

Visuell programmering ved design av fagverk i tre

Bruk av visuell programmering og parametrisk design for
effektivisering av den tradisjonelle dimensjoneringsprosessen



MARGUNN HELLELAND



DAN HELLELAND STRØMSNES

VEILEDER

Jonas Høgli Major, UiA
Jonas Ottersland, Sweco
Kjetil Nodeland Frøysnes, Sweco

Universitetet i Agder, 2023

Fakultet for teknologi og realfag
Institutt for ingeniørvitenskap



Obligatorisk egenerklæring/gruppeerklæring

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/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.

2. Jeg/vi erklærer videre at denne besvarelsen:
 - ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.
 - ikke refererer til andres arbeid uten at det er oppgitt.
 - ikke refererer til eget tidligere arbeid uten at det er oppgitt.
 - har alle referansene oppgitt i litteraturlisten.
 - ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.

3. Jeg/vi 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.

4. Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.

5. Jeg/vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.

6. Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.

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 tausehetsbelagt/konfidensiell vil ikke bli publisert.

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

JA NEI

Er oppgaven båndlagt (konfidensiell)?

JA NEI

(Båndleggingsavtale må fylles ut)

- 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 Institutt for Ingeniørvitenskap som en del av masterprogrammet for konstruksjonsteknikk ved Universitet i Agder. Masteroppgaven er den avsluttende oppgaven i emnet BYG508 og ble utarbeidet i det fjerde og siste semesteret.

Vi vil benytte anledningen til å takke vår interne veileder ved Universitet i Agder, Jonas Høgli Major, for engasjement og oppfølging. Videre har også de andre lektorene ved universitetet, med spesialisering innen tre- og stålkonstruksjoner, måttet sette av rikelig med tid for å besvare spørsmål. Vi ønsker derfor også å rette en stor takk til Preben Aanensen og Vidar Hellum for sin generøsitet. Til slutt ønsker vi å takke våre eksterne veiledere fra Sweco, Kjetil Nodeland Frøysnes, Jonas Ottersland og Oscar Widestam, for alt engasjement, oppfølging og faglige innsikt.

Målet med masteroppgaven er å fastslå om bruk av visuell programmering og parametrisk design kan effektivisere den tradisjonelle dimensjoneringsprosessen for trekonstruksjoner. Derfor er det ønskelig å identifisere både fordeler og utfordringer knyttet til designmetoden og verktøyene.

Summary

This master thesis deals with the use of visual programming and parametric design. The aim is to determine whether this can make the traditional process for structural design more efficient (for wooden structures). The traditional process is based on several manual transfers between software, and it is particularly vulnerable to changes that occur along the way. However, when using parametric design, one uses algorithms and formulas to make the model-objects dependent on each other. Thus, small changes will not require extensive modifications to surrounding components in the model.

The task is based on an existing glulam-truss, which has already been designed by the client (Sweco). The reports research question is: "How does visual programming in Grasshopper challenge the traditional method for designing glulam-trusses?".

To answer this, a system has been programmed in Grasshopper, which both performs linear elastic analysis and checks according to Eurocode for ULS and connections. When analyzing the result, it could be concluded that visual programming in Grasshopper makes it possible to build a more comprehensive, integrated system for the structural design process. It will remove unnecessary manual processes when transferring between software, and it may simplify the optimization process. On the other hand, the system has limitations because it is very resource-intensive, and it has great potential for improvement when it comes to the use of wood as a building material.

Innholdsfortegnelse

Obligatorisk egenerklæring/gruppeerklæring	i
Publiseringsavtale	ii
Forord	iii
Summary	iv
Figurliste.....	xi
Tabelliste	xviii
1 Innledning	1
2 Samfunnsperspektiv	2
2.1 Tre som byggemateriale.....	3
2.2 Digitalisering.....	3
3 Kunnskapsbakgrunn.....	5
3.1 Visuell programmering.....	5
3.1.1 Parametrisk design	6
3.1.2 Parametrisk design og BIM.....	7
3.1.3 Parametrisk modellering og parametrisk design.....	8
3.1.4 Grasshopper	9
3.1.5 Tilleggsverktøy i Grasshopper	16
3.2 Tre som byggemateriale.....	18
3.2.1 Fordeler med limtre sammenlignet med tradisjonelt konstruksjonstrevirke	18
3.2.2 Mikro- og makrostruktur.....	18
3.2.3 Spesielle forhold som må tas hensyn til ved bruk av limtre.....	19
3.2.4 Styrkeklasser	21
3.2.5 Limtres fasthet og stivhet.....	21
3.3 Fagverk	24
3.4 Dimensjonering i henhold til Eurokode.....	25
3.4.1 Grensetilstandsmetoden og partialfaktormetoden	25
3.4.2 Laster og lastkombinasjoner	26
3.4.3 Lastvirkning	26
3.4.4 Klimaklasser.....	27
3.4.5 Lastvarighetsklasser	27
3.4.6 Modifikasjonsfaktor <i>k_{mod}</i>	28
3.4.7 Bruddgrensekontroll	29
3.4.8 Bruksgrensekontroll	32

3.5	Forbindelser	32
3.5.1	Forbindelser i trekonstruksjon	32
3.5.2	Fagverksforbindelser	33
3.5.3	Forbindelser mellom stav og gurt	34
3.5.4	Stavdybler.....	35
3.5.5	Taueffekt	35
3.5.6	Bruddmekanismer i treet	36
3.5.7	Flersnittet forbindelse med stålplater.....	42
3.6	Optimalisering av fagverk.....	44
3.6.1	Optimalisering	44
3.6.2	Multiobjektiv optimalisering	48
3.7	Tidligere litteratur med bruk av Grasshopper	49
3.7.1	Begrensinger ved parametrisk design	49
3.7.2	Design av stålkonstruksjoner med gjenbrukbare elementer	51
3.7.3	Parametrisk design og analyse av trekonstruksjoner ved bruk av Beaver	53
3.7.4	Generativ design til BIM-arbeidsflyt for vekt optimalisering	56
3.7.5	Parametrisk design av fagverksbuer	57
4	Forskerspørsmål	59
4.1	Spesifikasjoner	59
4.2	Avgrensninger	59
5	Case/Materialer	60
5.1	Case	60
5.1.1	Geometri	60
5.1.2	Randbetingelser	61
5.1.3	Laster	61
5.1.4	Materialkvalitet.....	64
5.1.5	Resultater	64
5.2	Materialer	66
5.2.1	Rhinoceros 7 SR28.....	66
5.2.2	Grasshopper	66
5.2.3	Autodesk Robot.....	66
5.2.4	Microsoft Office Excel	67
6	Metode	68
6.1	Etablering av bakgrunnskunnskap	68

6.2	Datainnsamling	70
6.2.1	Krister for valg av litteratur	70
6.2.2	Induktiv analyse	71
6.2.3	Deduktiv analyse	72
6.3	Tilleggsverktøy til Grasshopper	72
6.3.1	Nedlastede plug-ins.....	72
6.4	Etablering av modell i Grasshopper	75
6.4.1	Oppbygging av pinnemodell.....	75
6.4.2	Styring av parametere	81
6.4.3	Fra linje- til bjelkeelementer	81
6.4.4	Material- og tversnittsegenskaper	83
6.4.5	Opplegg og forbindelser	88
6.4.6	Laster.....	89
6.4.7	Karamba-analyse.....	97
6.4.8	Utnyttelsesanalyse av elementer i Grasshopper.....	98
6.4.9	Beregning av forbindelser i Grasshopper	112
6.4.10	Modellering av forbindelser med stålplater og dybler.....	137
6.5	Metode for optimalisering	166
6.5.1	Metode for størrelsesoptimalisering av limtrefagverket	167
7	Resultat.....	173
7.1	Parametrisk modell	173
7.1.1	Endring av strukturen til fagverket.....	173
7.1.2	Endringer av tversnitt	173
7.1.3	Endringer i forbindelse	174
7.2	Største belastningskrefter på hvert element	178
7.2.1	Oppleggs-krefter for alle lastkombinasjonene fra Karamba3D.....	178
7.2.2	Overgurt	179
7.2.3	Undergurt.....	180
7.2.4	Vertikale staver	181
7.2.5	Diagonal 0.....	182
7.2.6	Diagonal 1.....	183
7.2.7	Diagonal 2.....	184
7.2.8	Diagonal 3.....	185
7.2.9	Diagonal 4.....	185

7.3	Resultater moment-, skjær-, og aksialkraftdiagram.....	186
7.3.1	Momentkraftdiagram.....	186
7.3.2	Skjærkraftdiagram.....	187
7.3.3	Normalkraftdiagram.....	187
7.4	Resultater fra dimensjonering av elementer	187
7.4.1	Overgrut	188
7.4.2	Undergurt.....	189
7.4.3	Vertikale staver	190
7.4.4	Diagonal 0.....	191
7.4.5	Diagonal 1.....	192
7.4.6	Diagonal 2.....	193
7.4.7	Diagonal 3.....	194
7.4.8	Diagonal 4.....	195
7.5	Resultater for forbindelsene	196
7.5.1	Forbindelse toppunkt.....	197
7.5.2	Forbindelse overgurt, diagonal 1 og 2.....	198
7.5.3	Forbindelse overgrut, diagonal 3 og 4.....	201
7.5.4	Forbindelse overgrut og vertikal stav.....	206
7.5.5	Forbindelse undergurt diagonal 0 og 1	208
7.5.6	Forbindelse undergurt, diagonal 2 og 3	209
7.5.7	Forbindelse undergurt diagonal og vertikal stav.....	210
7.5.8	Forbindelse under- og overgrut	211
7.6	Utnyttelse av stålet i forbindelsene	213
7.6.1	Forbindelse i toppunkt	214
7.6.2	Forbindelse overgurt, diagonal 1 og 2.....	214
7.6.3	Forbindelse overgurt, diagonal 3 og 4.....	215
7.6.4	Forbindelse overgurt og vertikal stav.....	215
7.6.5	Knutepunkt undergurt med diagonal 0 og 1	216
7.6.6	Knutepunkt undergurt med diagonal 2 og 3	216
7.6.7	Knutepunkt undergurt med diagonal 4 og vertikal stav.....	216
7.6.8	Forbindelse over- og undergurt	217
7.7	Optimalisering av limtrefagverket.....	217
7.7.1	Optimaliserte tverrsnitt.....	217
8	Diskusjon.....	219

8.1	Sammenligning av krefter	219
8.1.1	Skjærkraft	219
8.1.2	Momentkraft y-retning	220
8.1.3	Momentkraft z-retning.....	221
8.1.4	Trykkraft	222
8.1.5	Strekraft.....	223
8.1.6	Sammenligningen av oppleggs-krefter.....	223
8.1.7	Sammenligning med andre studier	224
8.2	Cluster-komponent for elementanalyse	225
8.2.1	Dimensjonerende last for beregning av utnyttelse.....	225
8.2.2	Nødvendige elementkontroller i fagverket	226
8.2.3	Sammenligning av utnyttelse	228
8.2.4	Utnyttelse av fagverk	231
8.3	Cluster-komponent for knutepunkts beregninger	232
8.3.1	Belastningskrefter i knutepunkt	232
8.3.2	Sammenligning av utnyttelser.....	233
8.3.3	Kontroll av stålplate	236
8.4	Valg av metode	238
8.4.1	Materialinnhenting	238
8.4.2	Kontroll i henhold til Eurokode	239
8.4.3	Ressurstung programvare	240
8.5	Brukbarhet	240
8.5.1	Gjenbruk.....	240
8.5.2	Forbedringspotensialer	241
8.5.3	Dokumentasjon	244
8.5.4	Kan man stole på resultatene?.....	244
8.6	Optimalisering av fagverket.	245
8.6.1	Optimalisering for størrelse, form og topologi.....	245
8.6.2	Optimaliseringen av limtrefagverket.....	246
8.6.3	Problemer og eventuelle løsninger av optimaliseringen.....	247
8.6.4	Sammenligning av optimaliseringsresultat med case-fagverket.....	248
8.6.5	Sammenligning med andre optimaliseringer	249
8.6.6	Algoritmen for optimalisering	249
9	Konklusjon	250

10	Anbefalinger	251
10.1	Anbefalinger før bruk av systemet	251
10.2	Anbefalinger ved bruk av systemet	251
10.3	Anbefalinger for videre utvikling av systemet	251
10.4	Andre anbefalinger	252
11	Referanser	253
12	Vedlegg	264

Figurliste

Figur 3.1: Illustrerer både tekstprogrammering og visuell programmering [35]	5
Figur 3.2: Kjedemodellen av Colònia Güell kapellet [7]	6
Figur 3.3 Overflaten fra såpefilm og banene fra fuktig ull [44]	7
Figur 3.4 Eksempler på enkelte parametere i Grasshopper	10
Figur 3.5: Komponenter "Construct Point"	11
Figur 3.6: Komponenter med taggete kant	11
Figur 3.7: Verktøytips fra komponenten "Construct Point"	11
Figur 3.8: Zoomable User Interface (ZUI) på "Addition" og «List Item» komponentene	12
Figur 3.9: Illustrasjon av forskjellige ledninger i Grasshopper	12
Figur 3.10: Objekt farger illustrert på enkelte komponenter og parametere	13
Figur 3.11: Plassering av element og indeks i en liste [35]	13
Figur 3.12: Illustrasjon av forskjellen mellom en liste og datatre [35]	14
Figur 3.13: Datatre for "Divide Curve" med "Panel"	14
Figur 3.14: "Flatten" av en liste [35]	15
Figur 3.15: "Grafting" av en liste [35]	15
Figur 3.16: "Simplify" av en liste [35]	15
Figur 3.17: Illustrasjon av "Flatten", "Graft" og "Simplify" i menyen og ikonene [35]	16
Figur 3.18: Komponenter Galapagos	17
Figur 3.19 De tre ortogonale retningene: lengderetning, radiell retning og tangentiell retning [22] ...	18
Figur 3.20 Skjematisk fremstilling av hvordan densitet påvirker treets egenskaper [57]	20
Figur 3.21 Symmetrisk kombinert-, asymmetrisk kombinert og homogent limtrevernsnitt [57]	21
Figur 3.22 Styrkeklasser for kombinert limtre [58]	21
Figur 3.23 Trevirkets ulike strekkfastheter [22]	22
Figur 3.24 Trevirkets ulike trykkfastheter [22]	22
Figur 3.25 Trevirkets bøyefasthet [22]	23
Figur 3.26 Trevirkets ulike skjærfastheter [22]	23
Figur 3.27 Illustrasjon av middelværdi og 5% fraktil i normalfordeling [22]	24
Figur 3.28 Stabiliserende tiltak: Kryssende strekkstag eller gjennomgående trekantmønster [58]	24
Figur 3.29 Eksempler på ulike fagverksformer. (a) Fagverk med parallelle gurter, (b) saltaksformet fagverk og (c) fagverk med krum gurt (Limtreboka)	25
Figur 3.30 Eksempel fagverksmodell [22]	26
Figur 3.31 Momentfritt ledd [64]	26
Figur 3.32 Eksempel Rammemodell [22]	26
Figur 3.33 Bøyingsstiv forbindelse [64]	26
Figur 3.34 Eksempel Blandet modell [22]	27
Figur 3.35 Staver leddet til gurtene [64]	27
Figur 3.36: Én riktig og flere utformingsfeil av et fagverksknutepunkt med stålplater og dybler [58].	34
Figur 3.37: Potensielle bruddmekanismer i en dybelforbindelse, basert på belastningen i forbindelsen [81]	37
Figur 3.38: De fem potensielle bruddmekanismer for dybelforbindelser [83].	37
Figur 3.39: Eksempel på hullkantsvikt [84].	38
Figur 3.40: Eksempel på splittingsbrudd [84]	38

Figur 3.41: Illustrasjon av skjærblokkbrudd og skjærpluggbrudd, pluss illustrasjon av svikt i tre plan [77, 79].	39
Figur 3.42: Sammenligning av lineært elastisk fjærsystem og brudd i tre-elementet [77].	40
Figur 3.43: Flere eksempler på radskjærbrudd [78, 87].	40
Figur 3.44: Eksempel på strekkbrudd [87, 91].	41
Figur 3.45: Bruddformer for tre-mot-tre og stål-mot-tre [57].	42
Figur 3.46: Bruddformene fra EK5 og effektiv tykkelse beregning [79, 94].	43
Figur 3.47: Størrelsesoptimalisering av fagverk [103].	45
Figur 3.48: Fritt opplagt fagverk [104].	46
Figur 3.49: Formoptimaliserings resultatene av fagverket [104, 108].	47
Figur 3.50: Topologioptimalisering av fagverk [113].	48
Figur 3.51: Metodikk for multiobjektiv optimalisering [114].	49
Figur 3.52: Stålforbindelsen designet av Boer [5].	52
Figur 3.53: Oversikt over parametere av rektangulære stålplaten [5].	52
Figur 3.54: Arbeidsflyten for parametrisk analyse med Bever og Karamba3D [6].	54
Figur 3.55: Arbeidsflyten for forbindelsesanalyse [6].	54
Figur 3.56: Forskjellig genererte strukturer fra casen [6].	55
Figur 3.57 Metode for å benytte C#-script for definering av lastkombinasjoner [4].	57
Figur 5.1: Valgt case er markert med flere farger.	60
Figur 5.2: Gjenbruksstasjonen av Avfall Sør-Husholdning.	60
Figur 5.3: Geometri av limtrefagverket.	61
Figur 5.4: Egenvekt saltak.	62
Figur 5.5: Snølast.	62
Figur 5.6: Skjev snølast.	62
Figur 5.7: Sentrert snølast.	62
Figur 5.8: Vindlast oppløft.	63
Figur 5.9: Vindlast jevnt trykk.	63
Figur 5.10: Skjev Vindlast.	63
Figur 5.11: Vindlast på tvers.	63
Figur 5.12: Oppleggs-krefter for hver lastkombinasjon fra Robot.	64
Figur 5.13: Normalkraft diagram.	65
Figur 5.14: Skjærkraft diagram z-retning.	65
Figur 5.15: Moment diagram y-retning.	65
Figur 5.16: Skjærkraft diagram y-retning.	65
Figur 5.17: Momentdiagram z-retning.	66
Figur 6.1: Bruk av Telepathy i Grasshopper.	74
Figur 6.2 Oversiktsbilde av Grasshopper modell.	75
Figur 6.3 Oversiktsbilde av gruppene fra Grasshopper-modellen som etablerer pinnemodell.	76
Figur 6.4 Etablering av pinnemodell del 1 – Hovedpunkter.	77
Figur 6.5 Etablering av pinnemodell del 2 – Hovedlinjer.	77
Figur 6.6 Etablering av pinnemodell del 3 – Linjeinndeling i over- og undergurt.	79
Figur 6.7 Etablering av pinnemodell del 4 – Staver i mønster.	80
Figur 6.8 Etablering av pinnemodell del 5 – Sortering og navngivning av staver.	81
Figur 6.9 Styring av parametere.	81
Figur 6.10 Fagverkets linjer omgjøres til bjelke-elementer.	82

Figur 6.11 Alle identiteter og inndeling av identiteter	83
Figur 6.12 Oversiktsbilde for definering av material- og tverrsnittsegenskaper.....	84
Figur 6.13 Definering av materialtype ved bruk av Beaver-komponenter.....	85
Figur 6.14 Innhentning av materialegenskaper for Eurokode-beregninger	87
Figur 6.15 Definering av tverrsnitts-dimensjoner	88
Figur 6.16 Oversiktsbilde av gruppen fra Grasshopper-modellen som etablerer opplegg- og forbindelser.....	89
Figur 6.17 Oversiktsbilde av gruppene fra Grasshopper-modellen som etablerer laster- og lastkombinasjoner.....	90
Figur 6.18 C#-komponenten for lastkombinasjoner i Grasshopper-systemet.	93
Figur 6.19 C#-komponenten for lastkombinasjoner fra innsiden	94
Figur 6.20 Cluster for Egenvekt (blå).....	95
Figur 6.21 Cluster for Egenlast tak (oransje).....	95
Figur 6.22 Cluster for snølaster (lilla)	96
Figur 6.23 Clustere for vindlaster.....	96
Figur 6.24 Presentasjon av alle laster- og lastkombinasjoner.....	97
Figur 6.25 Oversiktsbilde av gruppen fra Grasshopper-modellen som utfører- og visualiserer resultatene av Karamba-analyse.....	98
Figur 6.26 Overføring fra Karamba-elementer til Beaver-elementer	99
Figur 6.27 Overføring av «2.Overgurt» til Beaver-element	100
Figur 6.28 Utnyttelseskontroll med Beaver-komponenter	100
Figur 6.29 Resultat av utnyttelseskontroll med Beaver	101
Figur 6.30 Dataoverføring inn- og ut av Excel for kontroll av bruksgrensetilstand.....	101
Figur 6.31 Systemet for utføring av kontroll i bruksgrensetilstand basert på visuell programmering	102
Figur 6.32: Visuelle programmeringen for utnyttelsesanalyse av overgruten.....	103
Figur 6.33: Kontroll undergurt	104
Figur 6.34: Visuelle programmeringen for utnyttelsesanalyse av stavene	104
Figur 6.35: Cluster Output data.....	105
Figur 6.36: Oppbyggingen av Komponent for Belastningskrefter	105
Figur 6.37: Beregning av belastningskrefter i Cluster-komponent.....	106
Figur 6.38: Oppbygging av kontrollanalyse komponent	107
Figur 6.39: Oversikt over beregningskontrollen etter EK5 for elementer.....	108
Figur 6.40: Input data for kontrollanalysen	108
Figur 6.41: Beregning motstandsmoment	109
Figur 6.42: Utnyttelse av bøyemoment sterk og svak akse.....	109
Figur 6.43: Utnyttelseskontroll Skjær (Vz)	109
Figur 6.44: Utnyttelseskontroll av aksial trykk	110
Figur 6.45: Utnyttelseskontroll av aksial strekk	110
Figur 6.46: Kontroll kombinert bøyning og aksial strekk.....	110
Figur 6.47: Kontroll kombinert bøyning og aksial trykk	111
Figur 6.48: Knekkingskontroll både sterk og svak akse	111
Figur 6.49: Output verdier til utnyttelseskomponenten	112
Figur 6.50 Forsøk på beregning og modellering av knutepunkt med Beaver	113
Figur 6.51: Visuell programmering av kontrollanalyse av forbindelse.....	114
Figur 6.52: Valgte verdier for dybel diameter, tykkelse stålplate og materialkvaliteter.....	115

