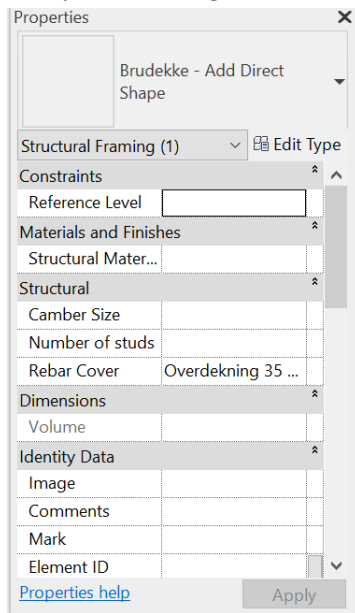
#### Komponentutvalg 3.1.1:



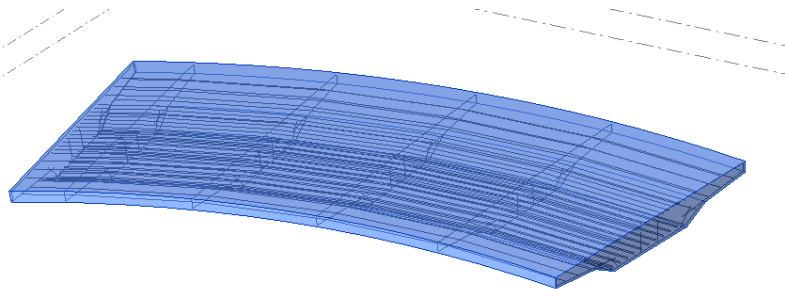
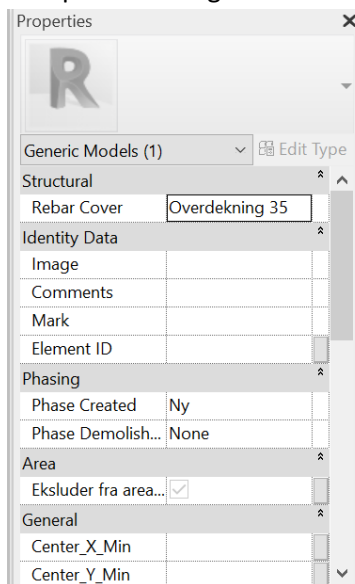
#### Komponentutvalg 3.1.2:



### Komponentutvalg 3.1.3:



### Komponentutvalg 3.1.4:



## Visnings-dybden til alle brudekkeproduktene

### Rvt-modell 1.1 – bjelke-element

View Range	<b>Nivå 1</b> ×	View Range	<b>Nivå 2</b> ×
<b>Primary Range</b>		<b>Primary Range</b>	
Top:	Level Above (Nivå 2) ▾	Offset:	0.0000
Cut plane:	Associated Level (Nivå 1) ▾	Offset:	0.1000
Bottom:	Unlimited ▾	Offset:	1.2000
<b>View Depth</b>		<b>View Depth</b>	
Level:	Unlimited ▾	Offset:	0.1000
<a href="#">Learn more about view range</a>		<a href="#">Learn more about view range</a>	
<< Show		OK Apply Cancel	


### Rvt-modell 3.1 – alle direkte formene

View Range	<b>Nivå 1</b> ×	View Range	<b>Nivå 2</b> ×
<b>Primary Range</b>		<b>Primary Range</b>	
Top:	Level Above (Nivå 2) ▾	Offset:	0.0000
Cut plane:	Associated Level (Nivå 1) ▾	Offset:	0.1000
Bottom:	Unlimited ▾	Offset:	1.2000
<b>View Depth</b>		<b>View Depth</b>	
Level:	Unlimited ▾	Offset:	0.1000
<a href="#">Learn more about view range</a>		<a href="#">Learn more about view range</a>	
<< Show		OK Apply Cancel	

# Vedlegg K

FEM-Design produktene

## Resulterende lastkombinasjoner i FEM-Design

 Load combinations

No	Name	Type	Factor	Included load cases
1	ULS-STR 1.1	U	1.35	Temperaturlast t1
			1.12	Temperaturlast t3
			0.95	Trafikklast vertikalt LM1 Q: Halve
			0.95	Trafikklast vertikalt LM1 q: Halve
2	ULS-STR 1.2	U	1.35	Temperaturlast t1
			1.12	Temperaturlast t2
			0.95	Trafikklast vertikalt LM1 Q: Hele
			0.95	Trafikklast vertikalt LM1 q: Hele
3	ULS-STR 1.3	U	1.35	Temperaturlast t1
			1.12	Temperaturlast t2
			0.95	Trafikklast vertikalt LM1 Q: Hele
			0.95	Trafikklast vertikalt LM1 q: Hele
4	ULS-STR 1.4	U	1.35	Temperaturlast t1
			1.12	Temperaturlast t4
			0.95	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.95	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
5	ULS-STR 1.5	U	1.35	Temperaturlast t1
			1.12	Temperaturlast t4
			0.95	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
6	ULS-STR 2.1	U	1.35	Temperaturlast t1
			0.00	Temperaturlast t3
			0.00	Trafikklast vertikalt LM1 Q: Halve
			0.00	Trafikklast vertikalt LM1 q: Halve
7	ULS-STR 2.2	U	1.35	Temperaturlast t1
			0.00	Temperaturlast t2
			0.00	Trafikklast vertikalt LM1 Q: Hele
			0.00	Trafikklast vertikalt LM1 q: Hele
8	ULS-STR 2.3	U	1.35	Temperaturlast t1
			0.00	Temperaturlast t2
			0.00	Trafikklast vertikalt LM1 Q: Hele
			0.00	Trafikklast vertikalt LM1 q: Hele
9	ULS-STR 2.4	U	1.35	Temperaturlast t1
			0.00	Temperaturlast t4
			0.00	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.00	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
10	ULS-STR 2.5	U	1.35	Temperaturlast t1
			0.00	Temperaturlast t4
			0.00	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.00	Trafikklast vertikalt LM1 q: Litt fra opplegg hele



11	ULS-STR 3.1	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t3
			1.35	Trafikklast vertikalt LM1 Q: Halve
			1.35	Trafikklast vertikalt LM1 q: Halve
12	ULS-STR 4.1	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t3
			0.95	Trafikklast vertikalt LM1 Q: Halve
			0.95	Trafikklast vertikalt LM1 q: Halve
13	ULS-STR 5.1	U	1.20	Temperaturlast t1
			1.60	Temperaturlast t3
			0.95	Trafikklast vertikalt LM1 Q: Halve
			0.95	Trafikklast vertikalt LM1 q: Halve
14	SLS-KAR: 1.1	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t3
			1.00	Trafikklast vertikalt LM1 Q: Halve
			1.00	Trafikklast vertikalt LM1 q: Halve
15	SLS-KAR: 2.1	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t3
			0.70	Trafikklast vertikalt LM1 Q: Halve
			0.70	Trafikklast vertikalt LM1 q: Halve
16	SLS-OFTE: 1.1	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t3
			0.70	Trafikklast vertikalt LM1 Q: Halve
			0.70	Trafikklast vertikalt LM1 q: Halve
17	SLS-OFTE: 2.1	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t3
			0.20	Trafikklast vertikalt LM1 Q: Halve
			0.20	Trafikklast vertikalt LM1 q: Halve
18	SLS-PERM: 1.1	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t3
			0.50	Trafikklast vertikalt LM1 Q: Halve
			0.50	Trafikklast vertikalt LM1 q: Halve
19	SLS-PERM: 2.1	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t3
			0.20	Trafikklast vertikalt LM1 Q: Halve
			0.20	Trafikklast vertikalt LM1 q: Halve
20	ULS-STR 3.2	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t2
			1.35	Trafikklast vertikalt LM1 Q: Hele
			1.35	Trafikklast vertikalt LM1 q: Hele