Figur 6.53: Valg av partialfaktor, trevirke, antall stålplater og innerfelt i forbindelsen	115
Figur 6.54 Generering av krefter for dybelgrupper i staver og toppunkt (blå cluster)	116
Figur 6.55 Antall elementer i en node.....	117
Figur 6.56 GhPython skript for å addere eller subtrahere skjær- og aksialkrefter.....	117
Figur 6.57 Generering av krefter for dybelgrupper i kontinuerlig over- og undergurt (brun cluster).118	
Figur 6.58 Generering av krefter for noden i krysningspunktet mellom over- og undergurt (lilla cluster)	118
Figur 6.59: Valg av materialfaktorer	119
Figur 6.60: Valg av antall og avstander mellom dyblene	119
Figur 6.61: Valg av ende- og kant avstand fra dybel til tredel og stålplate	119
Figur 6.62: Cluster for beregning av resultatkraften til dybelgruppen og vinkel mellom fiberretningen	120
Figur 6.63: Visuelt programmerte beregningen av vinkel og resultantkraft.....	120
Figur 6.64: Beregning av minimumsavstander og utnyttelse.....	120
Figur 6.65: Visuelt programmering for beregning av minimumsavstander.....	121
Figur 6.66: Utnyttelse mellom valgte lengder og minimumsavstander	121
Figur 6.67: Valg av kneklengde for stålplate i stavene	122
Figur 6.68: Komponent for alle kontroller av dybelforbindelse	122
Figur 6.69: Cluster og beregning av effektivt antall forbindere	123
Figur 6.70: Beregning av hullkantfasthet og flytemomentet	123
Figur 6.71: Illustrasjon for beregning av FM,max.....	124
Figur 6.72: Komponent for beregning av resultantkraft på mest påkjente dybel.....	125
Figur 6.73: Beregning av slissens dimensjon og effektiv tykkelse ytterfelt.....	126
Figur 6.74: Visuelt programmert beregning av bruddfrom c, d og e	126
Figur 6.75: Visuelt programmert beregning av bruddfrom l og m	127
Figur 6.76: Visuelt programmert beregning av bruddfrom f, g og h	127
Figur 6.77: Utnyttelse/kontroll av dybelforbindelsen	128
Figur 6.78: Kontroll splitting.....	129
Figur 6.79: Kontroll skjærblokkbrudd.....	130
Figur 6.80: Cluster for kontroll av stål	131
Figur 6.81: Beregning av hullkantrykk stålplate	131
Figur 6.82: Redusert stålplatehøyde	132
Figur 6.83: Skjærbrudd stålplate	132
Figur 6.84: Trykk-, og strekkbrudd i stålplaten.....	133
Figur 6.85: Eksempel på kneklengde i stålplate	134
Figur 6.86 Undersøkelse om knekking i platen må kontrolleres.....	134
Figur 6.87: Python skript for valg av bøyekapasitet i stålplaten	134
Figur 6.88 Bøyekapasitet i stålplaten	135
Figur 6.89 Kontroll kombinasjon av spenninger i stålplaten	136
Figur 6.90 Knekkingskontroll av stålplate	136
Figur 6.91 Cluster for forbindelse i kontinuerlig overgurt tilknyttet 2 staver (oransje)	137
Figur 6.92 Forbindelser i kontinuerlig overgurt tilknyttet 2 staver (Oransje cluster)	138
Figur 6.93 Nummerert fremgangsmåte for forbindelse i kontinuerlig overgurt tilknyttet 2 staver (Oransje cluster).....	138
Figur 6.94 Beregning av avstander for stålplaten høyde og bredde (ramme 2).....	140

Figur 6.95 Definerer av stavens endepunkt (ramme 3).....	141
Figur 6.96 Linjer som definerer stålplatas ytterkant parallelt med elementets fiberretning (ramme 4)	142
Figur 6.97 Linjer som definerer stålplatas ytterkant vinkelrett på elementets fiberretning (ramme 4)	143
Figur 6.98 Samling av stålplatas hjørner- og vendepunkter (ramme 5).....	144
Figur 6.99 Etablering av stålplater (ramme 6).....	144
Figur 6.100 Definerer av forbindelsens midtpunkt (ramme 7).....	145
Figur 6.101 Etablering av overgurtens dybler i rutenett-struktur (ramme 8).....	146
Figur 6.102 Etablering av stavenes dybler i rutenett-struktur (ramme 8).....	147
Figur 6.103 Måling av avstander for beregner r_i squared og r_{max} (ramme 9).....	147
Figur 6.104 Sortering av punktgruppe (ramme 9).....	148
Figur 6.105 Måler avstander mellom dybelgruppens senterlinje og punktrekkene på tvers av fiberretning (ramme 9).....	149
Figur 6.106 Måler avstander mellom dybelgruppens midtpunkt og punktrekkene i fiberretning (ramme 9).....	150
Figur 6.107 Beregning av « r_{max} » (ramme 9).....	150
Figur 6.108 Måling av knekk lengde (ramme 10).....	151
Figur 6.109 Måling av momentarm (ramme 11).....	151
Figur 6.110 Cluster for skjøttet forbindelse (grønn).....	151
Figur 6.111 Toppunktforbindelse (Grønn cluster).....	152
Figur 6.112 Nummerert fremgangsmåte for toppunktforbindelse (Grønn cluster).....	152
Figur 6.113 Avrundet topp på stålplate.....	153
Figur 6.114 Cluster for forbindelse i kontinuerlig overgurt tilknyttet 1 stav (lilla).....	154
Figur 6.115 Forbindelse i kontinuerlig overgurt tilknyttet vertikal stav (Lilla cluster).....	154
Figur 6.116 Nummerert fremgangsmåte for forbindelse i kontinuerlig overgurt tilknyttet vertikal stav (Lilla cluster).....	155
Figur 6.117 Beregning av avstanden som dybelgruppen i overgurt skal flyttes nedover i tverrsnittet	155
Figur 6.118 Forflytning av dybelgruppen i overgurt nedover i tverrsnittet.....	156
Figur 6.119 Dybelgruppens midtpunkt (lilla cluster).....	156
Figur 6.120 Momentarm i vertikal stav.....	157
Figur 6.121 Cluster for forbindelse i kontinuerlig undergurt (blå).....	157
Figur 6.122 Forbindelser i kontinuerlig undergurt (Blå cluster).....	157
Figur 6.123 Nummerert fremgangsmåte for forbindelser i kontinuerlig undergurt (Blå cluster).....	158
Figur 6.124 System for stavens startpunkt (hvor endeavstanden måles fra) i blå cluster.....	159
Figur 6.125 Cluster for forbindelse mellom over- og undergurt (brun).....	159
Figur 6.126 Forbindelse mellom over- og undergurt (Brun cluster).....	160
Figur 6.127 Nummerert fremgangsmåte for forbindelse mellom over- og undergurt (Brun cluster).....	160
Figur 6.128 Linjer som definerer stålplatas ytterkant parallelt med elementets fiberretning (brun cluster).....	161
Figur 6.129 Etablering av avgrensede linjer vinkelrett på fiberretning (brun cluster).....	161
Figur 6.130 Linjer for hver av dybelrekkene langs med fiberretning (brun cluster).....	162
Figur 6.131 Gruppe som danner punktene i overgurtens dybelgruppe (brun cluster).....	163
Figur 6.132 Linjer som utgangspunkt for forskjellige linjer (Brun.....	163

Figur 6.133 Etablering av forskjøvne linjer langs med overgurtens fiberretning (Brun cluster)	164
Figur 6.134 Etablering av punkter på linjene i fiberretning (Brun cluster).....	164
Figur 6.135 Ytre linjer som går parallelt med dyblene på tvers i overgurtene (Brun cluster).....	165
Figur 6.136 Etablering av dybler (Brun cluster).....	165
Figur 6.137 Måling av avstander for beregner r_i squared og r_{max} (Brun cluster).....	166
Figur 6.138: Opossum valg av optimalisering og resultater	167
Figur 6.139: Korrigert høyde og bredde parametere for tverrsnitt	168
Figur 6.140: størrelsesoptimalisering av overgurtene	169
Figur 6.141: Størrelsesoptimalisering av undergurt.....	169
Figur 6.142: Størrelsesoptimalisering av vertikale staver	170
Figur 6.143: Størrelsesoptimalisering av diagonal 0	170
Figur 6.144: Størrelsesoptimalisering av diagonal 1	171
Figur 6.145: Størrelsesoptimalisering av diagonal 2	171
Figur 6.146: Størrelsesoptimalisering av diagonal 3	172
Figur 6.147: Størrelsesoptimalisering av diagonal 4	172
Figur 7.1: Variablene for parametrisk design av fagverket	173
Figur 7.2: Oppdatering av kontrollanalyse ved endring av parametere	174
Figur 7.3: Oppdatering av stålplate og dybler når antallet av staver endres	174
Figur 7.4: Forbindelse i vertikal stav	175
Figur 7.5: Oppdatering av dybeler og stålplater, ved endring av tverrsnitt høyde	175
Figur 7.6: Endring av tverrsnitts bredde.....	175
Figur 7.7: Endring av avstand mellom stålplater	176
Figur 7.8: Endring av antall dybler	176
Figur 7.9: Oppdatering av kontrollen for knutepunktet i Overgurt	177
Figur 7.10: Oppdatering av kontrollen for knutepunktet i Overgurt	177
Figur 7.11: Oppleggs-krefter for hver lastkombinasjon	178
Figur 7.12: Element navn fra Robot og Grasshopper	179
Figur 7.13: Moment- diagram (M_y) fra Karamba3D	186
Figur 7.14: Skjærkraftdiagram (V_z) fra Karamba3D	187
Figur 7.15: Normalkraftdiagram (N_x) fra Karamba3D	187
Figur 7.16: Limtrefagverk med oversikt over elementer	188
Figur 7.17: Illustrasjon av stålplater og dybler programmert i Grasshopper	196
Figur 7.18: Visualisering av forbindelse i toppunkt fra Rhino3D.....	197
Figur 7.19: Forbindelse overgurt, diagonal 1 og 2 visualisert i Rhino3D.....	198
Figur 7.20: Forbindelse fra Grasshopper og Sweco.....	202
Figur 7.21: Sweco overgurt forbindelse	202
Figur 7.22: Knutepunkt i diagonal 4, koblet til undergurt	205
Figur 7.23: Knutepunkter i diagonal 4.....	206
Figur 7.24: Forbindelse overgurt og vertikal stav.....	206
Figur 7.25: Forbindelse undergurt, diagonal 0 og 1	208
Figur 7.26: Forbindelse som kobler sammen undergurt, diagonal 2 og 3.....	209
Figur 7.27: Forbindelsen som sammenkobler undergrut, diagonal 4 og vertikal stav	210
Figur 7.28: Forbindelse som sammenkobler undergrut og overgurt	211
Figur 7.29: Swecos prosjekterte forbindelse med over- og undergurt	212
Figur 7.30: Størrelsesoptimalisering fra Opossum.....	217

Figur 7.31: Korrigering av Opossum resultat.....	218
Figur 8.1: Sammenligning av skjærkraft, x-akse lastkombinasjoner, y-akse Kilonewton.....	220
Figur 8.2: Sammenligning av momentkraft y-retning, x-akse lastkombinasjoner, y-akse Kilonewton-meter.....	221
Figur 8.3: Sammenligning av momentkraft z-retning, x-akse lastkombinasjoner, y-akse Kilonewton-meter.....	221
Figur 8.4: Sammenligning av trykkraft, x-akse lastkombinasjoner, y-akse Kilonewton.....	222
Figur 8.5: Sammenligning av strekkraft, x-akse lastkombinasjoner, y-akse Kilonewton.....	223
Figur 8.6: Diagram for sammenligning av opplagerkrefter.....	224
Figur 8.7: Utnyttelse under- og overgurt vist i prosent %.....	228
Figur 8.8: Utnyttelse vertikal og diagonalstaver 0 og 1 vist i prosent %.....	230
Figur 8.9: Utnyttelse diagonal 2, 3 og 4 vist i prosent %.....	231
Figur 8.10: Sammenligning kontroller i overgurt.....	234
Figur 8.11: Sammenligning kontroll i undergurt.....	235
Figur 8.12: Sammenligning kontroll staver.....	236
Figur 8.13: De Kritiske utnyttelsene fra stålplate kontrollene i knutepunkter.....	237
Figur 8.14 Materialinnhenting ved bruk av Beaver versus materialinnhenting ved bruk av Karambas bibliotek.....	239
Figur 8.15 Egenskaper parallelt med fiberretning og vinkelrett på fiberretning (fra Karambas bibliotek).....	239
Figur 8.16 Problematikk knyttet til hvordan tredelene møtes.....	242
Figur 8.17 Fagverkets skjøter.....	243
Figur 8.18 Problematikk i dybelgruppe (forbindelse mellom over- og undergurt).....	243
Figur 8.19 Problematikk i dybelgruppen i 4. diagonal.....	244

Tabelliste

Tabell 3.1 Lastvarighetsklasser i henhold til Eurokode 5 [58]	28
Tabell 3.2 Verdier av k_{mod} (gjelder limtre, konstruksjonsvirke og LVL) [58]	28
Tabell 3.3 Dimensjonerende fasthetsverdier relevant for fagverkskontroller	29
Tabell 6.1 overføring av fasthet- og stivhetsdata fra <i>Beaver</i> til <i>Karamba</i>	86
Tabell 6.2 Oversikt over lastkombinasjoner	91
Tabell 6.3 Den totale ligningen for hver av lastkombinasjonene	91
Tabell 6.4 Hvordan scriptet deler opp ligningene for lastkombinasjonene og etablerer lister for hver enkelt last-type	92
Tabell 6.5 Beskrivelse av rekkefølgen på operasjonene som utføres for modellering av forbindelse	139
Tabell 6.6 Programmering som er særegen for modellering av toppunkt (grønn cluster)	153
Tabell 6.7 Programmering som er særegen for modellering av forbindelse i kontinuerlig overgurt tilknyttet vertikal stav	155
Tabell 6.8 Programmering som er særegen for modellering av forbindelsene i kontinuerlig undergurt	158
Tabell 7.1: Sammenligning av krefter overgurt	180
Tabell 7.2: Sammenligning av krefter undergrut	181
Tabell 7.3: Sammenligning av krefter vertikale staver	182
Tabell 7.4: Sammenligning av krefter Diagonal 0	183
Tabell 7.5: Sammenligning av krefter Diagonal 1	184
Tabell 7.6: Sammenligning av krefter Diagonal 2	184
Tabell 7.7: Sammenligning av krefter Diagonal 3	185
Tabell 7.8: Sammenligning av krefter Diagonal 4	186
Tabell 7.9: Geometri overgrut	188
Tabell 7.10: Kontroll overgurt	189
Tabell 7.11: Geometri undergrut	189
Tabell 7.12: Kontroll undergurt	190
Tabell 7.13: Geometri vertikale staver	190
Tabell 7.14: Kontroll vertikale staver	191
Tabell 7.15: Geometri diagonal 0	191
Tabell 7.16: Kontroll diagonal 0	192
Tabell 7.17: Geometri diagonal 1	192
Tabell 7.18: Kontroll diagonal 1	193
Tabell 7.19: Geometri diagonal 2	193
Tabell 7.20: Kontroll diagonal 2	194
Tabell 7.21: Geometri diagonal 3	194
Tabell 7.22: Kontroll diagonal 3	195
Tabell 7.23: Geometri diagonal 4	195
Tabell 7.24: Kontroll diagonal 4	196
Tabell 7.25: Kontroll overgurt i toppunkt	197
Tabell 7.26: Kontroll knutepunkter i diagonal 0	198
Tabell 7.27: Kontroll overgurt i forbindelsen med diagonal 1 og 2	199
Tabell 7.28: Kontroll diagonal 1	200
Tabell 7.29: Kontroll diagonal 2	201

Tabell 7.30: Kontroll overgurt forbindelse med diagonal 3 og 4.....	203
Tabell 7.31: Kontroll knutepunkter i diagonal 3.....	204
Tabell 7.32: Kontroll knutepunkt diagonal 4.....	205
Tabell 7.33: Kontroll overgurt med vertikal stav.....	207
Tabell 7.34: Kontrollknutepunktene i de vertikale stavene	208
Tabell 7.35: Kontroll undergurt med diagonal 0 og 1	209
Tabell 7.36: Kontroll undergurt med diagonal 2 og 3	210
Tabell 7.37: Kontroll undergrut i forbindelse med diagonal 4 og vertikal stav	211
Tabell 7.38: Kontroll knutepunkt undergrut i forbindelse med overgurt	212
Tabell 7.39: Kontroll knutepunkt overgrut i forbindelse med undergrut	213
Tabell 7.40: Kontroll stålplate toppunkt	214
Tabell 7.41: Kontroll stålplate overgurt, diagonal 1 og 2	214
Tabell 7.42: Kontroll stålplate overgurt, diagonal 3 og 4	215
Tabell 7.43: Kontroll stålplate overgurt og vertikal stav	215
Tabell 7.44: Kontroll stålplate undergurt med diagonal 0 og 1.....	216
Tabell 7.45: Kontroll stålplate undergurt med diagonal 2 og 3.....	216
Tabell 7.46: Kontroll stålplate undergurt med diagonal 4 og vertikal stav	216
Tabell 7.47: Kontroll stålplate under- og overgurt.....	217

1 Innledning

Den tradisjonelle dimensjoneringsprosessen for trekonstruksjoner er en tredelt prosess for oppdragsgiver (Sweco). Først beregnes kreftene i digitale beregningsmodeller som Robot eller FEM design. Deretter hentes opptredende krefter og andre nødvendige parametere manuelt inn i beregningsprogrammer som Excel, Mathcad eller lignende. I disse beregningsprogrammene utføres dimensjonering av elementer og forbindelser i henhold til Eurokode. Det forsøkes å finne en optimal løsning. Til slutt må resultatet overføres til et modelleringsprogram for å skissere hvordan løsningene skal utformes. Denne prosessen gir stor sannsynlighet for feiltakelser i overgangene mellom programvarene, og kan effektiviseres [1].

I tillegg er prosessen svært sårbar for endringer som oppstår underveis, da disse må korrigeres for i alle programvarene. Det er heller ikke tilrettelagt for rask korrigerende innad i de tradisjonelle BIM-programvarene, og endringer vil i mange tilfeller kreve omfattende modifikasjoner på de omkringliggende komponentene av modellen [2]. Dette er fordi BIM-modeller er sammensatt av uavhengige objekter [3]. I dag finnes det imidlertid programvarer basert på visuelle programmeringsspråk, som tilrettelegger for å bygge ett helhetlig, integrert system for hele dimensjoneringsprosessen [4-6]. Metoden for å lage slike systemer, kalles parametrisk design. Her bygges objektene til å være avhengig av hverandre, basert på algoritmer og formler [7].

I tillegg kan systemet bygges for å kunne optimalisere konstruksjonen. Da vil programvaren endre på nøkkelparametere som styrer objektene, og flere alternativer vil bli testet og vurdert. Prosessen er relativt rask, men tidsbruken avhenger av hvor mange parametere som endres og hvor mange mål man ønsker å oppnå. Til slutt presenteres ett sett med passende løsninger, som kan diskuteres med andre kollegaer for å finne den mest optimale løsningen [8].

Forskningen som er gjennomført på området per i dag, har hovedvekt på stål-konstruksjoner. Det observeres imidlertid en økning av trekonstruksjoner i bransjen, og derfor bør bruken av tremateriale i parametrisk design undersøkes ytterligere [9]. Oppgavens generelle problemområde omhandler derfor bruk av programvarer som baseres på visuell programmering, til bruk for effektivisering av tradisjonelle dimensjoneringsprosesser for trekonstruksjoner.

2 Samfunnsperspektiv

Bygg og anlegg er en av de største globale næringene og utgjør 10% av det globale bruttonasjonalproduktet (BNP) [10]. I følge Statista Research Department [11] hadde byggebransjen en årlig inntekt på 7,3 billioner dollar USD i 2021, og det er anslått at denne inntekten vil stige til 14,4 billioner dollar USD i 2030. Bygg og anlegg utgjør en vesentlig del av verdensøkonomien, og skaper verdi gjennom bygninger og infrastrukturer [12]. Det er estimert at folketallet kommer til å stige til 10 milliarder mennesker, og Bertollini [13] påpeker at det må bygges trettentusen bygninger hver dag fram til 2050 for å støtte denne folkeveksten. Bygg og anlegg har derfor et stort «Corporate Social Responsibility» (CRS) [14], som defineres som næringslivets kontinuerlige ansvar til å oppføre seg etisk og bidra til økonomisk utvikling, samtidig som livskvaliteten til mennesker, lokal- og verdenssamfunnet forbedres. CRS innen bygg og anlegg har i de siste årene blitt en økende agenda, hovedsakelig på grunn av to paradoksale grunner [15].

For det første er byggebransjen sosialt ansvarlig for etablering av et bredt spekter av nødvendige bygninger og infrastrukturer, som styrker utdanning, økonomi, helse, kulturell identitet, og sosiale aspekter av menneskeheten [15]. Ved å etablere nødvendige fasiliteter er byggebransjen direkte med på å påvirke og forme den sosiale livstilen til mennesket. Dette utføres eksempelvis ved å oppbygge leilighetskomplekser i nærheten av skoler, handelsmarkeder, helsetjenester, kollektivtransport og friluftsmuligheter [16]. Bygg og anlegg er i tillegg en viktig kilde for sysselsetting gjennom jobbmulighetene som tilbys til både utdannede, og ikke utdannede arbeidere [15]. I USA er bygg og anlegg den sjettede største sysselsetningsindustrien, og fra 2016 til 2022 økte sysselsettingen med 1,7 millioner [15, 17].

For det andre oppstår det derimot store utfordringer fra etableringen av nødvendige bygninger og infrastruktur. Byggebransjen har et overdrevent energi- og ressursforbruk som kommer fra byggeaktiviteter som utvinning, prosessering og transport av råvarer, design, oppbygging og nedrivning av bygninger og infrastruktur [15]. På grunn av disse aktivitetene bruker bygg og anlegg omtrent 50% av den globale råvaremengden, og 36% av den globale energimengden [18]. I tillegg viser statistikker at bygg og anlegg innen global forurensing, står for 50% av klimagassutslippet, 23% av luftforurensingen, 50% av deponiavfallet og 40% av drikkevannsforurensingen [19]. Xia m.fl. påpeker at dette gir negative påvirkninger på det fysiske miljøet og økosystemet i form av klimagassutslipp, luftforurensing, avfall, og støv. Det presiseres også at byggeaktiviteter generelt inneholder intensivt arbeid med høy eksponering for ulykker, noe som medfører at bygningsarbeidsforholdene er utrygge og fysiske skadelige for arbeiderne [15]. I Kina ble det registrert 734 konstruksjonsulykker og 840 dødsfall i 2018. I 2019 oppstod det over 5000 bygningsarbeidsulykker i USA, noe som er mer enn 100 i uken [10]. Det er også en ekstrem konkurranse og fokus på fortjeneste i byggebransjen, noe som resulterer i at mange byggeorganisasjoner eksponerer arbeidere i usikre og helsefarlige forhold, og i enkelte utviklingsland benyttes barnarbeid på visse områder [10, 15].

Utfordringene innenfor dette paradokset har økt forskningsinteressen av CSR i bygg og anlegg [15]. Xia m.fl. nevner at den økende forpliktelsen til å redusere de negative påvirkningene knyttet til forretningsdriften i byggebransjen kan for øvrig påvirke både miljøet og samfunnet. I denne forbindelsen bidrar byggebransjen både direkte eller indirekte bidra til å oppnå De Forente Nasjoners

(FN) bærekraftsmål. Eksempelvis trenger flere av bærekraftmålene for utdanning, helse, mat, energi og vann pålitelige produkter fra bygg og anlegg for å oppnå dem [20]. Flere byggeorganisasjoner har dermed forstått at de er en del av et større sosialt system, og engasjerer seg i de aktivitetene som skal forbedre systemet og virksomheten [15]. Siden det er anslått at flere mennesker kommer til å bosette seg i byer, er det viktig at byggeorganisasjoner er med på å sikre en framgang og fremtidig velvære for menneskene og samfunnene [18]. I de neste delkapitlene presenteres hvordan byggebransjen arbeider direkte er med på å oppnå enkelte av FNs bærekraftsmål.

2.1 Tre som byggemateriale

Studier indikerer en økning for bruk av tre i byggebransjen. I dag anerkjennes tre som et alternativt til de tradisjonene byggematerialene betong og stål for oppbygging av høyhus [9]. En av hovedårsakene er den reduserte miljøpåvirkningen i forhold til betong og stål. I dag står betongproduksjonen for 8% og stålindustrien for 7% av verdens totale CO₂ – utslipp [9, 21]. Tre er det eneste fornybare byggematerialet, og det har en stor tilgjengelighet over hele verden. I tillegg fanger trær opp CO₂ fra luften, som gjør trevirke til et karbon-bindende byggemateriale [9, 22]. Karbondioksid frigjøres til atmosfæren igjen enten ved en forbrenningsprosess av trevirke eller ved biologisk nedbryting [23]. Videre, har treindustrien en begrenset mengde med miljøfiendtlige utslipp, ettersom produksjonen har et lavt forbruk av fossil energi og høy andel av klimanøytral bioenergi [24]. Dermed vil bruken av tre som byggemateriale gi positive miljøgevinster, dersom det benyttes bærekraftig forvaltet skog [9].

FNs bærekraftsmål nr. 13 [25] omhandler å stoppe klimaendringene, et av tiltakene for å oppnå målet er å kutte, fange og lagre CO₂- utslippet. Bruk av tre som byggematerialer vil derfor være positivt med tanke på å bekjempe klimaendringene. Den Østeriske tømmer bedriften Binderholz [26] presiserer at dersom 10% av alle hus i Europa hadde vært bygget av tre, ville dette redusert det årlige CO₂ utslippet med 1,8 millioner tonn, noe som er 2% av det totale karbondioksidutslippet. Det Norske Treteknisk institutt [27] mener at for å oppnå både de internasjonale og nasjonale miljømålene må det bygges i tre, og presiserer at det kreves innovasjon i byggemetoder, beskrivelser, produkter, dokumentasjon og tilpassinger i produksjonen for å oppnå dette.

Delmål nr. 15.2 [28] handler om å «fremme innføringen av en bærekraftig forvaltning av all skog, og stanse avskogingen, gjenopprette forringede skoger, og i betydelig grad øke gjenreisningen og nyplantning av skog på globalt nivå». Kolbein Bell [22] forfatteren av «Dimensjonering av trekonstruksjoner» presiserer at dersom avkastningen av skogen er lik tilveksten, vil skogen være miljønøytral og skogsarealet regnes som uendret. Det nevnes også at dersom skogsmassen holdes konstant og hogstmoden skog benyttes til produksjon av bygningsmateriale, vil dette øke den totale karbonlagringen. Å lagre karbon i bygninger er miljømessig bedre enn at skogen råtner bort [22].

2.2 Digitalisering

Delmål 8.2 av FNs bærekraftsmål omhandler økning av økonomisk produktivitet med utgangspunkt i innovasjon og teknologisk fremgang [29]. Generelt har BIM og andre digitale verktøy stor innvirkning på bygge- og anleggsbransjen i form av forbedret produktivitet, sikkerhet og bærekraftighet [30]. Dessuten vil bruk av visuell programmering for utførelse av parametriske design kunne forbedre produktiviteten i prosjekteringsfasen enda et steg videre. Dette fordi revidering av enkeltfaktorer i

3D-modellen ved tradisjonell utførelse, i noen tilfeller krever omfattende modifikasjoner på de omkringliggende komponentene av modellen. Særlig problematisk er det når modellene er geometrisk komplekse. Programvarer som utfører parametrisert design (f.eks. Grasshopper) tillater derimot å spesifisere forhold mellom de ulike parameterne i modellen, og kan muligens løse denne problemstillingen. Teknologien gir rom for å optimalisere hver enkelt komponent på en tidseffektiv måte, og bidrar dermed til å øke økonomisk produktivitet i byggeprosjekter [2]. Bruk av visuell programmering kan i tillegg være økonomisk fordelaktig ved at det ikke krever ansettelse av særegen profesjonell kompetanse innen datateknologi. Det visuelle programmeringsspråket har nemlig brukervennlighet som en generell målsetning [31].

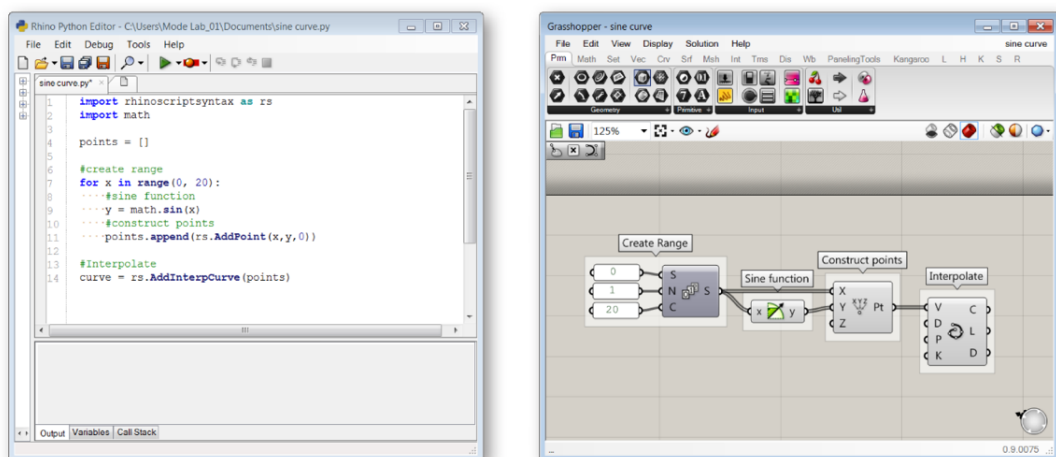
I en studie utført av Hwang m.fl. ble risikofaktorer for miljøvennlige byggeprosjekter i Singapore undersøkt. En av faktorene var at oppdragsgivere avventer med å implementere nye byggemetoder og velger heller velkjente teknikker og metoder [32]. Grunnen til dette, kommer ofte av at det er utfordrende å få forsikring på noe som ikke er godkjent og testet. Flere bedrifter ønsker ikke å ta i bruk ny teknologi, siden de forstår risikoen ved å være den første som tar i bruk teknologien, og vil dermed heller være den neste [33]. Det kan også påpekes at innovasjon av teknologien er stadig i endring, noe som gjør at implementeringen av ny teknologi ofte oppstår tidligere enn hva markedet er klar for [34]. Den største risikofaktoren i studien til Hwang m.fl. var mangelen på kompetanse innen moderne teknologi.

3 Kunnskapsbakgrunn

For å belyse problemområdet presenteres bakgrunnskunnskap om parametrisk design og visuell programmering, samt hvordan programvaren Grasshopper baseres på dette. Videre belyses teoretisk grunnlag for beregning av trekonstruksjoner og fagverk, samt hvordan dette utføres i henhold til Eurokode 5. I tillegg forklares teoretiske premisser bak fagverksforbindelser med dybler og slisseplater, særlig med hensyn til hvorfor beregningene utføres. Deretter presenteres relevant teori som er avgjørende for å forstå hvordan fagverk kan optimaliseres. Avslutningsvis presenteres relevante fagartikler og oppgaver som har utført tilsvarende forskning/arbeid.

3.1 Visuell programmering

Visuell programmering er en type visuelt programmeringsspråk, der brukeren oppretter programmer ved å manipulere logiske elementer grafisk, istedenfor å bruke tekstprogrammering [35, 36]. Det som kjennetegner tekstprogrammering, er at brukeren skriver koder som er bundet av språkspesifikk syntaks. Visuell programmering kobler derimot funksjonelle blokker eller komponenter til en sekvens med handlinger. Den eneste syntaksen som kreves er at inputene til blokkene mottar passende datatype [35]. I Figur 3.1 illustreres både tekstprogrammering og visuell programmering for tegning av en sinuskurve. Tekstprogrammeringen er utført i et Python skript, mens den visuelle programmeringen er utført i Grasshopper.



Figur 3.1: Illustrerer både tekstprogrammering og visuell programmering [35]

I løpet av 1960 til 1970-tallet begynte folk å opprette grafiske representasjoner av logikk ved å benytte flytskjemasystemer som Pygmalion og Graphic Input Language [37]. Disse systemene blir sett på som et tidlig visuelt programmeringsspråk, som hjalp folk med å visualisere koder og lage programmer uten å bruke tekstbaserte koder. Dette konseptet ble benyttet fram til 1990-tallet når datamaskiner ble oppgradert til anstendige grafiske grensesnitt. Både Apple og Microsoft kom med hver sine visuelle programmeringsverktøy, noe som økte bruken av visuell programmering. Siden den gang har visuell programmering økt betraktelig, og er nå et viktig verktøy innen databehandling [37].

Datamyte [37] som leverer digitale verktøy innen produksjon, presiserer at visuell programmering er mer forståelig og en enklere måte å lære om dataprogrammering i forhold til tekstkodning. Dette kommer av at det er en enklere tilnærming å forstå den underliggende koden og i tillegg er det

enklere å feil søke etter problemer. Programvareutviklere Kissflow [36] nevner at tekstkoding tvinger brukere til å tenke som en datamaskin, mens visuell programmering er en mer forståelig prosess som kan forklares med forståelige ord for mennesker.

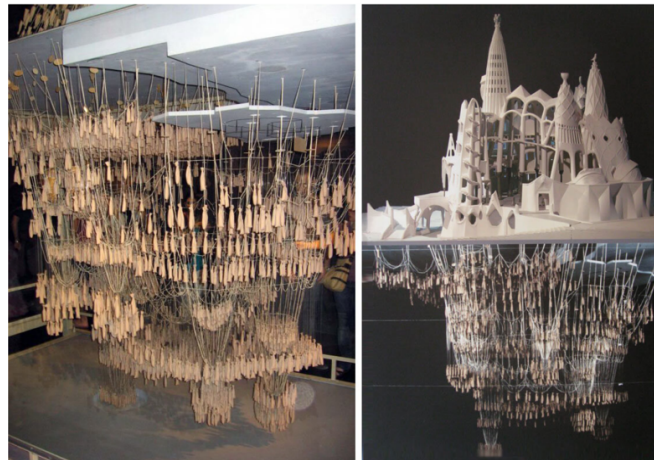
3.1.1 Parametrisk design

Orddelen «para» betyr i tillegg [38], mens «metrisk» beskrives som en avstandsfunksjon [39]. Begrepet parametrisk har opprinnelse innen matematikk, og relateres til et uttrykk i form av en parameter [40, 41]. Oxford Dictionary definerer parameter som en numerisk eller en målbar faktor, som utgjør en serie som definerer et system eller noe som danner betingelser for en prosess [40]. Parametrisk er dermed en type funksjon som har en uavhengig parameter som ofte er betegnet som « t » eller « x » i matematikken [42]. Hvis verdien til parameteren endres, forandres også resultatet til funksjonen.

Parametric Architecture [43] som er en nettbasert plattform innen parametrisk design og digitale verktøy innen blant annet arkitektur, beskriver parametrisk design som en prosess som er basert på algoritmer med uttrykk av parametere og regler som definerer et forhold mellom hensikten og resultatet til designet. Det presiseres også at det er et paradigme innen design, der forbindelsen mellom elementer benyttes til å manipulere og opparbeide komplekse strukturer og geometrier [43].

Historien til parametrisk design

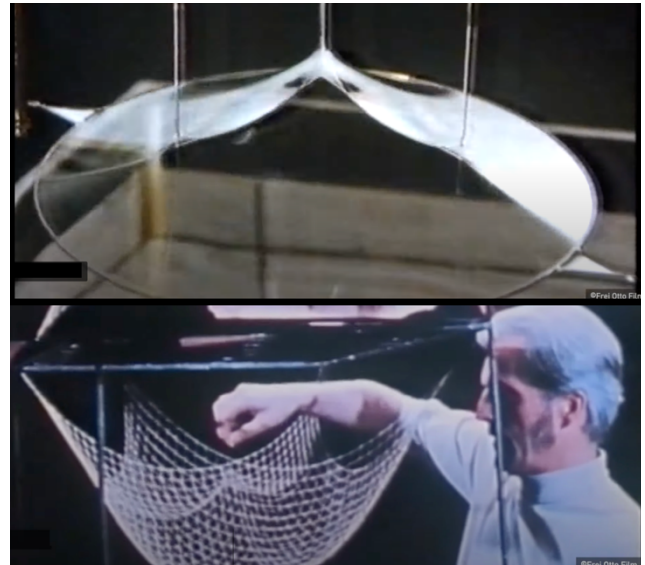
Det første eksempelet innen parametrisk design er den hengende kjedemodellen som Antoni Gaudi brukte til å designe Colònia Güell kapellet og Sagrada Família kirken [41, 43]. Ved å henge opp tråder med vekter, skapte han sammensatte buer med trådene. Dersom Gaudi justerte posisjonen til trådene, kunne han endre formen til buene og dermed endre hele modellen. Daniel Davis [41] som er en forsker innen teknologi og arkitektur, påpeker at den sammensatte formen til trådene som skapes av tyngdekraften,



Figur 3.2: Kjedemodellen av Colònia Güell kapellet [7]

alltid vil stå imot rent trykk på grunn av Hooke's lov. Davis nevner også at den hengende kjedemodellen inneholder alle komponentene i en parametrisk ligning. Vektene, lengdene til trådene og bue plasseringene blir de uavhengige parametere. I tillegg er toppunktene til de forskjellige buene et sett med utfall som kommer fra parametere i eksplisitte funksjoner. Gaudi kunne automatisk utlede formen til buene ut ifra den påvirkede tyngdekraften, istedenfor å manuelt beregne dem ved bruk av parametriske formler. Ved å manipulere de uavhengige parametere, kunne Gaudi generere flere versjoner av strukturen, og samtidig være sikker på at den resulterende strukturen ville stå imot rent trykk [41]. I Figur 3.2 illustreres kjedemodellen som skaper formen til Colònia Güell kapellet ved å snu modellen opp ned.

Gaudi's metode ble videre utviklet av arkitekten Frei Otto, som brukte overflater som kom fra såpefilm og baner som ble funnet gjennom ull som ble dyppet i væske [41]. Såpefilm er illustrert øverst i Figur 3.3, der et tynt lag med væske som er omgitt av luft skaper en overflate. Nederst på figuren skapes baner av fuktig ull. Otto kalte formene som ble designet av disse modellene for «from finding», som viser til utforskningen innenfor parametrisk design. I Gaudi's tilfelle kunne han modifisere parameterne til modellen, som er en betydningsfull komponent i det parametriske modelleringsdogmet for arkitekter. Fordelen med parametrisk modellering kommer av utforskningen med resultatet til modellen [41].



Figur 3.3 Overflaten fra såpefilm og banene fra fuktig ull [44]

Digitalisering av eksperimenteringene fra Gaudi og Otto kom i 1963 [41]. Ivan Sutherland utviklet det første interaktive designprogrammet Sketchpad. Denne programvaren gjorde det mulig å utforske parametriske ligninger, og endre relasjonene til modellen. I 1982 ble AutoCAD utgitt, som erstattet kurver og den selvreplicerende geometrien fra parametriske ligninger, med kommandoer som fikk designeren til å tegne todimensjonale linjer på skjermen. Ikke før atten versjoner senere, ble parametrisk funksjonalitet introdusert i AutoCAD2010 [41].

I dag finnes parametriske modellering i flere BIM-verktøy, og benyttes både i tekstsripping og visuell programmering [43]. Det fleste designer bruker parametrisk modellering innen visuell programmering. Programvarene Grasshopper og Dynamo er visuelt grensesnitt, som er basert på grafer som kartlegger flyten av forbindelser fra parametere som kommer fra brukerdefinerte funksjoner, noe som resulterer i genererte geometrier [43].

3.1.2 Parametrisk design og BIM

Gundnason m.fl. [3] beskriver at den tekniske konverteringen til bygningsinformasjonsmodellering (BIM) er iboende parametrisk. For å opprette et visuelt byggeprosjekt i et BIM-program, vil brukeren manuelt sette inn parametriske objekter som kan justeres og modelleres. Disse objektene utfører bestemte handlinger og sammenstilles med hverandre, ved å bestemme egenskapsverdiene til alle de påvirkede parameterne. Dersom parametere endres i et BIM-programmet vil geometrien genereres i henhold til endringen. Det som er begrensingen med et BIM-verktøy er at brukeren er avgrenset med mengden av hva som kan direkte modelleres i strukturen. Dette kommer av at programvaren har innebygde algoritmer og kunnskap om konstruksjonsdomenet, noe som avgrenser modelleringsmengden [3].

Når det kommer til parametriske modelleringsystemer, lager brukeren en oppskrift for et bestemt prosjekt [3]. Denne oppskriften er utledet av matematiske formler som frembringer en geometrisk

modell fra sekvenser av input data, som består av både geometrisk og numerisk informasjon. Dette gjør at designeren kontrollerer genereringen av objekter fra en logisk prosess [3].

I parametrisk design er hele modellen sammenstilt, noe som gjør at brukeren har full kontroll over overordnede former og de minste detaljene [3]. BIM derimot, er avhengig av parametrisk funksjonalitet, og benyttes på et objektnivå. Dette gjør at BIM-modellen er sammensatt av uavhengige objekter. Gundnason m.fl. påpeker at BIM-verktøy kan være mer brukervennlig, siden det er mye informasjon i programvaren. Gjennom modellering av objektene i BIM, skapes det informasjon om prosjektet, og inne i hvert objektene er det innebygd informasjon som kan være betydningsfulle. Med andre ord lager brukeren bæresystemet til prosjektet, mens BIM-programvaren lager informasjonen om hvert objekt i bæresystemet. I tillegg inneholder BIM all dokumentasjon som brukes igjennom hele prosjektet. Denne dokumentasjonen er også nødvendige for å få byggetillatelse. Dette medfører at brukere ofte importerer den parametriske modellen inn i BIM-programvarer, ettersom parametriske programvarer er orientert mot geometri og har dermed ikke den samme innebygde informasjon som BIM-verktøyene. Parametrisk modellering er derfor mer egent i designfasen av prosjektet, siden det parametriske systemet har en stor fleksibilitet og dyktighet ved testing og vurdering av modellens geometri og struktur. Den parametriske modellen er ofte ferdigstilt før den implementeres til BIM-verktøyene, dette medfører at den genererte geometrien ikke kan forandres [3].

Gundnason m.fl. [3] presiser også at BIM-programmer ikke har mulighet for å benytte programmering, noe som er mulig i parametriske programvarer. Dette er en stor begrensing innen BIM-verktøyene, ettersom programmering gjør det mulig for brukeren å lage enheter som kan utføre en oppgave. Dette gjør at det ikke er noen begrensinger for hva som er mulig å oppnå i en parametrisk modell [3].

3.1.3 Parametrisk modellering og parametrisk design

Edmonds m.fl. [8] presiserer at selv om parametrisk modellering og parametrisk design er koblet sammen innen beregningsdesign, er det viktig å forstå forskjellen mellom dem. Beregningsdesign handler om å bruke beregningsstrategier som skal hjelpe designprosessen. Dette har hatt en stor utvikling de siste årene, siden datamaskiner og det visuelle programmeringsspråket har utvidet og forbedret de tradisjonelle designmetodene. Dette har dermed økt mulighetene for å optimalisere og automatisere designprosessen innen flere felt [8].

Edmonds m.fl. [8] nevner at tradisjonell modellering begynner først med å definere den statiske geometrien og deretter tegne 3D-elementer, som tilslutt sammenkobles for å skape bygningen. Parametrisk modellering skiller seg ut fra den tradisjonelle modellering ved at den kan manipulerer geometriske elementer. Ved å variere parameterne som er tilkoblet bygningskomponentene, endres 3D-geometrien [8].

Parametrisk design er det neste steget i utviklingen av parametrisk modellering, og innebærer at designresultatet endres automatisk basert på manipuleringen av de geometriske elementene [8]. For å gjøre dette må arbeidet automatiseres, noe som er spesielt nyttig når en ingeniør må designe hver kolonne i en statisk modell. Hvis kolonnehøyden er en variabel, vil iterasjoner av denne høyden være en lang prosess for ingeniøren. Ved å bruke parametrisk design kan algoritmer automatisere denne

prosessen og dermed finne den mest optimale størrelsen på kolonnene basert på høyden. Dette gir en reaktiv modell som oppdateres automatisk basert på manipulasjonen av geometrien, som er fordelaktig hvis krav om strukturendringer oppstår [8].

3.1.4 Grasshopper

Grasshopper er utviklet av David Rutten ved Robert McNeel & Associates, og er en grafisk algoritmedigerer som er innkoblet i modelleringsverktøyet Rhino3D [35, 45]. I Grasshopper opprettes regler og instruksjoner som beskriver hvordan Rhino skal automatisere oppgaver og geometri modeller. Det er dermed ingen objekter eller geometri i Grasshopper, men ved hjelp av flere forskjellige grasshopper-komponerer illustreres all geometri i visningsporten til Rhino. Dette medfører at endringer i Grasshopper vil oppdateres direkte i Rhino, ettersom det er en dynamisk sammenkobling mellom dem [35].