21	ULS-STR 4.2	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t2
			0.95	Trafikklast vertikalt LM1 Q: Hele
			0.95	Trafikklast vertikalt LM1 q: Hele
22	ULS-STR 5.2	U	1.20	Temperaturlast t1
			1.60	Temperaturlast t2
			0.95	Trafikklast vertikalt LM1 Q: Hele
			0.95	Trafikklast vertikalt LM1 q: Hele
23	SLS-KAR: 1.2	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t2
			1.00	Trafikklast vertikalt LM1 Q: Hele
			1.00	Trafikklast vertikalt LM1 q: Hele
24	SLS-KAR: 2.2	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t2
			0.70	Trafikklast vertikalt LM1 Q: Hele
			0.70	Trafikklast vertikalt LM1 q: Hele
25	SLS-OFTE: 1.2	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.70	Trafikklast vertikalt LM1 Q: Hele
			0.70	Trafikklast vertikalt LM1 q: Hele
26	SLS-OFTE: 2.2	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.20	Trafikklast vertikalt LM1 Q: Hele
			0.20	Trafikklast vertikalt LM1 q: Hele
27	SLS-PERM: 1.2	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.50	Trafikklast vertikalt LM1 Q: Hele
			0.50	Trafikklast vertikalt LM1 q: Hele
28	SLS-PERM: 2.2	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.20	Trafikklast vertikalt LM1 Q: Hele
			0.20	Trafikklast vertikalt LM1 q: Hele
29	ULS-STR 3.3	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t2
			1.35	Trafikklast vertikalt LM1 Q: Hele
			1.35	Trafikklast vertikalt LM1 q: Hele
30	ULS-STR 4.3	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t2
			0.95	Trafikklast vertikalt LM1 Q: Hele
			0.95	Trafikklast vertikalt LM1 q: Hele

31	ULS-STR 5.3	U	1.20	Temperaturlast t1
			1.60	Temperaturlast t2
			0.95	Trafikklast vertikalt LM1 Q: Hele
			0.95	Trafikklast vertikalt LM1 q: Hele
32	SLS-KAR: 1.3	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t2
			1.00	Trafikklast vertikalt LM1 Q: Hele
			1.00	Trafikklast vertikalt LM1 q: Hele
33	SLS-KAR: 2.3	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t2
			0.70	Trafikklast vertikalt LM1 Q: Hele
			0.70	Trafikklast vertikalt LM1 q: Hele
34	SLS-OFTE: 1.3	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.70	Trafikklast vertikalt LM1 Q: Hele
			0.70	Trafikklast vertikalt LM1 q: Hele
35	SLS-OFTE: 2.3	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.20	Trafikklast vertikalt LM1 Q: Hele
			0.20	Trafikklast vertikalt LM1 q: Hele

36	SLS-PERM: 1.3	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.50	Trafikklast vertikalt LM1 Q: Hele
			0.50	Trafikklast vertikalt LM1 q: Hele
37	SLS-PERM: 2.3	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t2
			0.20	Trafikklast vertikalt LM1 Q: Hele
			0.20	Trafikklast vertikalt LM1 q: Hele
38	ULS-STR 3.4	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t4
			1.35	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			1.35	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
39	ULS-STR 4.4	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t4
			0.95	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.95	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
40	ULS-STR 5.4	U	1.20	Temperaturlast t1
			1.60	Temperaturlast t4
			0.95	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.95	Trafikklast vertikalt LM1 q: Litt fra opplegg hele

41	SLS-KAR: 1.4	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t4
			1.00	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			1.00	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
42	SLS-KAR: 2.4	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t4
			0.70	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.70	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
43	SLS-OFTE: 1.4	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.70	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.70	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
44	SLS-OFTE: 2.4	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.20	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.20	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
45	SLS-PERM: 1.4	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.50	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.50	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
46	SLS-PERM: 2.4	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.20	Trafikklast vertikalt LM1 Q: Litt fra opplegg
			0.20	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
47	ULS-STR 3.5	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t4
			1.35	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			1.35	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
48	ULS-STR 4.5	U	1.20	Temperaturlast t1
			1.12	Temperaturlast t4
			0.95	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.95	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
49	ULS-STR 5.5	U	1.20	Temperaturlast t1
			1.60	Temperaturlast t4
			0.95	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.95	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
50	SLS-KAR: 1.5	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t4
			1.00	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			1.00	Trafikklast vertikalt LM1 q: Litt fra opplegg hele