Siden Grasshopper bruker et grafisk algoritmisk programmeringsspråk, er det ikke nødvendig for brukeren å ha kunnskap om tekstprogrammering eller koding [46]. De grafiske algoritmene i Grasshopper gir en nøyaktig og fullstendig oversikt over prosedyrene, og muligheten til å presist definere og parametrisk kontrollere modellen. Plattformen utvikler også programlogistikk innenfor et induktivt grafisk grensesnitt [35]. Prosedyrene fremstilles ved å dra komponenter til lerretet som er hoved redigeringsvinduet i Grasshopper. Outputene til disse komponentene tilkobles inputene til påfølgende komponenter, noe som skaper sekvenser med prosederer som leses fra venstre til høyere [35].

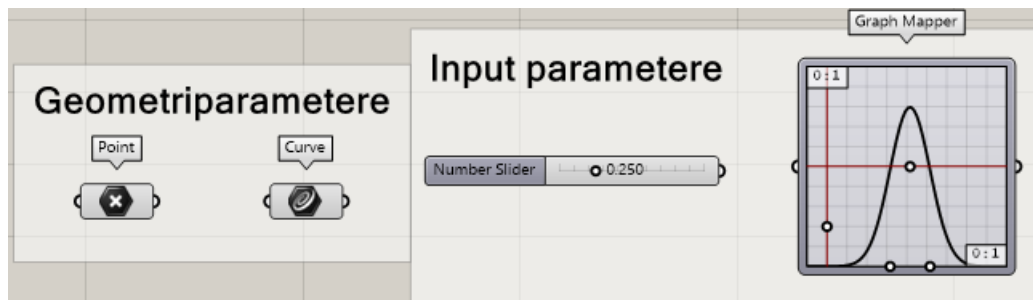
Opphavet til Grasshopper kommer fra funksjonaliteten til «Record History» knappen i Rhino 4 [35]. Denne funksjonen gjorde det mulig for brukere å lagre implisitte modelleringsprosedyrer i samspill med modelleringen. David Rutten ville ha mer eksplisitt kontroll over den lagrede modelleringsprosedyren, og utviklet dermed «Expisitt History», som er forløperen til Grasshopper. Dette medførte at brukere kunne detaljert redigere lagringen og utvikle logiske sekvenser av de innebygde funksjonene i Rhino 4. I 2014 ble Grasshopper et kraftig robust visuelt programmeringsverktøy i Rhino, og muliggjorde oppsette av generative algoritmer [35, 47]. På grunn dette er Grasshopper mye bruk for parametrisk modelleringen innen konstruksjonsteknikk og arkitektur. I de siste årene er det blitt utviklet flere plug-ins i Grasshopper som har utvidet funksjonaliteten til programvaren [35, 47].

Objekttyper i Grasshopper

Opprettelse av visuelle programmer i Grasshopper blir kalt definisjoner [35]. For å opprette definisjoner benyttes Grasshopper-objekter som samhandles med ledninger. I Grasshopper er det primært to typer brukerobjekter: parametere og komponenter. Kort forklart lagrer parametere data, mens komponenter utfører handlinger og gir resultater i form av data [35].

Parametere

Parametere er beholderobjekter som gjenkjennes av et sekskantet ikon i en rektangulær boks med en enkel input og output [35]. Dataene som lagres av parametere er bl.a. tall, geometri og farger, som sendes gjennom ledninger i den opprettede definisjonen. Eksempler på enkelte parametere illustreres i Figur 3.4 [35].



Figur 3.4 Eksempler på enkelte parametere i Grasshopper

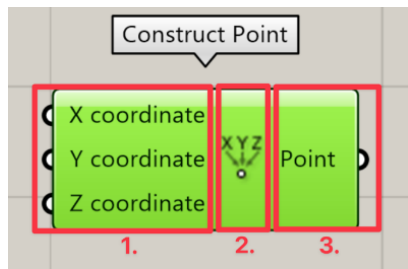
Geometriparametere kan enten få geometri fra andre komponenter, eller direkte fra geometri i Rhino [35]. Input parametere er dynamiske grensesnitt objekter som gir brukeren muligheten til å kommunisere med dataene som sendes til komponentene. Dette gjør det mulig å kontrollere og endre resultater i den opprettede definisjonen [35].

De fleste typer parametere har mulighet til å lagre to forskjellige datatyper: ustabile og vedvarende [35]. Ustabile data ødelegges hver gang løsningen utgår, noe som ofte frembringer en hendelse for å gjenopprette løsningen og oppdaterer geometrien. Data som genereres umiddelbart i modelleringen blir ansett som ustabile data. Vedvarende data er spesifisert av brukeren selv, og er avhengig av enkelte typeparametere. For eksempel kan brukeren definere ett eller flere punkt ved å bruke «point» parameteren, og en eller flere kurver ved å bruke «curve» parameteren. Disse parametere er vist i Figur 3.4 [35].

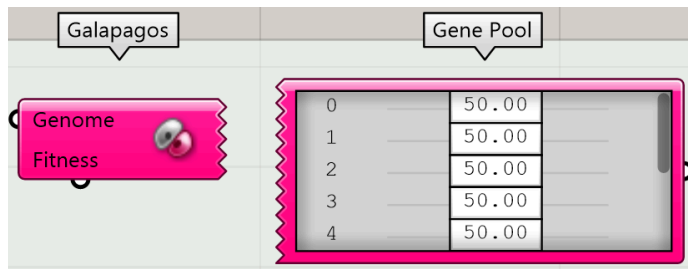
Eksempler på parametere som genererer ustabile data er input-parametere [35]. Disse parametere er bl.a. «Number Slider» og «Graph Mapper», som er vist i Figur 3.4. Input-parametere gjør det mulig å kommunisere med dataene som leveres til komponent inputene. Dette medfører at brukeren kan kontrollere og endre resultatene i den opprettede definisjonen, fordi parametere endres av inputene fra brukeren, noe som genererer de ustabile dataene. «Number Slider» er den mest brukte og viktigste input-parameteren. Ettersom den gir en verdi mellom et spesifisert ytterpunkt, og kan dermed brukes til både å identifisere endringer og vellykkede seksjoner i definisjonen [35].

Komponenter

Komponenter er objekter som krever data fra inputs-tilkoblingen for å utføre handlinger, og komme med et resultat [35]. Handlingene som utføres av komponentene kan være Rhino geometri eller operasjoner av matematiske funksjoner. De fleste komponenter har dermed et sett med både inputs- og outputs-tilkoblinger. Inputene er på venstre siden, mens outputene er på høyre side. Dette er illustrert i Figur 3.5 med komponenten «Construct point». Der punkt 1 viser tre ulike input-tilkoblinger, punkt 2 viser komponent typen, og punkt 3 viser output- tilkoblingen til komponenten [35]. Enkelte komponenter har enten kun input- eller output-tilkoblinger [35]. Dette illustreres med at komponenten har en taggete kant, som er vist i Figur 3.6 [35].

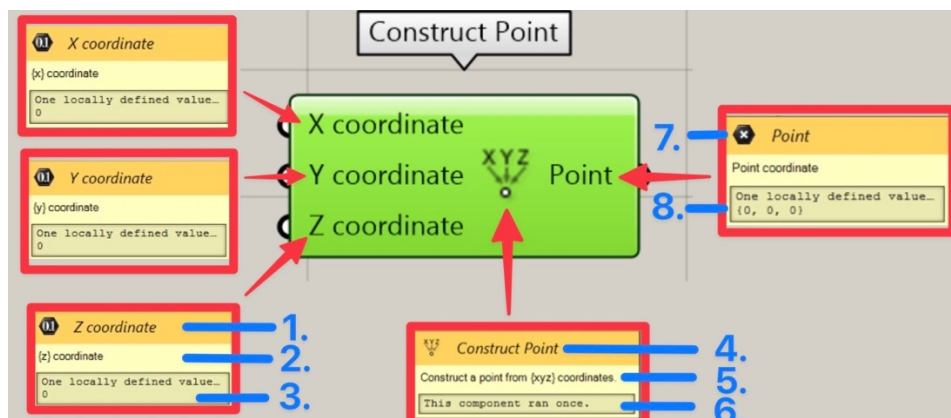


Figur 3.5: Komponenten "Construct Point".



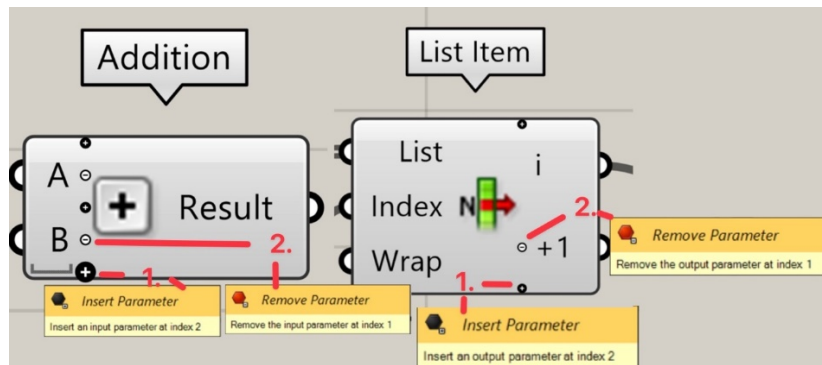
Figur 3.6: Komponenter med taggete kant

Input-tilkoblingene til komponentene forventer å motta diverse typer data [35]. Eksempelvis kan en komponent indikere at inputen må være et punkt eller en linje. Ved å holde musepekeren over de individuelle delene på komponenten, kommer det forskjellige verktøytips. Disse verktøytipsene er ganske informative, ettersom de informerer om både typen og dataene til individuelle parametere. Dette er illustrert i Figur 3.7, der punkt 1 viser ikonet til datatypen, navnet på komponenten og etiketten for inputen. Punkt 2 gir en tekstbeskrivelse av inputen til komponenten. Punkt 3. viser verdiene til inputen. Dette kan være fra tilkoblede objekter eller en gitt verdi fra komponenten. Punkt 4 viser ikonet og navnet på komponenten. Punkt 5 beskriver handlingen som utføres av komponenten med tekst. Punkt 6 indikerer antall sykluser komponenten har utført. Punkt 7 gir sammen beskrivelse som punkt 1, men for outputten. Punkt 8. viser resultatet av handlingen til komponenten [35].



Figur 3.7: Verktøytips fra komponenten "Construct Point"

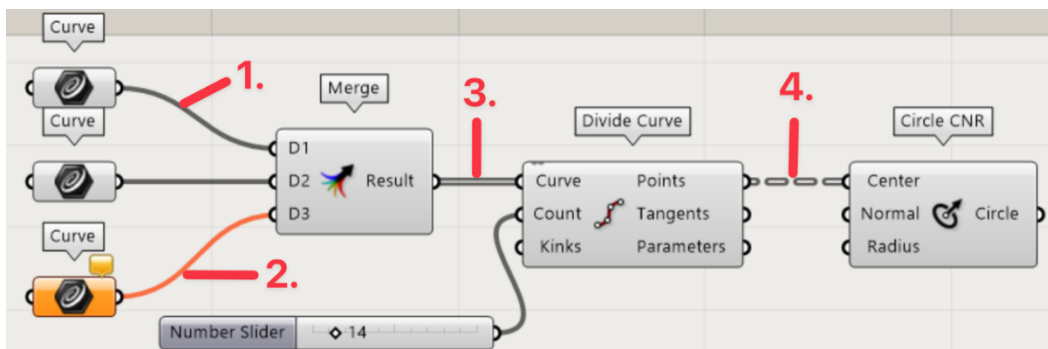
Ved å bruke «Zoomable User Interface (ZUI), kan noen komponenter modifiseres slik at antall input- og output-tilkoblinger kan varieres [35]. Dette utføres ved å zoome inn på den plasserte komponenten, til et sett med ekstra tilleggsmuligheter er tilgjengelige. I Figur 3.8 vises komponentene «Addition» og «List Item» som kan modifiseres av ZUI. Punkt 1 viser tegnet for økning av input og output, mens punkt 2 viser tegnet for reduksjon av input og outputen for hver av komponentene [35].



Figur 3.8: Zoomable User Interface (ZUI) på "Addition" og «List Item» komponentene

Ledninger

Data sendes gjennom ledninger som er tilkoblet til inputer og er koblet fra outputer til objektene i Grasshopper [35]. Ledninger representerer forbindelsen og dataflyten i den opprettede definisjonen. Grasshopper gir visuelle indiksjoner om hva som sendes igjennom ledningene. Det er fire visuelle forskjellige ledninger som forteller hvilken type informasjonen som sendes fra et objekt til et annet. Dette er illustrert i Figur 3.9, der punkt 1 viser en hel grå ledning, som sende et enkelt data element. Punkt 2 viser en oransje ledning, som indikerer at ingen data sendes. Punkt 3 viser en grå dobbel ledning, som indikerer at en liste med data sendes. Punkt 4 viser en strippet kantledning som sender informasjonen av en datastruktur [35].



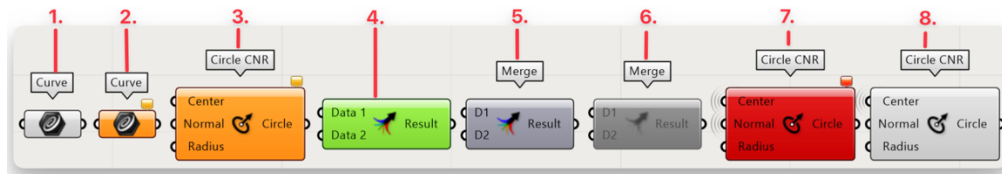
Figur 3.9: Illustrasjon av forskjellige ledninger i Grasshopper

Objekt farger

Grasshopper illustrerer tilstanden til hver av komponentene og parameterne basert på farger [35]. Dersom en parameter eller en komponent har en lys grå farge, indikerer dette at alt fungerer som det skal (se punkt 1 og 8 i Figur 3.10). Objekter som ikke klarer å innsamle data, markeres med en oransje farge, som er en advarsel om at objektet ikke bidrar med en løsning (se punkt 2 og 3 i Figur 3.10). Alle parametere som nylig innlegges i definisjonen har derfor en oransje farge. Dette indikere at de ikke inneholder data og har dermed ikke noen funksjonell påvirkning på resultatet til løsningen. Som standard har både parametere og komponenter en liten oransje snakkeboble, som presiserer hva som potensielt er problemet med definisjonen [35].

Når et objekt velges i lerretet, indikeres dette av en lysegrønn farge (se punkt 4 i Figur 3.10) [35]. Hvis det valgte objekter, har generert en geometri i Rhino, vil dette lyse med samme grønnfarge. Dersom et objekt er mørkegrå (se punkt 5 i Figur 3.10), indikerer dette at forhåndsvisningen til resultatet er

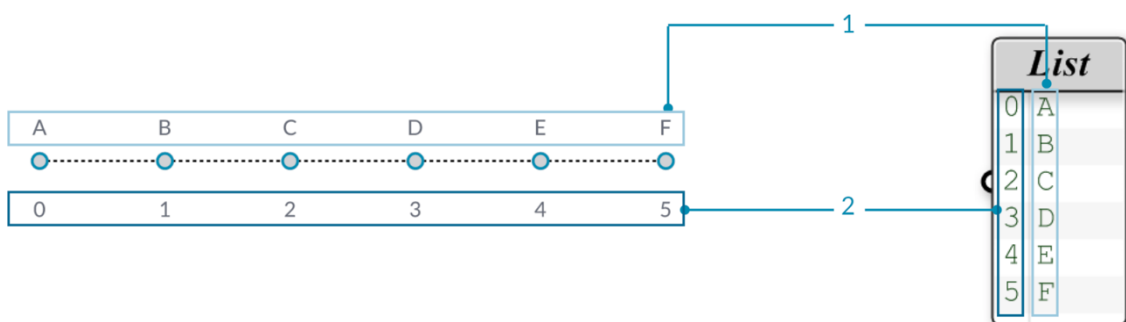
deaktivert. Et objekt vil ha en matt grå farge, når det er blitt deaktivert (se punkt 6 i Figur 3.10). Dette medfører at objektet slutter å sende data. Dersom en komponent inneholder feil, vises det med rød farge (se punkt 7 i Figur 3.10). Feilen kan enten komme fra inputene, outputene eller fra selve objektet. Den små røde boblen vil presisere feilen [35].



Figur 3.10: Objekt farger illustrert på enkelte komponenter og parametere

Lister

En kraftig funksjon i Grasshopper er muligheten for hurtig oppbygging og manipulering av lister med data [35]. Det er dataene som definerer informasjonen som sendes inn og ut av komponentene. Grasshopper lagrer de ustabile og vedvarende dataene i en matrise, som en liste med variabler. For å manipulere dataene er det viktig å forstå plasseringen til hvert element i en liste. Ved å tilkoble en «panel» i outputen til et objekt, er det mulig å få et innblikk i type data som er lagret i en liste. Som standard vises indeksnummeret på venstre side i «panel» (se punkt 1 i Figur 3.11). Indeksnummeret starter fra 0, siden matrisen benytter et nullbasert tallsystem. På høyre side i «panel» vises dataelementene (se punkt 2 i Figur 3.11). Disse elementene trenger ikke å være tall, men kan være hvilken som helst datatype som godtas av Grasshopper. Eksempelvis kan dette være bl.a. punkt, linjer, kurve eller overflater [35].



Figur 3.11: Plassering av element og indeks i en liste [35]

Data tree

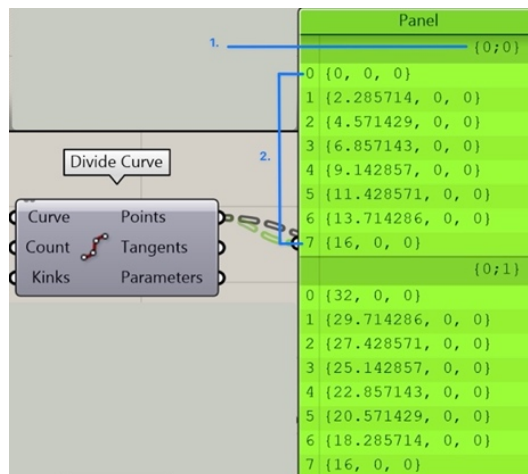
Data tree eller datatre på norsk, beskrives som en hierarkisk struktur for sortering av data i del-lister. Med andre ord er det en liste over lister. Datatre etableres når en komponent er strukturert for å innta et datasett og utsender flere datasett. Grasshopper behandler de nye dataene ved å sammenkoble dem i form av under lister. Eksempelvis fungerer disse sammenkoblede listene på samme måte som mappestrukturen til en datamaskin, der brukeren må følge stien gjennom mapper med undermapper for å finne riktig dokument.

Forskjellen mellom en liste og et datatre illustreres i Figur 3.12 [35]. Til venstre i figuren illustreres en liste med en rekke og fire kolonner som har seks punkter, nummerert opp til 21. Til høyre i figuren illustreres det samme utvalget av punkter som er i et datatre. Her er datatreet en liste med fire kolonner og hver kolonne er en liste med seks punkter. Hvert punkt har kolonne- og radnummer som defineres som indeks [35].



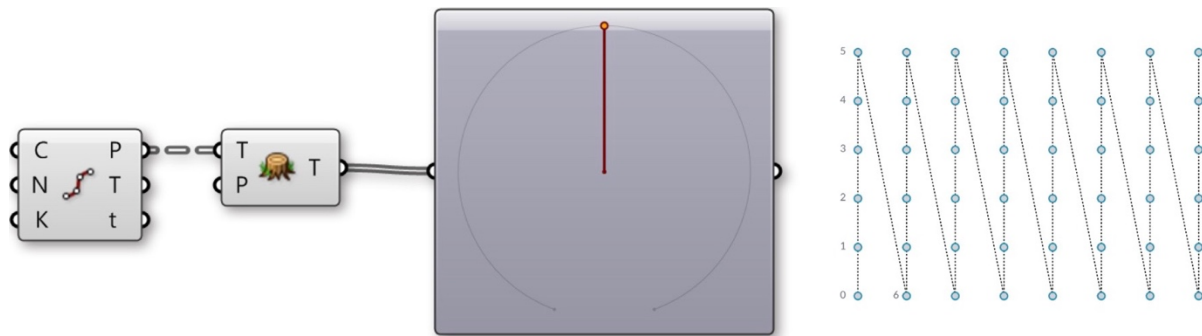
Figur 3.12: Illustrasjon av forskjellen mellom en liste og datatre [35]

Grasshopper har verktøy som visualiserer de lagrede dataene i treet [35]. Figur 3.13 illustrerer «panel» verktøyet som er tilkoblet outputen til «Divide Curve» komponenten. Punkt 1 viser stien for listen og punkt 2, indikerer alle elementene som er definerte punkt og koordinatene til hver av dem i x-, y- og z-retning [35].



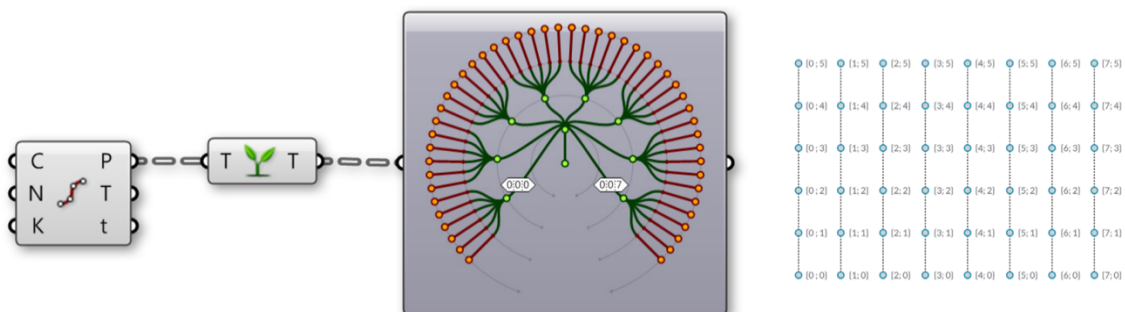
Figur 3.13: Datatre for "Divide Curve" med "Panel"

For å kunne endre strukturen til datatreet, brukes diverse verktøy som kan gi tilgang til spesifikke data i datatreet [35]. Disse verktøyene kan endre måten dataene lages, identifiseres og sendes. En av verktøyene er «Flatten», som fjerner alle nivåene i et datatre, slik det kun er en enkelt liste. Dette er illustrert i Figur 3.14, der komponenten «Flatten Tree» er tilkoblet Point-outputen til «Divide curve», og «Param Viewer» er tilkoblet Tree-outputen til «Flatten Tree». I «Param Viewer» illustreres den flattede listen, som har en gren som består av en liste med 48 punkter [35].