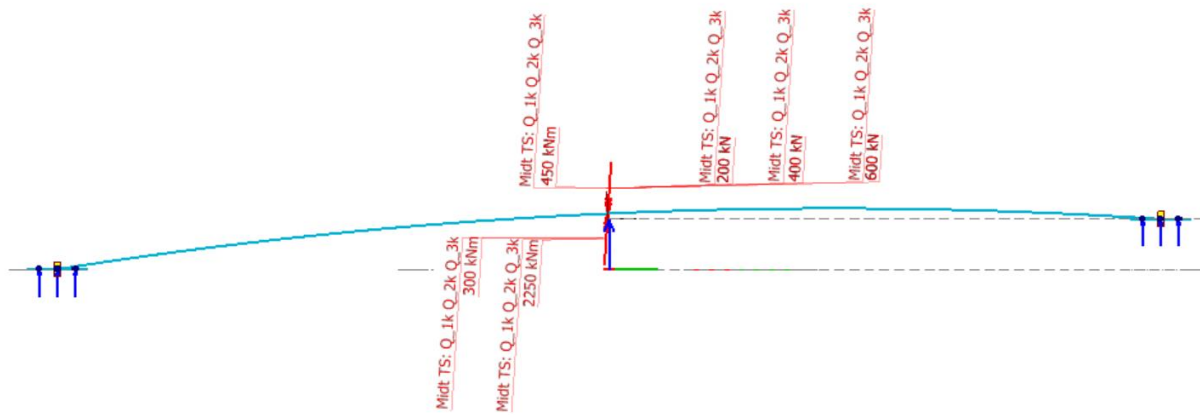
51	SLS-KAR: 2.5	Sc	1.00	Temperaturlast t1
			0.70	Temperaturlast t4
			0.70	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.70	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
52	SLS-OFTE: 1.5	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.70	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.70	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
53	SLS-OFTE: 2.5	Sf	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.20	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.20	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
54	SLS-PERM: 1.5	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.50	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.50	Trafikklast vertikalt LM1 q: Litt fra opplegg hele
55	SLS-PERM: 2.5	Sq	1.00	Temperaturlast t1
			0.00	Temperaturlast t4
			0.20	Trafikklast vertikalt LM1 Q: Litt fra opplegg hele
			0.20	Trafikklast vertikalt LM1 q: Litt fra opplegg hele

### Resulterende laster på brudekket i FEM-Design

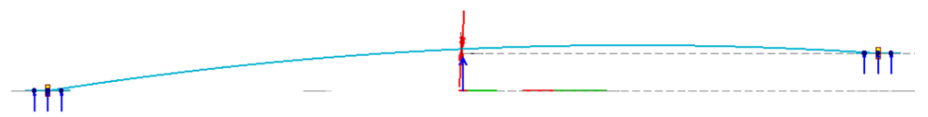
Trafikk:

TS

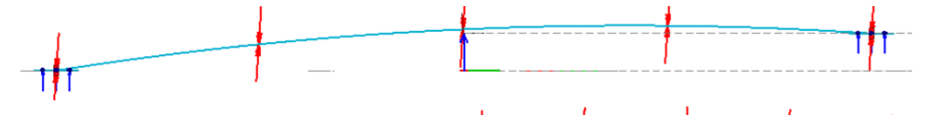




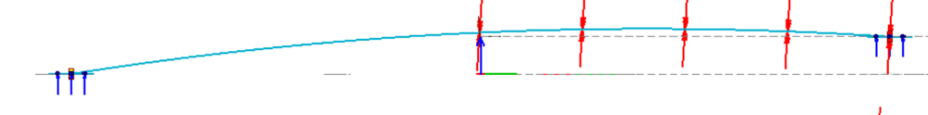
TS Midt



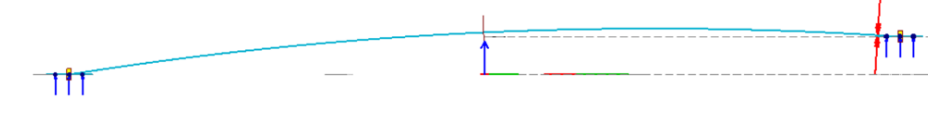
TS Hele



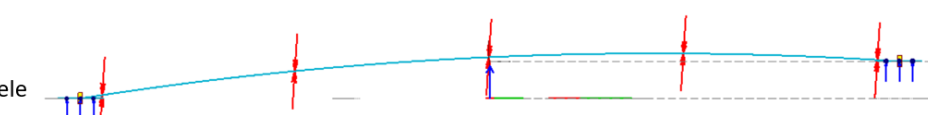
TS Halve



TS Litt fra opplegg

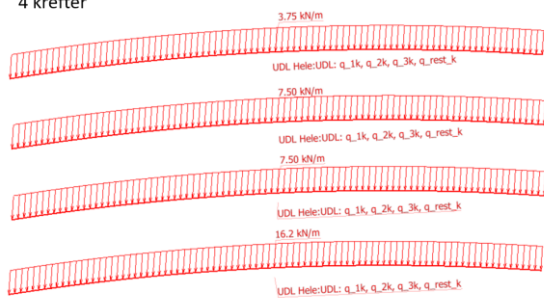


TS Litt fra opplegg hele

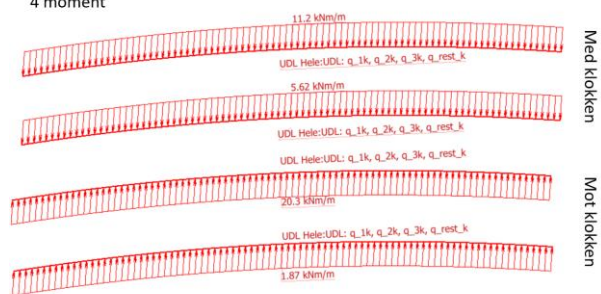


UDL

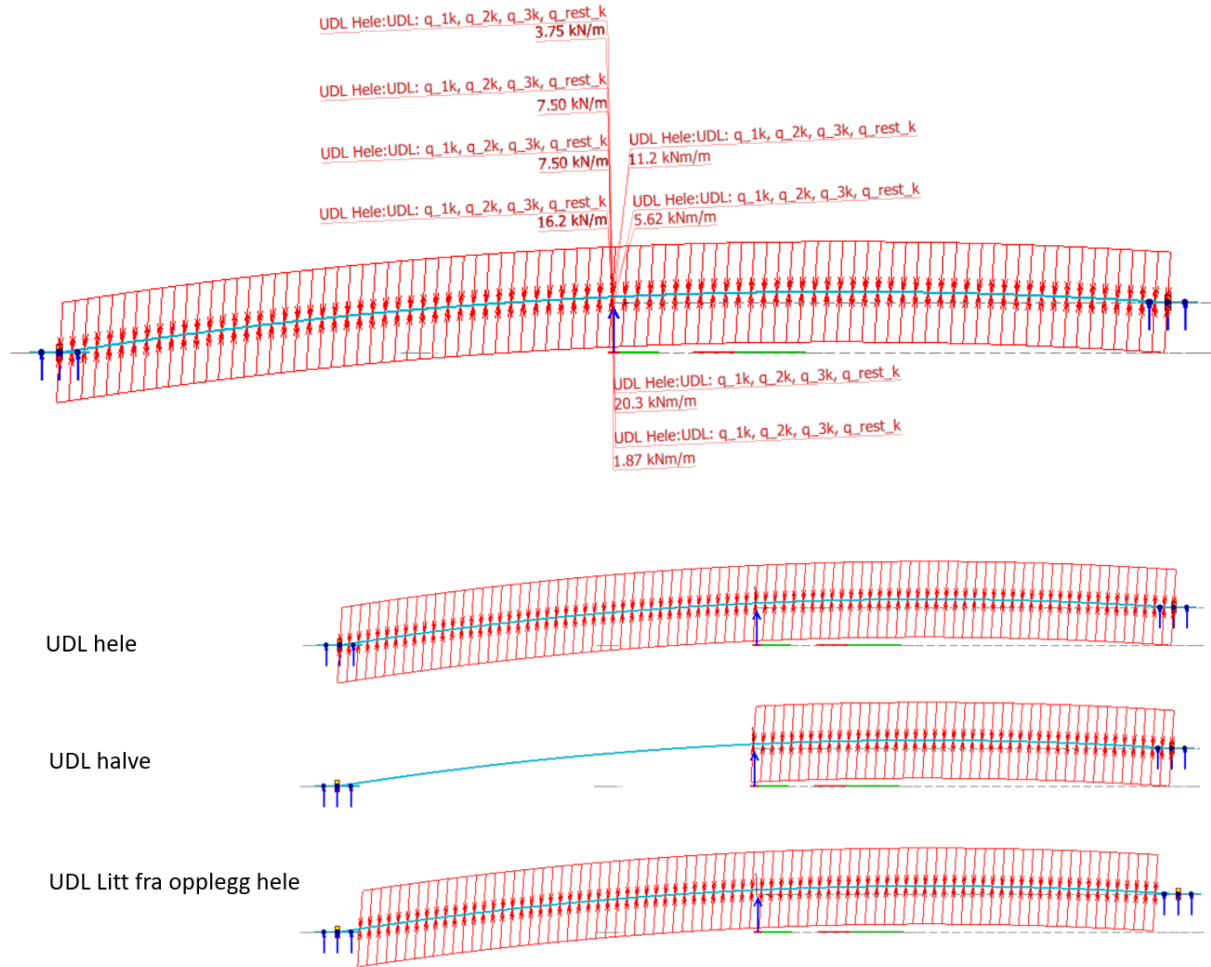
4 krefter



4 moment



Med klokken  
Mot klokken



**Resulterende beregningstid for FEM-modell 3.1 av de 4 tilfellene av brudekkeutforming**

Design calculation type: «Autodesign»

Tilfelle 1	Tilfelle 2	Tilfelle 3	Tilfelle 4
<p>Calculation - Finished</p> <ul style="list-style-type: none"> <li>Stiffness matrix calculation</li> <li>Load calculation</li> <li>Stiffness matrix factorization</li> <li>Solution of load cases/combinations</li> <li>Internal forces of cases/combinations</li> <li>Average internal forces</li> <li>Internal forces of combinations (1st order)</li> <li>Validating of results</li> <li>Calculating converted load results</li> <li>Scaling load case results</li> <li>Scaling load combination results</li> </ul> <p>Calculation finished. Total calculation time: 0 : 00 : 11</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>	<p>Calculation - Finished</p> <ul style="list-style-type: none"> <li>Load calculation</li> <li>Stiffness matrix factorization</li> <li>Solution of load cases/combinations</li> <li>Internal forces of cases/combinations</li> <li>Average internal forces</li> <li>Internal forces of combinations (1st order)</li> <li>Validating of