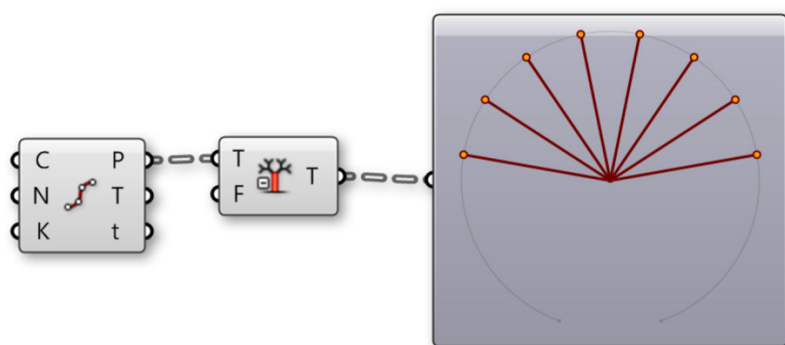
Figur 3.14: "Flatten" av en liste [35]

Ved å bruke verktøyet «Grafting» opprettes en ny gren for hvert dataelement [35]. Dersom data sendes igjennom komponenten «Graf Tree», vil hvert divisjonspunkt få en egen individuell gren. Dette er illustrert i Figur 3.15, der «Param Viewer» viser at det opprinnelig var data med 8 grener på 6 elementer, men er nå blitt 8 grener med 6 undergrenser som består av et element hver [35].



Figur 3.15: "Grafting" av en liste [35]

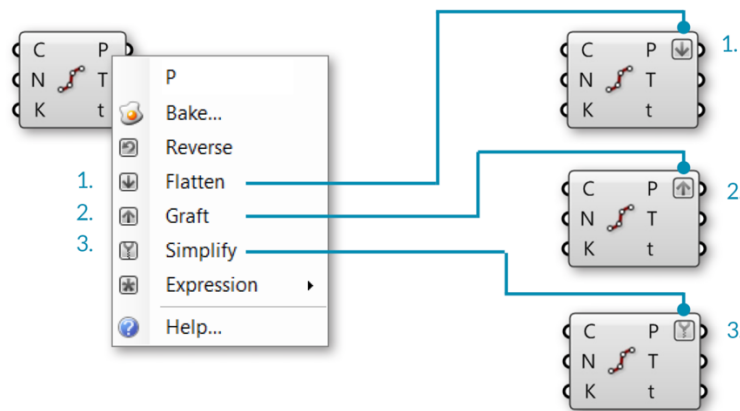
Et annet verktøy er «Simplify», som fjerner overlappende grener fra et datatre [35]. Dersom dataene sendes igjennom komponenten «Simplify Tree», fjernes den første grenen som ikke inneholder data. Dette er illustrert i Figur 3.16, der «Param Viewer» illustrerer 8 grener som har 6 elementer hver, hvor den første grenen som ble vist i Figur 3.15 er fjernet [35].



Figur 3.16: "Simplify" av en liste [35]

Både «Flatten», «Graft», og «Simplify» er tilgjengelige i selve objektene [35]. Ved å høyre klikke på enten inputene eller outputene til objektene er verktøyene tilgjengelige fra menyen (se valg menyen i Figur 3.17). Objektene vil vise et ikon, for å indikere at datatreet er endret. Dette er illustrert i Figur 3.17, der punkt 1, 2, og 3 viser ikonet for hver av verktøyene i Point-outputen. Å benytte verktøyene direkte i objektene er mer effektivt enn å tilkoble separate komponenter som var vist i Figur 3.14, Figur 3.15 og Figur 3.16. Det er viktig å huske programmeringsflyten i Grasshopper. For eksempel,

dersom «Flatten» benyttes i inputen til komponenten, flates dataene før komponentoperasjonen utføres. Dersom «Flattens» benyttes i outputen, flates dataene etter komponentoperasjonen er utført [35].



Figur 3.17: Illustrasjon av "Flatten", "Graft" og "Simplify" i menyen og ikonene [35].

3.1.5 Tilleggsverktøy i Grasshopper

I dette kapittelet presenteres forskjellige tilleggsverktøy som er essensielle for strukturelle beregninger i Grasshopper.

Karamba3D

Karamba3D er et interaktivt, parametriske Finite Element-program (FE), som er utviklet av Clemens Preisinger i samarbeid med Bollinger + Grohmann Engineers [48]. Programvaren er utviklet for å forutsi oppførselen til en struktur under ytre belastninger. Et av målene til Clemens Preisinger var å etablere et enkelt og effektivt verktøy som forenkler dataflyten mellom strukturelle og geometriske modeller [48]. Veiledningshåndboken til karamba3D 2.2.0 presiserer at programvaren er utrettet for alle behov i den tidlige designfasen for arkitekter og ingeniører [49]. Karamba3D er et kommersielt tilleggsverktøy for Grasshopper, og er innebygd i det parametriske designmiljøet til Grasshopper. Dette gjør at den har tilgang til alle geometrier og objekter i Grasshopper. I tillegg har Karamba3D også et sett med egne parametere og komponenter. Ved å sammenkoble geometrien fra Grasshopper med objektene fra karamba3D, vil programvaren utføre elementberegninger fortløpende, og resultatet vil oppdateres i sanntid med endringer av parametere. Dette gjør det mulig for å utforske forskjellige geometrier ut ifra resultatet fra karamba3D, enten manuelt eller gjennom optimaliseringsmodeller [49].

Galapagos

Galapagos er en plug-in modul i Grasshopper og er utviklet av David Rutten [50]. Modulen er en «evolutionary solver» som implementerer to generiske løsere; en genetisk algoritme og en simulert annealing-algoritme (SA). En genetisk algoritme beskrives som en metode for å løse både begrensede og ubegrensede optimaliseringsproblemer innenfor kunstig intelligens, og er basert på utviklingslæren fra Charles Darwin [51-53]. Dette handler om at de best tilpassede individene får flere avkom, og over tid utvikles nye individer som overtar, siden de er bedre tilpasset for å oppnå en optimal løsning [4, 53]. Mens en simulert annealing-algoritme er en metode for å løse bundne begrensinger og ubegrensede optimaliseringsproblemer [54]. Denne metoden er basert på den

fysiske prosessen med å oppvarme et metall, og deretter senke temperaturen langsomt for å oppnå en jevn krystallisk tilstand som leder søket til et optimalt punkt [55]. Rutten [50] påpeker at begge tilfellene har både fordeler og ulemper. SA-algoritme er bedre egnet til å navigere i komplekse strukturer, mens den genetiske algoritmen er bedre egnet til å tidlig finne pålitelige mellomløsninger. [51-53]. Dette handler om at de best tilpassede individene får flere avkom, og over tid utvikles nye individer som overtar, siden de er bedre tilpasset for å oppnå en optimal løsning [4, 53]. Mens en simulert annealing-algoritme er en metode for å løse bundne begrensninger og ubegrensede optimaliseringsproblemer [54]. Denne metoden er basert på den fysiske prosessen med å oppvarme et metall, og deretter senke temperaturen langsomt for å oppnå en jevn krystallisk tilstand som leder søket til et optimalt punkt [55]. Rutten [50] påpeker at begge tilfellene har både fordeler og ulemper. SA-algoritme er bedre egnet til å navigere i komplekse strukturer, mens den genetiske algoritmen er bedre egnet til å tidlig finne pålitelige mellomløsninger.

Galapagos er en komponent med to inputer; «Genome» og «Fitness» [56], vist i Figur 3.18. «Genome» tilkobles parametere som den kan endre verdier på, og «fitness» tilkobles et brukerdefinert mål som Galapagos vil rangere hver konfigurasjon etter. Først vil Galapagos variere verdiene i parametere helt tilfeldig, og måle dem opp mot det definerte målet. Dette gjør at den har flere rekker med tilfeldig løsninger, som er utvalg nr. 1. Deretter fjernes de tilfeldige løsningene som er lengst unna målet, og den har igjen et sett med verdier som ikke er helt tilfeldige lengre, dette er utvalg nr. 2. Deretter testes flere muligheter innenfor utvalget den har, og fjerner igjen løsninger som er lengst unna målet. Slik fortsetter den, helt til den har funnet den beste løsningen [4]. Med andre ord prøver den ikke hver eneste kombinasjon av verdier i parameterne, men den lærer av hver påfølgende runde med eksperimentering, slik at den gradvis nærmer seg den mest optimale løsningen [56].



Figur 3.18: Komponent Galapagos

Det presiseres av Rutten [50] at det ikke er noen garanti for at Galapagos vil finne den beste løsningen i en begrenset tidsperiode, og nevner at det beste man kan håpe på er løsning med akseptabel kvalitet til en akseptabel tid. Kvaliteten på løsningen til Galapagos avhenger av topologien til strukturen [50].

3.2 Tre som byggemateriale

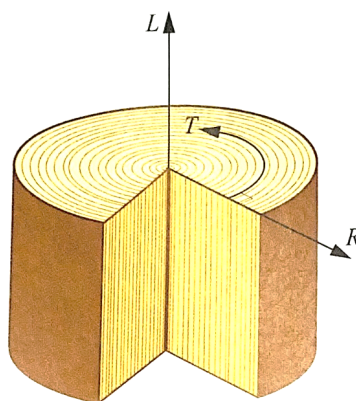
Bruken av limtre er sentralt for denne oppgaven, og derfor vil først og fremst egenskapene til limtre som konstruksjonsmateriale bli vektlagt videre i dette kapittelet. Likevel foreligger flere fellestrekk med tradisjonelt konstruksjonstrevirke av heltre.

3.2.1 Fordeler med limtre sammenlignet med tradisjonelt konstruksjonstrevirke

Trevirke er et inhomogent materiale. Det vil si at materialegenskaper som tetthet, styrke og elastisitetsmodul kan variere fra ett punkt til et annet [57]. Det svakeste tverrsnittet i ett bord, fremtrer vanligvis ved en kvist eller lignende [58]. Konstruerte tre-produkter som limtre minsker derimot spredningen i styrkeegenskapene sammenlignet med tradisjonelle bord av heltre [57, 58]. En limtrebjelke er sammensatt av minimum to bord, der tykkelsen til hvert bord ligger mellom seks og førtiseks millimeter. Hvert av bordene bindes sammen med fiberretning parallelt med lengderetningen, med lim-fuger som plasseres på tvers av fiberretningen. Dette vil bedre homogeniteten betraktelig, ettersom det svakeste tverrsnittet i hvert enkelt bord sjelden havner i samme snitt. Likevel vil limtre også ha varierende materialegenskaper innad i hver enkelt komponent, og fra en komponent til en annen [58]. Ved dimensjonering i henhold til eurokode 5, medregnes derfor en sikkerhetsfaktor γ_M for å ta hensyn til usikkerhet i materialegenskaper [58].

3.2.2 Mikro- og makrostruktur

Små prøvestykker av heltre som ikke inneholder defekter som kvister og vekstujevnheter, refererer til treets mikrostruktur. Treets mikrostruktur regnes som et anisotrop materiale, som vil si at materialet reagerer ulikt basert retningen spenninger påvirker materialet [22, 59]. Dette er særdeles komplisert å forholde seg til [22]. Heldigvis er det materialegenskapene fra makrostrukturen som er relevant for bruk i trekonstruksjoner. Treets makrostruktur refererer til større prøvestykker med defekter, og her kan en med god tilnærming gjøre forenklinger [22, 59]. På makronivå regner man nemlig materialet som ortotrop. Det vil si at materialet fremdeles reagerer ulikt basert på retningen spenninger påvirker materialet, men en regner bare med tre retninger. Figur 3.19 viser de tre ortogonale retningene: lengderetning (L) parallelt med fibre, samt radiell retning (R) og tangentiell retning (T) på tvers av fibre. I tillegg er forskjellene mellom radiell og tangentiell retning såpass små, at de regnes som like [22]. Derfor spesifiseres egenskaper parallelt med fiberretningen i materialtabeller o.l. med indeks «0», hvorimot egenskaper på tvers av fiberretningen spesifiseres med indeks «90» [57].



Figur 3.19 De tre ortogonale retningene: lengderetning, radiell retning og tangentiell retning [22]

3.2.3 Spesielle forhold som må tas hensyn til ved bruk av limtre

Fuktighet

Variasjon i fuktighetsinnhold har betydelig innvirkning på alle egenskaper i tre, og vil påvirke både styrke og stivhet [57, 58]. Særlig tydelig innvirkning er det når vanninnholdet er lavere enn fibermetningspunktet. Fuktighetsinnholdet når frem til fibermetningspunktet idet hele trevirkets mikrosystem fylles med vann, og ligger omtrent på 28% (med noe variasjon avhengig av tresort). Økende fuktighet opp mot fibermetningspunktet vil altså skape reduksjon i stivhet og styrke, samt større risiko for soppinfeksjon, da hovedsakelig ved fuktighetsinnhold større enn 20%. I tillegg er det fare for kryp ved langvarig belastning. En ytterligere økning av fuktighetsinnhold over fibermetningspunktet vil ikke påvirke materialets egenskaper, men vil i kombinasjon med høy temperatur gjøre materialet langt mer formbart og utsatt for permanent deformasjon. Ettersom fuktinnholdet varierer avhengig av relativ fuktighet, er det avgjørende å sikre ett passende fuktighetsinnhold og vise hensyn til omgivelsene konstruksjonen monteres i [57]. Ved dimensjonering i henhold til Eurokode 5 benytter man derfor klimaklasser [58].

I tillegg bør en unngå uønsket svelling og krymping, som kommer av henholdsvis lagring og frigjørelse av vannmolekyler. Krymping og svelling varierer betydelig i de ortotrope retningene, da eventuell svelling/krymping parallelt med fibrene er svært lav i motsetning til vinkelrett på fibrene. For å unngå dette bør en tilstrebe minimale fuktforskjeller [57]. Raske fuktsvingninger gir også utfordringer med sprekkdannelse. Ved et fuktsvingninger, temperaturøkning og jevnt fordelt fuktighetsinnhold i hele tverrsnittet, vil trevirket tørke raskere i overflaten enn dypere inn i tverrsnittet. Dermed vil svinnebevegelsene som oppstår på overflaten hindres fordi tverrsnittets kjerne ikke beveger seg i samme retning. Med disse spenningene, samt lav strekkfasthet på tvers av fibrene, er risikoen høy for sprekkdannelse i overflaten. Motsatt vil nedfukting skape risiko for sprekkdannelse i kjernen av tverrsnittet. Sprekker som dette vil svekke tverrsnittets skjærkapasitet, og derfor nyttes sprekkfaktoren k_{cr} i eurokodeberegninger for å sikre reduksjon i tverrsnittets effektive bredde [22].

Temperatur

Temperatur vil også føre til noe utvidelse og sammentrekninger i materialet, men har langt mindre betydning enn krymping og svelling som oppstår som et resultat av fuktinnhold. Likevel vil temperatur også endre fuktighetsinnholdet, og vil derfor indirekte være med å påvirke både krymping, svelling og krypdeformasjon [57]. I tillegg oppstår også små reduksjoner i både fasthet og stivhet ved stigende temperatur, der de største endringene forekommer i trykkfasthet og E-modul [22, 57]. Siden betydningen av temperatur først og fremst påvirker luftfuktigheten indirekte, blir den ikke tatt hensyn til eksplisitt i eurokode-beregninger. Likevel er det i noen tilfeller nødvendig å ta hensyn til termisk utvidelseskoeffisient (λ), ettersom denne er vesentlig lavere for tre sammenlignet med stål og betong. Når trevirke benyttes sammen med andre materialer, kan utvidelseskoeffisienten skape problemer ved at det oppstår dimensjonsendringer [22].

Lastvarighet og kryp

Lastvarighet har innflytelse på fastheten til konstruksjonstrevirke, hvor økende lastvarighet vil føre til at fastheten avtar. Derfor er det innført lastvarighetsklasser i Eurokode 5 [22, 58]. I tillegg vil deformasjonen øke, ved konstant last over tid. Denne deformasjonen kalles krypdeformasjon, og er

den mest kritiske av treets reologiske egenskaper [57]. Dersom lasten fjernes i etterkant, vil den etterlate en liten permanent deformasjon [22]. Krypdeformasjonen påvirkes av fuktighetsinnhold, lastens varighet, lastendring, temperatur, spenningsnivå, stivhet og kvist. Særlig vesentlig er fuktighet og lastvarighet [22, 57]. Et økende fuktighetsinnhold øker nemlig krypdeformasjonene, og ved hyppige forandringer er forskjellen særlig merkbar [22]. Ved dimensjonering i henhold til eurokodens bruksgrensetilstand, tas det derfor hensyn til serviceklasse for fukt, samt effektiv krypfaktor ($\Psi_2 * k_{def}$) for lastvarighet. De andre påvirkningsfaktorene blir ikke eksplisitt hensyntatt i Eurokode 5 [22, 58].

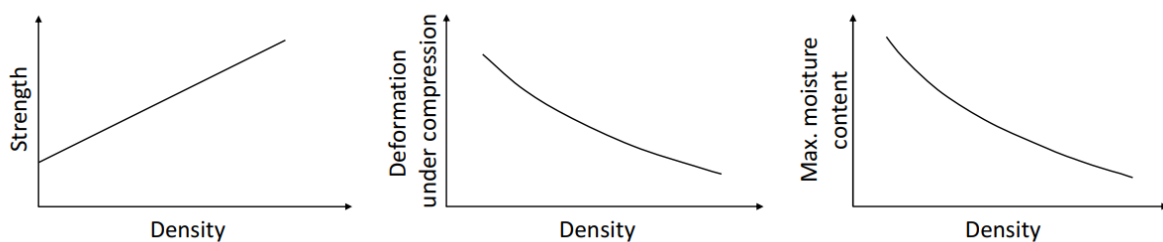
Volumeffekt

Resultater fra styrketesting viser at store stykker trevirke er svakere enn mindre stykker trevirke. Dette forklares ved at sannsynligheten er større for at en alvorlig defekt befinner seg i et kritisk punkt, desto større trestykket er. Dette er tatt hensyn til i eurokodeberegninger ved bruk av høydefaktor k_h for bøyestivhet og strekkfasthet parallelt med fibrene, og volumfaktor k_{vol} for strekkfasthet normalt på fibrene [22].

Densitet

Densitet er et forhold mellom masse og volum. Trematerialets masse og volum varierer basert på mengder vann, vanddamp, luft og matevæske [57]. Derfor påvirkes densiteten i stor grad av fuktinnhold, og må alltid spesifiseres i tilknytning til klimaet. Densitetsverdiene som benyttes i EK5 forutsetter omgivelsestemperatur på 20 grader, 65% relativ fuktighet og refererer enten til middelvei ρ_{mean} eller karakteristisk verdi ρ_k . Samtidig forutsetter styrkeklassene en normalfordelt tetthet med variasjonskoeffisient (v_p) lik 0,1 [22, 57].

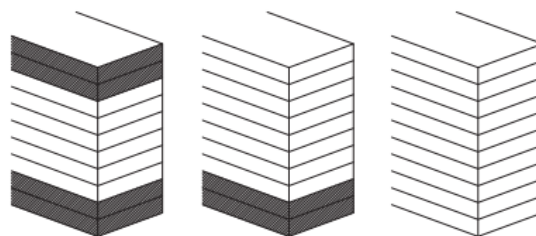
Densitet påvirker i praksis alle treets egenskaper. Figur 3.20 viser at densitetsverdien stort sett korrelerer positivt med treets egenskaper. Fremstillingen viser at trevirkets styrke øker ved stigende densitet, hvorimot deformasjon og fuktighetsinnhold reduseres ved stigende densitet [57].



Figur 3.20 Skjematisk fremstilling av hvordan densitet påvirker treets egenskaper [57]

3.2.4 Styrkeklasser

Limtre er spesielt egnet for å tåle bøyepenninger, der store deler av strekk- og trykkraftene stort sett påvirker de ytterste bordene. Av den grunn skiller en mellom homogent og kombinert limtre, som navnesettes med henholdsvis h og c , som illustrert i Figur 3.21. I homogent limtre har alle bordene samme styrkeklasse, mens kombinerte limtreklasser har lavere styrkeklasse i de midterste bordene enn i de ytterste. Som presentert i Figur 3.21. trenger ikke sammensetningen nødvendigvis å være symmetrisk. For lave tverrsnitt med inntil ti komponenter, utgjør de «ytterste bordene» minimum ett bord. De vil imidlertid utgjøre minimum to bord i et høyere tverrsnitt [57].



Figur 3.21 Symmetrisk kombinert-, asymmetrisk kombinert og homogent limtrevversnitt [57]

Standarden EN14080 definerer fasthetsklassene for limtre. Figur 3.22 er en tabell som er hentet fra limtreboka, som viser standardens styrkeklasser for kombinert limtre. Tilsvarende tabell foreligger også for styrkeklassene i homogent limtre. Fasthetsklassene er definert basert på beregninger (med utgangspunkt i egenskapene til lamellene), samt testing av et stort antall prøvestykker [58]. Som presentert i tabellen, har trematerialet store forskjeller i fasthet og stivhet parallelt og vinkelrett på fibrene. Materialet har vesentlig bedre styrke parallelt med fibrene [22]. Dette diskuteres videre i neste delkapittel.

Egenskap	Symbol	Limtre fasthetsklasse						
		GL20c	GL22c	GL24c	GL26c	GL28c	GL30c	GL32c
Bøyefasthet	$f_{m,g,k}$	20	22	24	26	28	30	32
Strekkfasthet	$f_{t,0,g,k}$	15	16	17	19	19,5	19,5	19,5
	$f_{t,90,g,k}$	0,5						
Trykkfasthet	$f_{c,0,g,k}$	18,5	20	21,5	23,5	24	24,5	24,5
	$f_{c,90,g,k}$	2,5						
Skjærfasthet (skjær og torsjon)	$f_{v,g,k}$	3,5						
Rulleskjærfasthet	$f_{r,g,k}$	1,2						
Elastisitetsmodul	$E_{0,g,mean}$	10400	10400	11000	12000	12500	13000	13500
	$E_{0,g,05}$	8600	8600	9100	10000	10400	10800	11200
	$E_{90,g,mean}$	300						
	$E_{90,g,05}$	250						
Skjærmodul	$G_{g,mean}$	650						
	$G_{g,05}$	542						
Rulleskjærmodul	$G_{r,g,mean}$	65						
	$G_{r,g,05}$	54						
Densitet	$\rho_{g,k}$	355	355	365	385	390	390	400
	$\rho_{g,mean}$	390	390	400	420	420	430	440

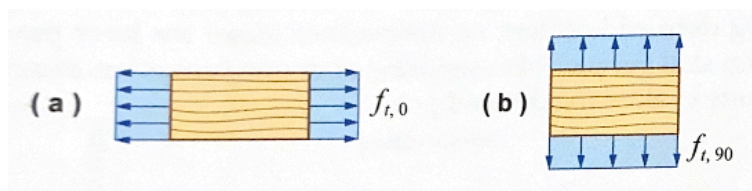
Figur 3.22 Styrkeklasser for kombinert limtre [58]

3.2.5 Limtres fasthet og stivhet

Testing for fastsettelse av materialets stivhet og fasthet må utføres på makronivå, da forsøksverdier fra mikronivå er urimelige for fulle bruksdimensjoner. Verdier fra mikronivå medregner nemlig ikke defekter som kvister og vekstujevnheter i materialet. Defekter reduserer særlig materialets strekk-egenskaper, og derfor er forskjellen i strekkfasthet mellom mikro- og makronivå parallelt med fibrene spesielt bemerkelsesverdig [22].