results</li> <li>Calculating converted load results</li> <li>Scaling load case results</li> <li>Scaling load combination results</li> <li>Bar reinforcement calculation</li> </ul> <p>Calculation finished. Total calculation time: 0 : 03 : 37</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>	<p>Calculation - Finished</p> <ul style="list-style-type: none"> <li>Load calculation</li> <li>Stiffness matrix factorization</li> <li>Solution of load cases/combinations</li> <li>Internal forces of cases/combinations</li> <li>Average internal forces</li> <li>Internal forces of combinations (1st order)</li> <li>Validating of results</li> <li>Calculating converted load results</li> <li>Scaling load case results</li> <li>Scaling load combination results</li> <li>Bar reinforcement calculation</li> </ul> <p>Calculation finished. Total calculation time: 0 : 04 : 42</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>	<p>Calculation - Finished</p> <ul style="list-style-type: none"> <li>Load calculation</li> <li>Stiffness matrix factorization</li> <li>Solution of load cases/combinations</li> <li>Internal forces of cases/combinations</li> <li>Average internal forces</li> <li>Internal forces of combinations (1st order)</li> <li>Validating of results</li> <li>Calculating converted load results</li> <li>Scaling load case results</li> <li>Scaling load combination results</li> <li>Bar reinforcement calculation</li> </ul> <p>Calculation finished. Total calculation time: 0 : 04 : 13</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>

Design calculation type: «Check»

Tilfelle 1	Tilfelle 2	Tilfelle 3	Tilfelle 4
<p>Calculation - Finished</p> <p>Load calculation Stiffness matrix factorization Solution of load cases/combinations Internal forces of cases/combinations Average internal forces <b>Internal forces of combinations (1st order)</b> Validating of results Calculating converted load results Scaling load case results Scaling load combination results Bar reinforcement calculation</p> <p>Calculation finished. Total calculation time: 0 : 00 : 12</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>	<p>Calculation - Finished</p> <p>Load calculation Stiffness matrix factorization Solution of load cases/combinations Internal forces of cases/combinations Average internal forces <b>Internal forces of combinations (1st order)</b> Validating of results Calculating converted load results Scaling load case results Scaling load combination results Bar reinforcement calculation</p> <p>Calculation finished. Total calculation time: 0 : 00 : 45</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>	<p>Calculation - Finished</p> <p>Load calculation Stiffness matrix factorization Solution of load cases/combinations Internal forces of cases/combinations Average internal forces <b>Internal forces of combinations (1st order)</b> Validating of results Calculating converted load results Scaling load case results Scaling load combination results Bar reinforcement calculation</p> <p>Calculation finished. Total calculation time: 0 : 00 : 48</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>	<p>Calculation - Finished</p> <p>Load calculation Stiffness matrix factorization Solution of load cases/combinations Internal forces of cases/combinations Average internal forces <b>Internal forces of combinations (1st order)</b> Validating of results Calculating converted load results Scaling load case results Scaling load combination results Bar reinforcement calculation</p> <p>Calculation finished. Total calculation time: 0 : 00 : 11</p> <p>Computational warning messages: Equilibrium error. Large nodal displacement or rotation was found.</p> <p>Close</p>

**Resulterende armering beregnet med «Check» og «AutoDesign» for de fire tilfellene**

Tilfelle 1:

*AutoDesign*

Auto design

Calculation...  
Load combinations

Display table

Utilization

Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
Brudekke.1.1	d = 16, 8, 8, ...	18.564	302	302
Tverrbærer.1.1	d = 16, 8, 8, ...	2.510	289	289
Tverrbærer.2.1	d = 16, 8, 8, ...	1.660	202	202

Show only visible objects in the table

Parameters Design Delete Show details >

*Check*

Auto design

Calculation...  
Load combinations

Display table

Utilization

Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
Brudekke.1.1	d = 16, 8, 8, ...	-	-	-
Tverrbærer.1.1	d = 16, 8, 8, ...	-	-	-
Tverrbærer.2.1	d = 16, 8, 8, ...	-	-	-

Show only visible objects in the table

Parameters Design Delete Show details >

Tilfelle 2:

*AutoDesign*

Auto design

Calculation...  
Load combinations

Display table

Utilization

Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
Tverrbærer.1.1	d = 16, 8, 8, ...	2.487	305	305
Tverrbærer.2.1	d = 16, 8, 8, ...	1.765	212	212

Show only visible objects in the table

Parameters Design Delete Show details >

*Check*

Auto design

Calculation...  
Load combinations

Display table

Utilization

Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
Tverrbærer.1.1	d = 16, 8, 8, ...	-	-	-
Tverrbærer.2.1	d = 16, 8, 8, ...	-	-	-

Show only visible objects in the table

Parameters Design Delete Show details >

Tilfelle 3:



### AutoDesign

Auto design

Calculation...  
Load combinations

Display table

Utilization

	Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
⊖	Tverrbærer.1.1	d = 16, 8, 8, ...	2.434	293	293
⊖	Tverrbærer.2.1	d = 16, 8, 8, ...	1.814	209	209

Show only visible objects in the table

Parameters Design Delete Show details >

### Check

Auto design

Calculation...  
Load combinations

Display table

Utilization

	Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
	Tverrbærer.1.1	d = 16, 8, 8, ...	-	-	-
	Tverrbærer.2.1	d = 16, 8, 8, ...	-	-	-

Show only visible objects in the table

Parameters Design Delete Show details >

### Tilfelle 4:

### AutoDesign

Auto design

Calculation...  
Load combinations

Display table

Utilization

	Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
⊖	Tverrbærer.1.1	d = 16, 8, 8, ...	2.421	291	291
⊖	Tverrbærer.2.1	d = 16, 8, 8, ...	1.840	202	202

Show only visible objects in the table

Parameters Design Delete Show details >

### Check

Auto design

Calculation...  
Load combinations

Display table

Utilization

	Group	Design parameters	Total weight [t]	Max. [%]	Min. [%]
	Tverrbærer.1.1	d = 16, 8, 8, ...	-	-	-
	Tverrbærer.2.1	d = 16, 8, 8, ...	-	-	-

Show only visible objects in the table

Parameters Design Delete Show details >