Strekfasthet

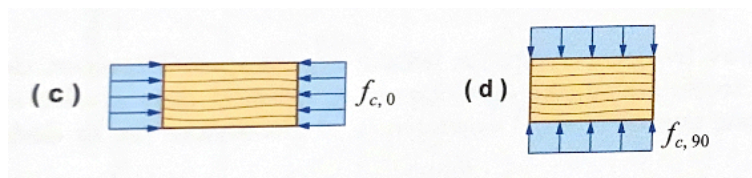
Figur 3.23 viser trevirkets ulike strekkfastheter. Strekkfasthet parallelt med fiberretning « $f_{t,0}$ » er illustrert ved Figur 3.23(a) [22]. Denne verdien er vesentlig redusert sammenlignet med tester på mikrostruktur, på grunn av veksturegelmessigheter i materialet [57]. Likevel er strekkfasthet vinkelrett på fiberretning « $f_{t,90}$ » trevirkets desidert dårligste egenskap. Denne er illustrert ved Figur 3.23 (b) [22]. Fra verdiene som er presentert i Figur 3.22, kan en fastslå at « $f_{t,90}$ » er hele 30-40 ganger lavere enn strekkfastheten parallelt med fibre [58]. Strekkspenninger vinkelrett på fiberretning bør derfor unngås, og oppstår som regel bare i buede bjelker, innsnitt eller skjøter. I tillegg kan strekkspenninger opptre med en vinkel på fiberretning, og dermed resultere i kombinerte spenningsforhold. Med dette oppstår både strekkspenninger parallelt- og vinkelrett på fiberretning, samt skjærspenninger. For å håndtere denne spenningstilstanden i henhold til Eurokode 5, utvides og omgjøres *Hankinsons* ligning [57].



Figur 3.23 Trevirkets ulike strekkfastheter [22]

Trykkfasthet

Figur 3.24 viser trevirkets ulike trykkfastheter. Trykkfasthet parallelt med fiberretning « $f_{c,0}$ » er illustrert ved Figur 3.24(c) [22]. I motsetning til strekkfasthet parallelt med fibre blir trykkfastheten langt mindre påvirket av veksturegelmessigheter i materialet, og er derfor noe høyere [57, 58]. I tillegg er trykkfasthet vinkelrett på fiberretning « $f_{c,90}$ » illustrert ved Figur 3.24(d) [22]. Trykkfasthet parallelt med fiberretning er langt større enn strekkfastheten. Dette skyldes relasjonen til fiberhelning, sprøbruddsoppførsel og volumeffekten [57].

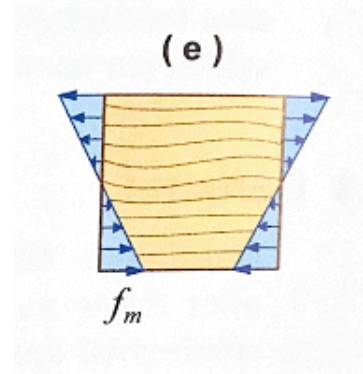


Figur 3.24 Trevirkets ulike trykkfastheter [22]

Bøyefasthet

Figur 3.25(e) fremstiller trevirkets bøyefasthet « f_m », som er en kombinasjon av strekk- og trykkfasthet. Illustrasjonen fremstiller en lineær variasjon av bøyespenningen over hele tverrsnittshøyden [22]. Bøyefasthet-verdien regnes som nominell, altså noe som fungerer teoretisk,

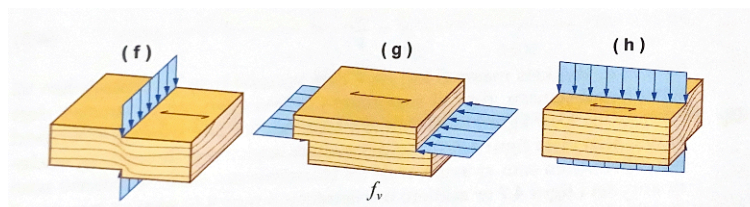
men ikke nødvendigvis i praksis [22, 60]. Bakgrunnen for dette er at spenningen i praksis vil være vesentlig mer komplisert og variert. Derfor bestemmes bøyefasthet-verdien heller som en omregning av bruddmoment basert på Euler – Bernoullis *bjelketeori*, og *Naviers hypotese* om lineærelastisk materiale [22]. Videre må det også gjøres forenklinger ved bøyning om to akser, da verdier for fasthet og stivhet for bøyning om svak akse (vinkelrett på lim-fugene) ikke eksisterer i standarden. Limtreboken anbefaler derfor å bruke samme fasthet om begge akser ($f_{m,y,d} = f_{m,z,d}$). Dette er en forenkling til sikker side, og den vil heller ikke ha nevneverdig betydning [58].



Figur 3.25 Trevirkets bøyefasthet [22]

Skjærfasthet

Figur 3.26 illustrerer trevirkets ulike skjærfastheter « f_v », hvor fasthet mot avskjæring på tvers av fibre er presentert ved (f). Spennings situasjonen som er vist, gir risiko for avskjæringsbrudd. Deretter viser Figur 3.26 (g) fasthet mot avskjæring parallelt med fibre. Et eventuelt skjærbrudd vil skape forskyvning langs fiberne. Dette oppstår som regel alltid i tilfeller hvor elementer utsettes for bøyning [22, 58]. Spenningen vil da være null i tverrsnittets over- og underkant, og størst langs nøytralaksen [58]. Til slutt illustreres rulleskjærfasthet « f_r » i Figur 3.26 (h). Denne spennings tilstanden er særlig relevant for massivtre/CLT-elementer. Størrelsen på de tre ulike skjærfasthetene er nokså ulike, der fasthet mot avskjæring på tvers av fibre er betydelig større enn de to andre tilfellene. Vanligvis har bjelker en spenningskomponent parallelt med fiberretningen, og en komponent vinkelrett på fiberretningen. Den sammenlagte påkjenningen vil medføre at fastheten mot avskjæring parallelt med fibre bestemmer verdien for « f_v » [22].

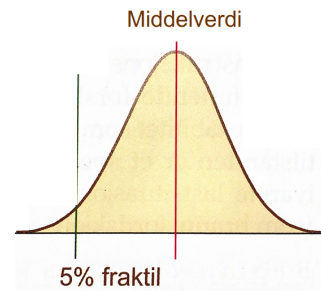


Figur 3.26 Trevirkets ulike skjærfastheter [22]

Stivhet

Limtres stivhet karakteriseres i EN14080 med elastisitet- og skjærmoduler, hvor elastisitetsmodul er et forholdstall mellom trevirkets strekk/trykk-spenninger og dets relative deformasjon [61]. Her skiller standarden mellom elastisitetsmodulens middelve- og 5% fraktil parallelt med fibre ($E_{0,mean}$ og $E_{0,05}$). Det samme gjelder for elastisitetsmodulene vinkelrett på fibre ($E_{90,mean}$ og $E_{90,05}$).

Skjærmodulen er derimot et forholdstall mellom skjærspenning og skjærdeformasjon [62]. Også her skiller standarden mellom verdier for skjærmodulens middelvei og 5% fraktil (G_{mean} og $G_{0,5}$). I Figur 3.27 illustreres en teoretisk normalfordelt elastisitet- eller skjærmodul, hvor middelveien representerer «mean»-indeksene og 5% fraktil representerer «05»-indeksene [22]. Ved beregninger i henhold til eurokode 5 er det imidlertid bare $E_{0,\text{mean}}$, $E_{0,05}$, $E_{90,\text{mean}}$ og G_{mean} som er avgjørende [22, 58].

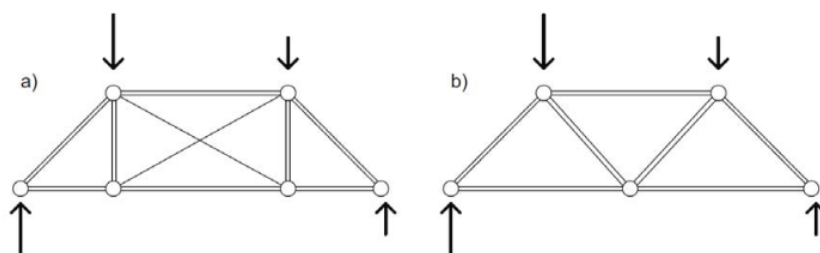


Figur 3.27 Illustrasjon av middelvei og 5% fraktil i normalfordeling [22]

3.3 Fagverk

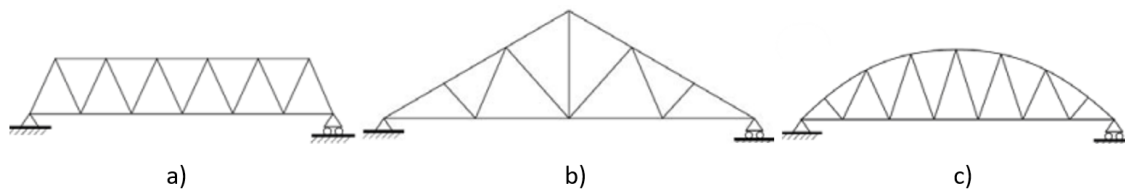
Fagverk er en konstruksjonsform som bærer via aksialkrefter. Ettersom tredimensjonale «romfagverk» ikke er relevant for oppgaven, omtales kun «plant fagverk» videre i dette kapittelet. I et plant fagverk ligger alle knutepunkter i samme plan [58], og består av «indre» staver, overgurt og undergurt. Disse komponentene danner en serie av trekanter, som kan betraktes som stive konstruksjonskomponenter. Til sammen fungerer fagverket som et stabilt system [22], som effektivt og økonomisk overfører laster frem til oppleggene [58]. Når det er sagt, bør en skille mellom mekanikkens fagverk og konstruksjonsteknikkens fagverk. Mekanikkens fagverk er et system av aksiallaste som er bundet sammen med momentfrie ledd, der kreftene kun virker i knutepunktene. Konstruksjonsteknikkens fagverk er en sammensatt konstruksjonsform, ikke en enkel beregningsmodell [22].

De «indre stavene» kan være både diagonaler og vertikaler [22]. Det er imidlertid mulig å overføre laster uten «indre staver», men da må belastningen være fullstendig symmetrisk. En bør likevel benytte kryssende strekkstag (vaiere) mellom stagene som vist i 3.28(a), eller sørge for at alle knutepunkt er forbundet med trekanter som vist i Figur 3.28(b). Disse tiltakene sikrer god stabilitet og gjør at en unngår skjevfordeling. Fagverk har ellers stor geometrisk frihet, og den resulterende utformingen er ofte et kompromiss mellom arkitektoniske, funksjonelle og økonomiske krav/ønsker. En bør dessuten tilstrebe flere geometriske fordeler ved prosjektering. Blant annet er det økonomisk lønnsomt med et lavt antall knutepunkter, da disse krever mye arbeid og skaper økt forskyvning. Samtidig er det ikke ønskelig med slanke trykkkomponenter, og vinklene mellom stav og gurt bør være så nær 45° som mulig ($\pm 15^\circ$). I tillegg bør en redusere bøyemomentet i gurtene så langt det lar seg gjøre [58].



Figur 3.28 Stabiliserende tiltak: Kryssende strekkstag eller gjennomgående trekantmønster [58]

Fagverk blir tradisjonelt brukt som bæresystem i tak, broer og master [22]. Noen fagverk utformes med parallelle gurter, som vist i Figur 3.29(a). Denne utformingen er økonomisk gunstig sammenlignet med eksempelvis massive limtrebjelker for spenn fra 20 til 25 meter. Samtidig vil en måtte vise stort hensyn til knutepunktene, da de må overføre betydelige krefter. Andre fagverk utformes med saltaksform, som vist i Figur 3.29(b). I dette tilfellet vil en del av lastene føres direkte til oppleggene via overgurten, mens stavene får beskjedne krefter. Derfor blir knutepunktene også enkle å utføre. I tillegg kan også fagverk med krum gurt benyttes, som vist i Figur 3.29(c). Med krum gurt er det mulig å realisere limtre-fagverk med spenn i overkant av 60 meter. Ettersom gurtene bærer det aller meste av den jevnt fordelte lasten, blir diagonale staver og knutepunkter billige og ukompliserte [58].



Figur 3.29 Eksempler på ulike fagverksformer. (a) Fagverk med parallelle gurter, (b) saltaksformet fagverk og (c) fagverk med krum gurt (Limtreboka).

3.4 Dimensjonering i henhold til Eurokode

Fagverk av limtre dimensjoneres i henhold til eurokode 5. Eurokode 5 består av tre deler: Allmenne regler for bygninger (EK5-1-1), brannteknisk dimensjonering (EK5-1-2) og bruer (EK5-2) [58, 63]. I denne oppgaven er det kun EK5-1-1 som er av interesse. I tillegg benyttes de nasjonalt bestemte regler og parametere som er gitt i nasjonale tillegg [58, 63].

3.4.1 Grensetilstandsmetoden og partialfaktormetoden

Ved prosjektering i henhold til Eurokode 5 benyttes grensetilstandsmetoden og partialfaktormetoden. Grensetilstandsmetoden etablerer et akseptabelt maksimalt nivå på sannsynligheten for å nå en gitt grensetilstand. Det fundamentale tilfellet av grensetilstandsmetoden er at den dimensjonerende lastvirkning som påføres en konstruksjon, må være mindre enn konstruksjonens dimensjonerende motstandsevne ($E_d < R_d$), for å unngå sammenbrudd. Kravet må verifiseres for ulike lastkombinasjoner og brudd-former. Grensetilstandsmetoden kombineres med partialfaktormetoden, hvor flere sikkerhetsfaktorer (partialfaktorer) tar hensyn til ulike typer usikkerhet som kan påvirke beregningene. Metoden er bygget på et grunnlag av karakteristiske verdier, som er basert på tydelige, karakteristiske definisjoner. Deretter vil sikkerhetsfaktorens verdi være avhengig av dens tilhørende karakteristiske verdi [58].

I bruddgrensetilstand vil f.eks. dimensjonerende permanent- og variabel last (G_d og Q_d) bestemmes av lastens tilhørende partielle sikkerhetsfaktor (γ_G og γ_Q), multiplisert med dens karakteristiske verdi. På samme måte vil dimensjonerende fasthet (f_d) bestemmes av fasthetens karakteristiske verdi, dividert med materialets partielle sikkerhetsfaktor (γ_m). Denne vil i tillegg multipliseres med en faktor (η) som tar hensyn til ytre forhold. I bruksgrensetilstand settes betingelser som at: betinget

nedbøyning, vibrasjon, glidning i forbindelse osv., ikke skal overskride en absolutt eller relativ påkrevd verdi [58].

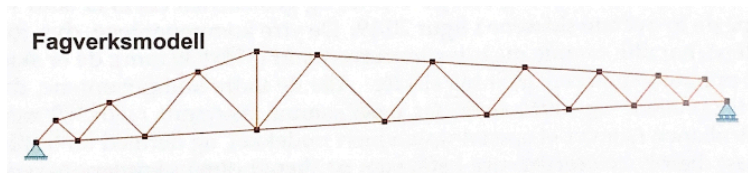
3.4.2 Laster og lastkombinasjoner

Konstruksjoner dimensjoneres for en kombinasjon av laster, f.eks. fra egenvekt, vind og snø. Hver av lastene må betraktes, en etter en, med sin fulle verdi som dominerende variabel last. Disse verdiene suppleres med de resterende lastene og deres reduksjonsfaktor. Reduksjonsfaktorene ψ_0 , ψ_1 og ψ_2 multipliseres med verdien til de variable karakteristiske lastene, som ikke er dominerende. Definerte regler for å kombinere laster i ulike dimensjonerings situasjoner er definert i Eurokode 0 [58].

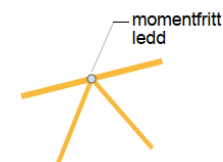
3.4.3 Lastvirkning

«Lastvirkning» er et uttrykk som omfatter f.eks. indre krefter, momenter og forskyvninger forårsaket av laster [58]. For å bestemme fagverkets indre krefter benyttes én av tre mulige beregningsmodeller: fagverksmodellen, rammemodellen eller en modell som blander de to førstnevnte. Et rammeprogram vil kunne håndtere alle tre modellene, men de er vanlig å benytte kun én av dem [64]. Videre følger forklaringer på de ulike beregningsmodellene.

Et eksempel på fagverksmodellen er skissert i Figur 3.30 [22]. Alle staver (inkl. gurtene) regnes som rene aksialelementer. Det vil si at de kun overfører aksialkrefter, og at alle knutepunkter er «perfekte» momentfrie ledd uten eksentrisiteter. Illustrasjon av et momentfritt knutepunkt er presentert i Figur 3.31. I tillegg er alle stavene individuelle, og alle laster må angripe i knutepunktene [22, 58, 64]. Fagverksmodellen gir de største aksialkreftene sammenlignet med de andre modellene, men er også relativt konservativ [22].

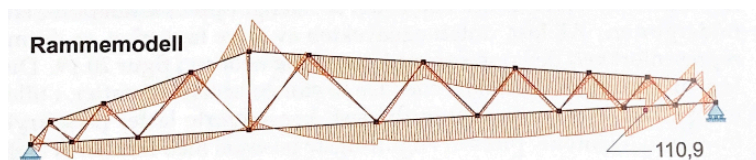


Figur 3.30 Eksempel fagverksmodell [22]

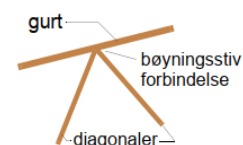


Figur 3.31 Momentfritt ledd [64]

Figur 3.32 illustrerer det samme fagverket som ovenfor, men nå med beregninger basert på rammemodellen [22]. Alle elementer regnes som bjelker med bøyestivhet, samtidig som lastene teoretisk sett kan angripe hvor som helst. I tillegg er alle forbindelser bøyings/moment-stive, som vist i Figur 3.33 [22, 64].



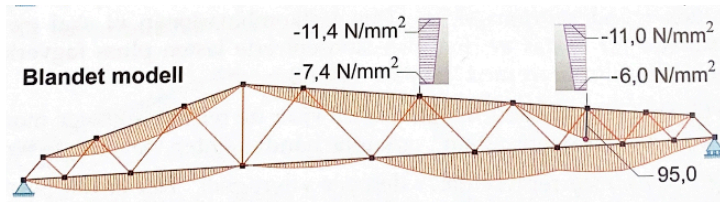
Figur 3.32 Eksempel Rammemodell [22]



Figur 3.33 Bøyingsstiv forbindelse [64]

Fagverkseksempellet beregnet med blandet modell er skissert i Figur 3.34 [22]. Dette er en sammensetning av modellene ovenfor, og er den mest realistiske. I denne modellen regnes gurtene

som bjelkeelementer, mens de «indre stavene» regnes som stavelement. Gurtene vil i dette tilfellet ha bøyingsstive forbindelser i alle knutepunkter bortsett fra skjøtene som modelleres med momentfrie ledd, som vist i Figur 3.35. Stavelementene overfører bare aksialkrefter, og mangler rotasjonsfrihetsgrad i endepunktene. Stavene regnes altså som leddet til gurtene. Også her kan lastene teoretisk sett plasseres hvor som helst [22, 64].



Figur 3.34 Eksempel Blandet modell [22]



Figur 3.35 Staver leddet til gurtene [64]

Punkt 9.2.1(5) i Eurokode 5 forutsetter at fagverksmodellen benyttes [58, 63]. De resulterende indre kreftene må da ikke overskride 70% utnyttelse i forbindelser, og det samme gjelder for strekk- og trykkspenninger. I tillegg sier Punkt 9.2.1(1) i Eurokode 5 at trykkspenninger fra kombinert bøyning og aksiallast begrenses til 90% for fagverk som hovedsakelig blir belastet i knutepunktene [22, 63]. Men ettersom konstruktøren i dag har fullgode regneverktøy, argumenteres det mot standardens bestemmelser, da de i noen tilfeller virker lite relevante og svakt begrunnet [22]. Bruk av «blandet modell» kan tenkes å være en bedre metode. Det må uansett legges begrensninger for utnyttelsen av den beregnede kapasiteten til forbindelsen, da en typisk forbindelse med innlattede stålplater og dybler ikke kan regnes som et 100% ideelt momentfritt ledd. Derfor argumenteres det for å benytte 90% av kapasiteten, ettersom dette vil ta hensyn til et beskjedent moment i forbindelsen [22, 64].

3.4.4 Klimaklasser

Ettersom trematerialets egenskaper blir sterkt påvirket av varierende fuktighetsinnhold, har eurokode 5 innført tre klimaklasser. Ved klimaklasse 1 plasseres trevirket i omgivelser med en temperatur på 20°C og relativ fuktighet under 65%. Denne klimaklassen benyttes i bygninger som blir oppvarmet, og gjelder f.eks. fagverk i kaldt, luftet loft over rom som blir permanent oppvarmet. Videre benyttes klimaklasse 2 i omgivelser med en temperatur på 20°C og relativ fuktighet over 85%. Fagverk i limtremateriale dimensjoneres derfor i henhold til klimaklasse 2, ved plassering i uisolerte bygninger som ikke blir permanent oppvarmet. Klimaklasse 3 benyttes dersom fuktighetsinnholdet i treet overskrider 20%, for eksempel i konstruksjoner som ikke er beskyttet mot regn og vann [58].

3.4.5 Lastvarighetsklasser

Lastvarighetsklasser tar hensyn til trevirkets reduksjon i styrke ved langvarig belastning. Tabell 3.1 presenterer eurokodens fem forskjellige lastvarighetsklasser [58].

Tabell 3.1 Lastvarighetsklasser i henhold til Eurokode 5 [58]

Lastvarighetsklasse	Varighet	Eksempler på belastning
Permanent last	mer enn 10 år	egenvekt
Langtidslast	6 måneder til 10 år	lagring av varer
Halvårslast	1 uke til 6 måneder	trafikklast, snø
Korttidslast	mindre enn en uke	snø, vind
Øyeblikkslast		vind, støt, eksplosjon

3.4.6 Modifikasjonsfaktor k_{mod}

Tabell 3.2 viser hvordan modifikasjonsfaktor k_{mod} er en funksjon av både klimaklasser og lastvarighetsklasser. Faktoren kan både redusere og øke limtres karakteristiske fastheter, da den varierer mellom 0,5 og 1,1. Ved lastkombinasjoner bestående av flere laster med ulik lastvarighet, benyttes verdien av k_{mod} som tilsvarer den korteste varighet [58].

Tabell 3.2 Verdier av k_{mod} (gjelder limtre, konstruksjonsvirke og LVL) [58]

Klima- klasse	Lastvarighetsklasse				
	Permanent	Langtids	Halvårs	Korttids	Øyeblikks
1	0,6	0,7	0,8	0,9	1,1
2	0,6	0,7	0,8	0,9	1,1
3	0,5	0,55	0,65	0,7	0,9

3.4.7 Bruddgrensekontroll

Bestemmelsene i bruddgrensetilstanden (ULS) skal bekrefte at en konstruksjon og dens tilhørende komponenter har tilstrekkelig lastbærende kapasitet. Grensetilstandens formål er å sikte mot sammenbrudd og alvorlig konstruksjonssvikt. Ved utførelse i henhold til disse bestemmelsene benyttes dimensjonerende verdier for laster og materialer [58].

Dimensjonerende kapasitet

I henhold til partialfaktormetoden, uttrykkes alle dimensjonerende fasthetsverdier for limtre som i ligning 3.1 [58]:

$$f_d = k_{mod} \cdot k_h \cdot \frac{f_k}{\gamma_m} \quad (3.1)$$

Hvor f_d er dimensjonerende fasthet, f_k er karakteristisk fasthet og γ_m er materialets partielle sikkerhetsfaktor. γ_m settes lik 1,15 i henhold til standardens nasjonale tillegg (NA.2.4.1(1)) [63]. Videre er k_{mod} og k_h faktorer som tar hensyn til ytre forhold [58].

Høydefaktor k_h bestemmes jf.EK5 3.2(3), og har som formål å korrigere fastheten for størrelseseffekt. Høydefaktoren gjelder kun bøyestivhet og strekkfasthet parallelt med fibre, hvor det benyttes rektangulære limtrevversnitt med høyde mindre enn 600 mm. Ved større tverrsnittshøyde eller andre fasthetsparametere, settes k_h lik 1,0. Ifølge standarden er dette en faktor som «kan» benyttes, og man regner til sikker side ved å se bort fra denne. [58, 63]. Tabell 3.3 presenterer de dimensjonerende fasthetsligningene som er særlig relevante for brudd- og bruksgrensekontroller for takstoler i limtre [64].

Tabell 3.3 Dimensjonerende fasthetsverdier relevant for fagverkskontroller

DEFINISJON	LIGNING
Dimensjonerende fasthet for bøyning om y-akse	$f_{m,y,d} = k_{mod} \cdot k_h \cdot \frac{f_{m,y,k}}{\gamma_m} \quad (3.2)$
Dimensjonerende fasthet for bøyning om z-akse	$f_{m,z,d} = k_{mod} \cdot k_h \cdot \frac{f_{m,z,k}}{\gamma_m} \quad (3.3)$
Dimensjonerende fasthet for skjær	$f_{v,d} = k_{mod} \cdot \frac{f_{v,k}}{\gamma_m} \quad (3.4)$
Dimensjonerende fasthet for trykk parallelt med fibre	$f_{c,0,d} = k_{mod} \cdot \frac{f_{c,0,k}}{\gamma_m} \quad (3.5)$
Dimensjonerende fasthet for strekk parallelt med fibre	$f_{t,0,d} = k_{mod} \cdot k_h \cdot \frac{f_{t,0,k}}{\gamma_m} \quad (3.6)$

Kontroll av fagverkets overgurt

Overgurten vil normalt bli påkjent av kombinert bøyning og trykk, og skal derfor dimensjoneres i henhold til eurokode 5, kapittel 6.3.2 [58, 64]. For takstoler er knekking ut av planet (om z-aksen) som regel forhindret av takflaten [64]. Basert på statiske beregninger, kan dimensjonerende bøye- og trykkspenninger i fagverkets overgurt finnes ved bruk av formel 3.7, 3.8 og 3.9 [64]:

$$\sigma_{c,0,d} = \frac{N_{y,d}}{A} \quad (3.7) \quad \sigma_{m,y,d} = \frac{M_{y,d}}{W} \quad (3.8) \quad \sigma_{m,z,d} = \frac{M_{z,d}}{W} = 0 \quad (3.9)$$

Deretter regnes relativ slankhet jf. 6.3.2(1) som vist i formel 3.10 [63, 64]:

$$\lambda_{rel,y} = \frac{\lambda_y}{\pi} \sqrt{\frac{f_{c,0,k}}{E_{0,05}}} \quad (3.10)$$

Hvorav slankhetstallet (λ_y) om y-akse regnes som $\frac{L_{ky}}{i_y}$. For overgurter settes kneklengde L_k lik systemlengden L , mens treghetsradius i_y settes lik $\sqrt{\frac{I}{b \cdot h}} = \frac{h}{\sqrt{12}}$ for rektangulære tverrsnitt [58].

Dersom $\lambda_{rel,y} > 0,3$ må spenningene kontrolleres jf. 6.3.2(2) og 6.3.2(3), tilsvarende formel 3.11 og 3.12 [63, 64]:

$$\frac{\sigma_{c,0,d}}{k_{cy} \cdot f_{c,0,d}} + \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \cdot \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (3.11)$$

$$\frac{\sigma_{c,0,d}}{k_{cy} \cdot f_{c,0,d}} + k_m \cdot \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (3.12)$$

Hvorav $k_{c,y}$ og k_y beregnes som formel 3.13 og 3.14:

$$k_{c,y} = \frac{1}{k_y + \sqrt{k_y^2 - \lambda_{rel,y}^2}} \quad (3.13) \quad k_y = 0,5[1 + \beta_c(\lambda_{rel,y} - 0,3) + \lambda_{rel,y}^2] \quad (3.14)$$

($\beta_c = 0,1$ for limtre)

I tillegg benyttes faktor k_m i henhold til EK5-1, kapittel 6.1.6. Denne faktoren korrigerer for tverrsnittets inhomogeniteter og tar hensyn til omfordeling av spenninger. For rektangulære tverrsnitt av limtre benyttes $k_m = 0,7$ [63]. Videre må overgurten kontrolleres for skjær i henhold til Ek5-1, kapittel 6.1.7 [63]. Basert på statiske beregninger, kan dimensjonerende skjærspenning i fagverkets overgurt finnes ved bruk av formel 3.15 [64]:

$$\tau_{v,d} = \frac{3 \cdot V_d}{2 \cdot k_{cr} \cdot b \cdot h} \quad (3.15)$$

Her benyttes sprekkfaktoren k_{cr} (jf. 6.1.7(2)) for å redusere tverrsnittets effektive bredde på grunn av sprekkdannelse ved raske fuktsvingninger. k_{cr} settes lik 0,8 for limtre. Videre kontrolleres skjærspenningene jf. 6.1.7(1) tilsvarende formel 3.16 [22, 63]:

$$\frac{\tau_{v,d}}{f_{v,d}} \leq 1 \quad (3.16)$$

Kontroll av fagverkets trykkdiagonaler og vertikaler

Staver (diagonaler og vertikaler) som er påkjent av trykk dimensjoneres i henhold til eurokode 5, kapittel 6.1.4 og 6.3.2. En må kontrollere knekking i begge retninger [58]. Basert på statiske beregninger, kan dimensjonerende trykkspenning i fagverkets diagonaler finnes ved bruk av formel 3.17 [64]:

$$\sigma_{c,0,d} = \frac{N_{y,d}}{A} \quad (3.17)$$

Trykkspenningene kontrolleres jf. 6.1.4(1) tilsvarende formel 3.18 [63, 64]:

$$\sigma_{c,0,d} \leq f_{c,0,d} \quad (3.18)$$

Videre kontrolleres spenningene i forhold til knekking jf. 6.3.2 på samme måte som ved overgurt, men for trykkstaver settes ikke knekk lengden alltid lik systemlengden L . Dersom flere staver er forbundet til et knutepunkt på gurten, vil dette skape noe innspenning. Knekk lengdene, for knekking i planet, kan i disse tilfellene reduseres til $L_k = 0,9 \cdot L$. En bør imidlertid benytte $L_k = l$, dersom knutepunktet bare består av en enkelt stav. Ved knekking ut av planet, regnes derimot systemlengden alltid lik systemlengden. Systemlengden regnes som avstanden mellom knutepunktene ved knekking i planet. Ved knekking ut av planet regnes derimot systemlengden som avstanden mellom avstivede punkter. Her kan en derfor benytte avstivingsystemer om det er nødvendig [58]

Kontroll av fagverkets strekkdiagonaler

Diagonaler som er påkjent av strekk dimensjoneres i henhold til eurokode 5, kapittel 6.1.2 [63]. For strekkstaver er det spesielt viktig å ta hensyn til det faktum at sagsnitt og dybdehull vil redusere tverrsnittet. Ved dimensjonering mot fagverksmodellen kan en, til sikker side, anta ett nettotverrsnitt « A_{net} » fra 60- til 80% av bruttotverrsnittet « A » [58]. Det er derfor vanlig å utføre strekkkontroll med effektivt tverrsnitt antatt å være 70% av brutto tverrsnitt, som vist i formel 3.19 [22, 58, 64]:

$$A_{net} = 0,7 * A \quad (3.19)$$

Basert på statiske beregninger, kan dimensjonerende strekkspenning i diagonalene finnes ved bruk av formel 3.20 [64]:

$$\sigma_{t,0,d} = \frac{N_{y,d}}{A_{net}} \quad (3.20)$$

Strekkspenningene kontrolleres jf. 6.1.2(1) tilsvarende formel 3.21 [63]:

$$\frac{\sigma_{t,0,d}}{f_{t,0,d}} \leq 1 \quad (3.21)$$

For de mest påkjente strekkstavene må en også kontrollere uttrekkingsbrudd (strekk-skjær-brudd). I tillegg stikk-kontroller på at en ikke får stålpatebrudd [22]. Disse brudd-formene forklares ytterligere under kapittel 3.5.

Kontroll av fagverkets undergurt

I undergurten er momentkreftene stort sett lave [64]. Det er derfor mest avgjørende å kontrollere undergurten for strekk i henhold til eurokode 5 (kapittel 6.1.2.), på samme måte som strekkstavene [63]. I noen tilfeller er det også nødvendig å kontrollere for sideveis knekking, dersom undergurten blir utsatt for vind-sug som løfter oppover (ytterligere enn den permanente egenlasten trykker nedover) [58].

3.4.8 Bruksgrensekontroll

Bruddgrensetilstandens bestemmelser (SLS) har som formål å tilfredsstillte definerte brukskrav som nedbøyning og vibrasjoner [58]. Ved bruksgrensekontroll for fagverk må en kontrollere at største nedbøyning fra brukslaster er mindre enn tillatt [58].

3.5 Forbindelser

Her gjennomgås teori for forbindelser i trekonstruksjoner. Det er valgt å gå dypt innen dette området, ettersom forbindelser i trekonstruksjoner er ofte kritiske for konstruksjonen.

3.5.1 Forbindelser i trekonstruksjon

Kolbein Bell [22] presiserer at forbindelser er den vanskeligste delen innen trekonstruksjon. Dette kommer av at en forbindelse sammenbinder flere enkeltkomponenter i konstruksjon, og vil dermed ofte bestemme bæreevnen til hele konstruksjonen [58]. Hvis en forbindelse er feil utformet, kan dette føre til plutselig brudd uten noen forvarsel. Derfor er både kapasiteten og duktiliteten til forbindelsen utslagsgivende for i hvilken grad kritiske situasjoner kan oppstå uten eller med forvarsel. På grunn av dette presiseres det i Limtreboka at forbindelser er ofte trekonstruksjonens svakeste punkt [58].

Det finnes flere forskjellige typer forbindelser som kan utføres på mange ulike måter, men de mest økonomiske løsningene er ofte standardløsninger [58]. For de fleste limtrekonstruksjonene er imidlertid standardløsningene ikke de mest økonomiske gunstige løsningene. Siden størrelsen på de overførte kreftene er for store, må det derfor designes spesielle løsninger, som Limtreboka påpeker er vesentlig dyrere [58].

I de fleste forbindelsene benyttes det ofte metaller eller lim [58]. Tidligere ble det kun benyttet trevirke, men på grunn av arbeidskraften ble dyrere og størrelsen til konstruksjonene vokste, ble det nødvendig å ha effektive og sikre forbindelser [22]. I limtrekonstruksjoner brukes det stålplater sammen med dybler, bolter, skruer eller spiker [58].

Ved dimensjonering av trekonstruksjoner er det viktig at konstruktøren undersøker hvordan den enkelte forbindelsen overfører og håndterer de forventede kreftene fra den statiske beregningen [58]. Det er spesielle forhold som må tas hensyn til ved utforming og dimensjonering av forbindelsene i trekonstruksjoner. Ettersom tre er et hygroskopisk materiale, må utformingen av forbindelsen ta hensyn til bevegelsene fra svelling og krymping som skyldes variasjoner i relativ fuktighet. Det er ofte forskjellige deler i forbindelsen som ikke reagerer på luftfuktigheten på samme måte som tre, noe som kan skape spenninger. Dermed er det viktig å grundig undersøke den

geometriske utformingen og samvirket mellom delene i forbindelsen. I tillegg på grunn av den lave styrkeegenskapen som trevirket har normalt på fiberretningen, må belastningen reduseres mest mulig i denne retningen [58].

I forbindelser benyttes det vanligvis metalldele for å overføre krefter på en sikker og effektiv måte, men disse kan bli det svake leddet i forbindelsen. Ettersom metallet kan korrodere og bæreevnen reduseres ved høy temperatur [58]. Geometrien til forbindelsen kan også svekke trevirket, noe som må vurderes i dimensjoneringen. Innleste stålplater, skruer og lignende kan i tillegg redusere tverrsnittet til trekonstruksjonen, noe som limtreboka påpeker vanligvis oppstår [58].

Forbindelser i trekonstruksjoner er hverken fullstendig momentstive eller momentfrie, og det vil alltid være en viss rotasjon i forbindelsene ifølge limtreboka [58]. Selv om det ikke er mulig oppnå fullstendige momentstive forbindelser, kan en nesten momentfrie forbindelse oppnås ved å utføre selve forbindelsen av metall. [58]. Ved dimensjonering og utforming av forbindelser i trekonstruksjon er det viktig å ta hensyn til slike faktorer.

Ved håndberegninger av forbindelser, er det vanlig å anta momentfrie ledd [58]. Dette gir ofte mindre beregningsarbeid og resultatet er ofte overdimensjonerende, men dette er ikke tilfellet hver gang. I beregningsprogrammer finnes det flere valgmuligheter. Limtreboka bemerker viktighet med å eksperimentere med forskjellige stivheter for å undersøke hvordan stivheten påvirker resultatet, og for å finne hvilke stivheter som gir rimelige bevegelser i forbindelsene. Videre anbefaler limtreboka at det bør modelleres med en rotasjonsstivhet mellom 50-85% i en momentstiv forbindelse [58].

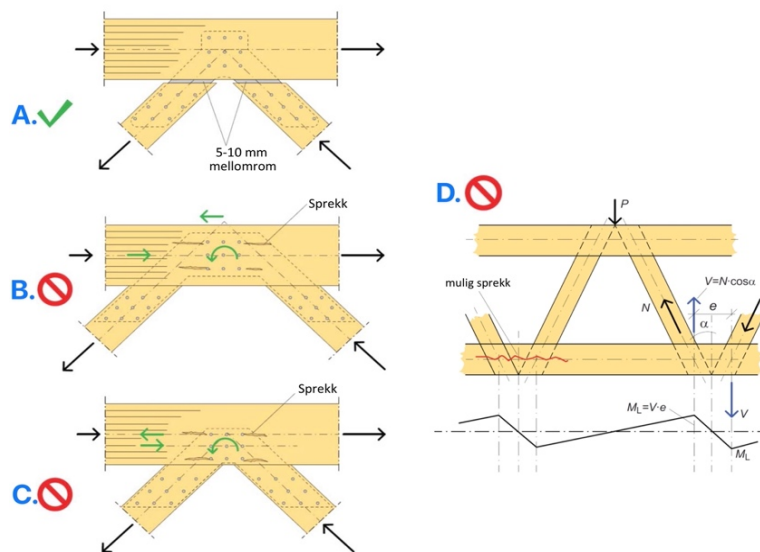
3.5.2 Fagverksforbindelser

I fagverksforbindelser benyttes stålplater sammen med spiker, skruer, bolter eller dybler [58]. Stålplater kan festes enten utvendig eller bli innslisset i tre-elementet. Ved dimensjonering og utforming av forbindelser i limtre-fagverk er det to forhold som trenger stor oppmerksomhet [58].

For det første må forbindelsen ha mest mulig utstrekning [58]. En måte å oppnå dette på er å bruke innslissede stålplater med dybler eller bolter, som vil øke belastningskapasiteten og reduserer utstrekningen i forbindelsen [65, 66]. Dette kommer av at stålet tar opp kreftene fra belastningen og fordeler dem jevnt over konstruksjonen. Ved å øke antall diagonaler i fagverket, kan utstrekningen i forbindelsen også reduseres. Diagonaler er stive elementer som begrenser deformasjoner i fagverket og bidrar til å jevnt fordele kreftene. Imidlertid vil økningen av antall diagonaler påvirke stivheten og vekten til fagverket. I tillegg vil økning av antall diagonaler øke kostnaden til konstruksjonen. Derfor må en balanse mellom antall diagonaler og andre designfaktorer vurderes [67-69]. For det første må forbindelsen ha mest mulig utstrekning [58].

For det andre må eksentrisiteter unngås [58]. Siden tverrestrekkspenninger kan oppstå som følge av sekundære bøyemomenter, som kan frembringe sprøbrudd i knutepunktene. Dette unngås ved at stavenes senterlinjer skjærer hverandre i det samme knutepunktet, som er vist i Figur 3.36-A. Figur 3.36-B viser hva som skjær dersom systemlinjene ikke skjærer hverandre i det samme punktet. Figur 3.36-C viser at dersom rotasjonssentrert til dybelgruppen i gurten er plassert under skjæringspunktet

til stavenes systemlinjer, vil det oppstå eksentrisitet. Figur 3.36-D viser eksentrisitetsmomentene (M_L), og mulig sprekkdannelse i et feil utformet knutepunkt [58].



Figur 3.36: Én riktig og flere utformingsfeil av et fagverkknutepunkt med stålplater og dybler [58].

Noen ganger kan det være gunstig å utforme forbindelsen med moderat eksentrisitet [58], spesielt for store fagverk med mye strekk i undergurten. Eksentrisiteten vil påføre et moment i undergurten, som kan redusere eller øke påkjenningen på gurten. I slike tilfeller vil beregninger vise hva som er mest gunstig. Ved moderat eksentrisitet kan mindre plater benyttes, noe som vil redusere dybden på innsliiset, og dermed mindre reduksjon av limtretversnittet. Det er presiseres i Limtreboka at det er uansett viktig at platene og dybel-konfigurasjonen er utført uten eksentrisitet [58].

3.5.3 Forbindelser mellom stav og gurt

I limtre-fagverket brukes vanligvis stålplater og stavforbindelser. Stålplatene kan enten plasseres utvendig eller innsliises inn i limtreet. For lange spenn og med stor belastning benyttes det vanligvis innsliisde plater med stavdybler i knutepunktene [58]. Dette gjelder også når tverrsnittet til staven er mindre enn gurten [22]. Dette skyldes at innliisde plater med dybler kan overføre store krefter i systemlinjene, og redusere deformasjonen. Utvendige stålplater kan også benyttes, men fuktighet kan lettere samle seg mellom tredelen og dyblene, noe som reduserer levetiden [68].

Normalt benyttes stålplater med tykkelse på 8 mm og dybler med en diameter på 12 mm i store limtre-fagverk [58]. Disse ståldelene er vanligvis konstruksjonsstål med kvalitet S355. For å oppnå optimal styrke og tilfredsstillende duktilitet i forbindelsen, bør avstandskravene for stålplatene og dyblene utføres i henhold til kravene fra EK5. Det er spesielt viktig å være oppmerksom kombinasjonen av flere stålplater og et stort antall dybler, siden dette kan føre til forskjellige bruddmekanismer [58].

I en rapport fra vegdirektoratet nevnes det at i ideelle limtre-fagverk går staver fra knutepunkt til knutepunkt [68]. Hvis gurtstavene forsetter i samme retning, fungerer knutepunktet som en skjøt, og som oftest får gurtstavene de største kreftene. Det kan derfor være ideelt å la gurtstavene være

gjennomgående i flere knutepunkt. Dette medfører at fagverket ikke er et helt ideelt fagverkt, men ved å gjøre stavene så lange som mulig, reduseres antall skjøter [68].

Ved å skjøte lange gurtstaver kan det virke fordelaktig å legge skjøten i knutepunkter som allerede har innslissede stålplater. Bell [22] påpeker imidlertid at dette vil skape et uoversiktlig og kompliserte kraftoverføringer i knutepunktet. I tillegg skapes det større belastninger i skjøten, som medfører mer spenning og deformasjoner. Det anbefales derfor å legge gurtskjøten midt mellom knutepunktene, slik at skjøten kun utsettes for rent strekk og ikke har noe momentvirkning [22, 68].

3.5.4 Stavdybler

Dybler fungerer på samme måte som bolter, men de kan ikke ta opp aksialkrefter og mangler den samme klemvirkningen som bolter har [22]. Dybler er derfor ikke egnet for enkeltsnittende forbindelser, fordi en eksentrisk lastinnføring kan få tre-delene til å gli fra hverandre og løsne dybelen [22]. Bolter med innslissede stålplater er ofte brukt i trekonstruksjoner, men tradisjonelle bolteforbindelser har noen svakheter. Studier viser at bolteforbindelser har lav rotasjonstivhet, utilstrekkelig momentkapasitet og kan være uforutsigbare ved sprøbrudd [70]. En studie utført av He m.fl. [71] viser resultatet at initiale sprekker betydelig reduserer den initiale stivheten og momentkapasitet til boltede ledd. Reduksjonen var mellom 15% til 52%. Videre viser studiet at leddets duktilitet påvirkes av plasseringen av de initiale sprekkenes, spesielt for ledd med moderat duktilitet. Reduksjonsforholdet var opp mot 61%, og plasseringen av de initiale sprekkenes kunne endre sviktmodusen fra duktilt brudd i til sprøtt brudd. Shu m.fl. [72] påpeker at den lave rotasjonsstivheten kommer av at diameteren til det forbore boltehullet er større enn boltediameteren. Dette fører til innledende glidning inne i skjøten, som igjen gir begrenset duktilitet og dårlig fordeling av kreftene i de boltede forbindelsene [72].

I en dybel-forbindelse er ikke diameteren til det forbore dybel-hullet i treet større enn selve dybelen [22]. Studier viser at dybler har bedre rotasjonsstivhet og en optimal duktil oppførsel ved vekslende belastninger [72]. Dybler i kombinasjon med stålplater, bidrar derfor til en betydelig stivhet i konstruksjonen, og kan overføre langt større krefter i knutepunktene enn det bolter kan [68, 73]. Dette er noen av grunnene for at dybler foretrekkes i store limtrekonstruksjoner [22].

3.5.5 Tau effekt

Når en stavforbindere bøyes og det skapes en vinkel mellom stavens opprinnelige posisjonen, og den deformerte staven ved brudd, mobiliseres tau effekten [22]. Dette skyldes den vertikale komponenten fra aksialkraften som bidrar til å overføre skjærkraften mellom trestykkene [22]. Tau effekten oppstår ikke bare av dette, men også av friksjonen mellom stavforbindere og tredelen [74]. Friksjonskreftene øker styrken i forbindelsen ved å tillate overføring av større belastningskrefter. I tillegg økes tau effekten av uttrekningsmotstanden til festene. Dermed utgjør overføringen av skjærkraft fra vertikalkomponenten og friksjonen mellom stålet og tredelene den såkalte tau effekten [74].

I Eurokode 5 er tau effekten satt til 0% for dybler [63], siden tau effekten er beregnet av den aksiale bæreevnen til festemidlene. Dybler er ikke egnet til å ta opp aksialkrefter alene, siden det er

forbindelsen som opptar denne belastningen. I tillegg har glatte dybler, ingen uttrekningsmotstand, noe som også er grunnen for at taueffekten neglisjeres for dybler [75].

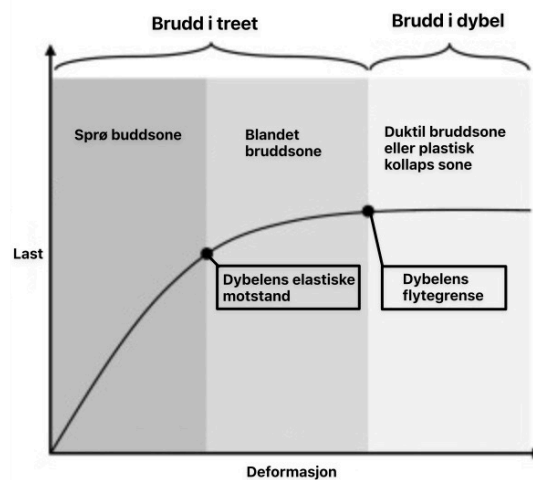
3.5.6 Bruddmekanismer i treet

Innenfor design av konstruksjoner er hovedkravet at konstruksjonen skal ha en duktil oppførsel bruddgrensetilstanden [76]. I trekonstruksjon er det spesielt viktig å oppnå høy duktilitet, siden sprøbrudd er en risiko knyttet til sprekker i treet, og kan oppstå i forbindelser. Dette medfører redusert duktilitet, og hvis dette oppstår, kan konstruksjonen plutselig kollapse [77].

Utformingen av dybel-forbindelser i trekonstruksjoner er godt etablert i EK5, ved bruk av Johansens flyteleddsteori [78]. Denne teorien forutsetter at dybel-festene er tilstrekkelig slanke, og vurderer svikt i både dybel-festet og i hullkantfasthet til trevirket [78].

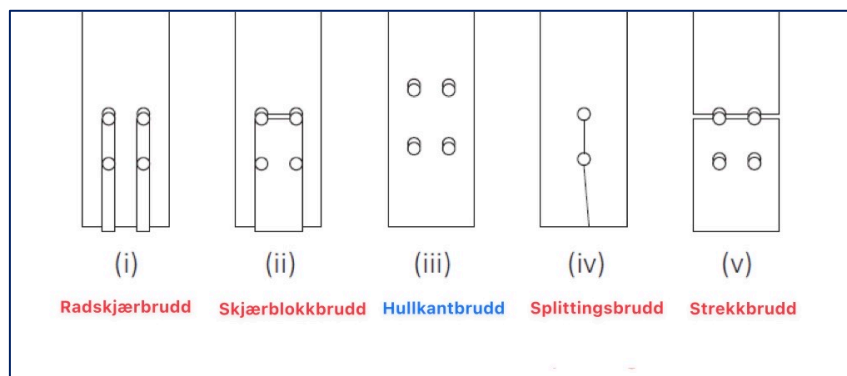
Hvis det brukes slanke dybler med liten diameter i forbindelser med høy kapasitet, kreves det et høyt antall av dybler [78]. For å unngå dette, kan diameteren til dyblene økes, og stålplater bør benyttes som mellomkomponenter i forbindelser med høy kapasitet. Dette er bedre enn å koble treet sammen med mange dybler. Når dybel-diameteren økes, vil også stivheten til dyblene økes. Dette fører til at stivheten til dyblene blir større enn hullkantfastheten til tredelen. Stålet til dyblene kan dermed ikke oppnå flyt, og det oppstår en ettergivelse i hullkantfastheten til tredelen. Dette medfører at forbindelsen ikke vil oppføre seg duktilt [77-79]. Yuttita m.fl. [77] beskriver dette som duktilt brudd eller plastisk kollaps [80], siden både stålet og hullkantfastheten betraktes som plastisk deformasjon med økt duktilitet i dette tilfellet. Hanhijärvi m.fl. påpeker at i forbindelser med stive og stor dybel-diameter, kan føre til både sprøtt og duktilt brudd i knutepunktet. Dette påpeker Yuttita m.fl. at også gjelder for dybler med liten diameter, som belastes parallelt med fiberretningen [77, 78]. I EK5, anbefales det at dybel-diameteren må være større enn 6 mm og mindre enn 30 mm [63].

I dybelforbindelser kan duktilt og sprøbrudd oppstå, det er derfor viktig å kartlegge og fastslå de mulige bruddmekanismene som kan fremkomme i knutepunktet [79]. Bruddformen i knutepunktet avhenger av utformingen, kapasiteten og fastheten til de ulike elementene i forbindelsen, og kan variere mellom sprø, blandet eller duktilt. Zarnani m.fl. [81] har illustrert disse bruddformene i Figur 3.37, som viser en last-deformasjonskurve for tre-elementer. Figuren viser at i den sprø bruddsonen er deformasjonen til dybelfestet innenfor det elastiske området. I denne sonen vil tre-komponentens effektive tykkelse for forbindelsen samsvare med den elastiske deformasjonen til dybelfestet. Kapasiteten til tredelen i forbindelsen er mindre enn flytegrensen til dybelene i denne bruddsonen. Etter hvert som deformasjonen forsetter mot flytegrensen til dyblene, vil den effektive tre-tykkelsen reduseres. Hvis kapasiteten til tredelen ikke kan motstå lasten og den lave deformasjonen fra dyblene, vil det plutselig oppstå brudd i treet før flytegrensen til dyblene er oppnådd. Dette kalles blandet brudd, siden det er et brudd i treet og et duktilt trefestebrydd [81, 82].



Figur 3.37: Potensielle bruddmekanismer i en dybel-forbindelse, basert på belastningen i forbindelsen [81]

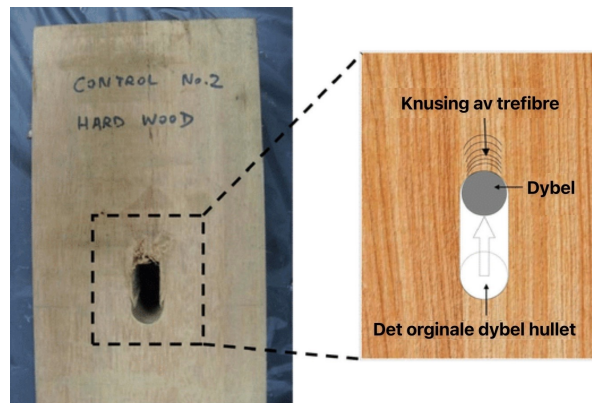
Når forbindelser belastes parallelt med fiberretningen kan mange og varierende bruddmekanismer oppstår [83]. Det er fem potensielle bruddmekanismer for dybel-forbindelser som er; radskjærbrudd (i), skjærblokkbrudd (ii), hullkantbrudd (iii), splittingsbrudd (iv), og strekkbrudd (v). Disse er illustrert i Figur 3.38 [83]. Hullkantbrudd er den eneste duktile brudd formen, fordi denne er forårsaket av trykk knusing av treet. De fire andre er karakterisert som sprø brudd former [78].



Figur 3.38: De fem potensielle bruddmekanismer for dybel-forbindelser [83].

Hullkantsvikt

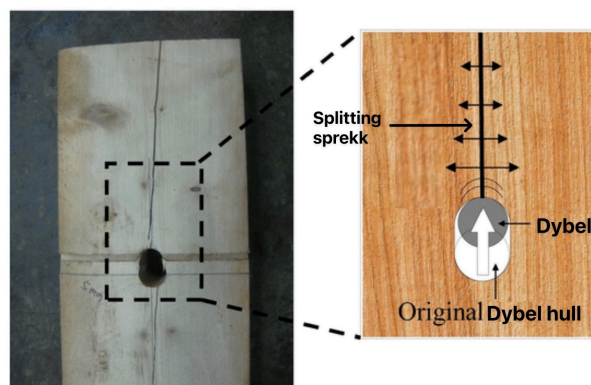
Hullkantfasthet er avhengig av flere faktorer som densiteten til trevirket, dybel-diameteren, vinkelen mellom kraft og fiberretningen, fuktinnhold, friksjon mellom dybel og treet og av eventuelle forsterkinger vinkelrett på fiberretningen til treet [22]. Når kapasiteten til hullkantfastheten er mindre enn den påførte belastningen, medføre dette knusing av treet som er rundt dybelen som kalles hullkantsvikt [78, 79]. Dette er vises i Figur 3.39, og som nevnt i kapittel 3.5.6 anses som en duktil bruddform siden det involverer plastisk deformasjon både i treet og dybelen [77]. Johansesnteorien som er implementert i EK5, vurdere denne bruddformen som ønskelig på grunn av dens duktile oppførsel [78, 79]. For å unngå hullkantsvikt, bør analysene i EK5 følges [79].



Figur 3.39: Eksempel på hullkantsvikt [84].

Splittingsbrudd

Splittingsbrudd oppstår når strekkspenninger påføres vinkelrett på fiberretningen [58]. Det er flere grunner for at splittingsbrudd kan oppstå. For det første kan splitting være en sekundær virkning fra hullkantfastheten til tredelen, der dybler eller det knuste trevirket blir neddratt imellom fibrene til treet og dermed forårsake splitting. Dette er vist i Figur 3.40. For det andre kan spekker oppstå ved dybelhullene som er nærmest enden til tre-elementet eller i selve enden, som kommer av skjær- og strekkspenninger vinkelrett på fiberretningen [78, 79]. For det tredje kan momentbelastning også påføre strekk vinkelrett på fiberretningen. Limtreboka påpeker at ved dimensjonering og utforming av trekonstruksjoner må det alltid undersøkes måter som reduserer strekkspenninger vinkelrett på fiberretningen [58].



Figur 3.40: Eksempel på splittingsbrudd [84].

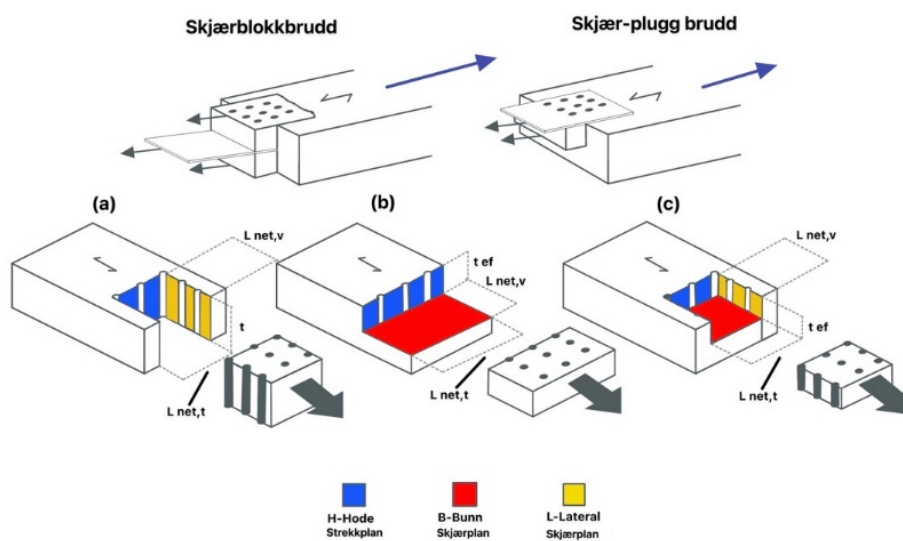
Ifølge Yurrita m.fl. [77] er det størst risiko for splittingsbrudd i forbindelser med en rad av festemidler. For å redusere risikoen for splittingsbrudd bør antall rader med festemidler økes. Dette vil også redusere risikoen splittingsbrudd oppstår i kun en rad [77].

Skjærblokkbrudd

I stål-mot-tre forbindelser med flere dybler, kan det oppstå spesielle bruddformer når tre-komponenten blir utsatt for en strekkraft som er vinkelrett på fiberretningen [22]. Disse bruddformene inkluderer skjærblokk- og skjærpluggbrudd (se øverst Figur 3.41), som begge kan oppstå av at skjær- og strekkapasiteten til trevirket overstiges. Dette fører til at treet ryker i en omkrets rundt dybel-festet [22], og et volum av trevirke frigjøres i dybel-festet på grunn av svikt i tre forskjellige plan. Disse bruddformene er illustrert i Figur 3.41-c. H-planet er et strekkplan som er

markert med blå, mens B-planet og L-planet er skjærplan som er markert med rødt og gult. Disse bruddformene er klassifiseres som sprø eller blandet brudd, og bestemmes av kapasiteten til tredelen [77].

Forskjellen mellom skjærblokkbrudd og skjærpluggbrudd avhenger av diameteren og lengden til dyblene i tredelen [77]. Når det er både en stor dybel-diameter og lang dybel-lengde, dannes ikke flyteledd i dyblene, ettersom dyblene har stor kapasitet og stivhet. Dette resulterer i et skjærblokkbrudd som er illustrert i Figur 3.41-a, der hele dybelfestet trekkes ut fra tredelen. Når det derimot er en liten dybel-diameteren og kort dybel-lengde, kan dette føre til slanke og flyteledd i dyblene. Dette resulterer i skjærpluggbrudd som er illustrert i b og c i Figur 3.41. I skjærpluggbrudd vil den effektive tre-tykkelsen t_{ef} uttrekkes fra tredelen, mens den resterende tredel blir gjenstående [77, 79].

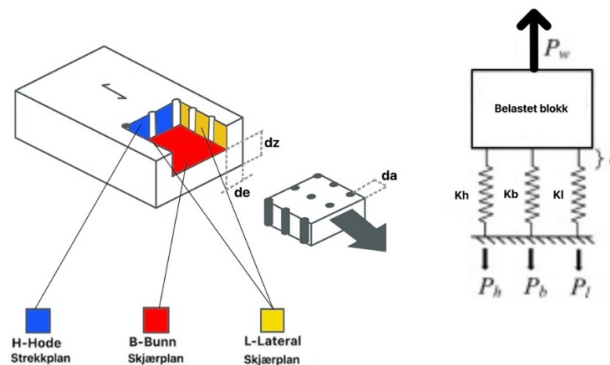


Figur 3.41: Illustrasjon av skjærblokkbrudd og skjærpluggbrudd, pluss illustrasjon av svikt i tre plan [77, 79].

Zarnani m.fl. [81] har undersøkt utrivningsmotstanden til tre-elementet når de belastes parallelt med fiberretningen. Zarnani m.fl. sammenligner utrivningsmotstanden med et lineært elastisk fjærsystem, som er illustrert på høyre side i Figur 3.42. I modellen representerer fjærene stivheten til tre-blokken som belaster bruddplanene H, B og L i Figur 3.42. Modellen forutsetter at belastningen jevnt fordeles på fjærene K_h , K_b og K_l i forhold til styrke deres, og belastningsblokken ryker når kapasiteten til fjærene overstiges. Modellen kan forutsi stivheten til hver side av motstandsblokken og beregner fordelingen av kreftene som overføres til hvert plan i blokken. Dette gir en antydning om hvilke bruddplaner som mest sannsynlig vil svikte i den belastede forbindelsen [81].

Svikten i bruddplanene kommer av stivhetsforskjellen mellom strekk- og skjærplanene, som fører til ujevn lastfordeling mellom motstandsplanene [81]. Forskjellen mellom strekk- og skjærplanene avhenger av flere faktorer. Dette er elastisitetsmodulen, stivhetsmodulen, volumet av de ytre tredelene som omgir bruddplanene (avstander til bunn- d_z , ende- d_a , og kanten- d_e), og geometrien til forbindelsen. Zarnani m.fl. nevner at kapasiteten til skjærplanene ikke kan overstige strekkkapasiteten til den belastede motstandsblokken. Hvis skjærplanene får en større belastning enn det tilstøtende motstandsblokken kan motstå, vil blokken som rives ut av treet være like bred eller dyp som tre-

elementet. Dette resulterer i tre forskjellige sviktmoduler (a, b og c i Figur 3.41), og betyr at belastningskapasiteten til forbindelsen er lik minimumsmotstanden til de tilsvarende tre bruddmodulene [81].

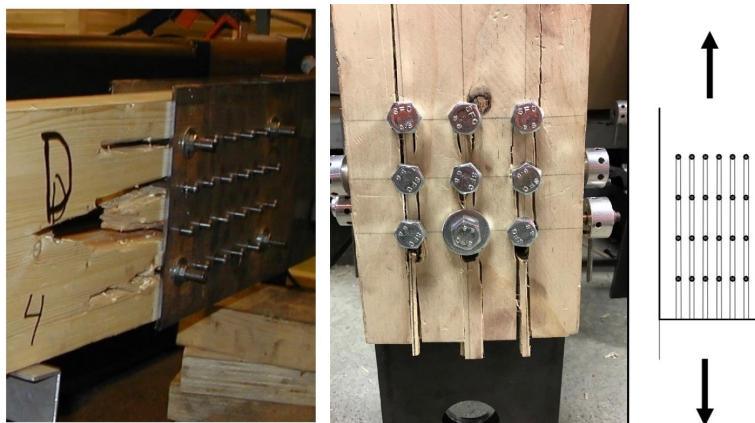


Figur 3.42: Sammenligning av lineært elastisk fjærsystem og brudd i tre-elementet [77]

I EK5 kontrolleres disse bruddformene i tillegg A [22]. Bell påpeker imidlertid at disse kontrollene kun er informative, og at det er rapportert flere skadesituasjoner knyttet til disse bruddformene. Derfor mener Bell at det bør gjennomføres kontroller for denne typen brudd i forbindelser der det kan forekomme [22].

Radskjærbrudd

Radskjærbrudd oppstår når skjærkapasiteten parallelt med fibrene til tre-elementet overstiges. Dette kommer av eksentristene fra den påførte belastningen eller ved utilstrekkelige avstander mellom dyblene [58, 79, 85]. Denne bruddtypen fører til at tre-fibrene rundt dyblene svikter langs raden med årringene, og skaper splitting parallelt med fiberretningen fra den øvre dybelen og ned til enden av elementet [86]. Se Figur 3.43 for illustrasjon av denne bruddtypen.



Figur 3.43: Flere eksempler på radskjærbrudd [78, 87]

I en studie utført av Sjödin m.fl. [88] ble bruken av dybler med ru og glatt overflate undersøkt. Resultatene viser at ved bruk av dybler med ru overflate, oppstår radskjærbrudd, mens dybler med glatte overflater fører til splittingsbrudd. En annen studie utført av Jensen m.fl. [89] ble det vist at i en forbindelse med ru dybel, var skjærspenningen veldig høy ved brudd, mens strekkspenning var derimot veldig liten. I forsøket med glatt dybel overflate, var både skjær- og strekkspenning like

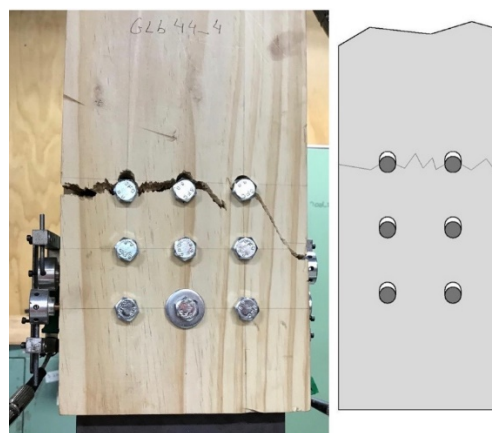
høye. Studien til Jensen m.fl. [89] og Dorn m.fl. [90] viser at rue dybler øker belastningskapasiteten til forbindelsen. Dette kommer av at dybler med glatt overfalte har mindre friksjonsmotstand mellom treet og dyblene, og kontaktsonen mellom treet og dybelen er halvparten av dybel-diameteren. Dette medfører en kile-linkende virkning av dybelen, som kan gi store strekkspenninger vinkelrett på fiberretningen, og fører til splittingsbrudd ved lav belastning. Dybler med ru overfalte har derimot både økt friksjonsmotstand og kontaktområdet, og i tillegg reduserer de sidestrekspenninger. Skjærkapasiteten til forbindelsen økes også, og ved belastningen vil skjærspenninger oppstå i en større avstand fra symmetriplanet til forbindelsen [89, 90].

Kombinasjon av alle disse effektene er grunnen for den økte bæreevnen i forbindelser med rue dybler [90]. Radskjærbruddet oppstår som følge av de høye skjærspenningene fra de rue dyblene, som resulterer i et brudd i skjærplanet som nesten er plassert tangentialt på hullet. Ved brudd, sprekker treet i skjærplanet og knuses deretter på grunn av trykk under dybelen. I studiene presiseres det at dette er en mer duktil bruddform, som øker deformasjonen før det oppstår brudd [90].

Strekbrudd

Yurrita m.fl. [87] påpeker at det ikke finnes en klar definisjon på denne bruddformen, men nevner at ifølge paragraf 6.1.2 EK5, må den dimensjonerende strekkspenningen langs fiberretningen, være mindre eller lik trematerialets strekkkapasiteten langs fiberretningen. Yurrita m.fl. presiserer også at dette er tilfellet for forbindelser.

Strekbrudd er et sprø brudd, og kan oppstå i en dybelforbindelse på grunn av at strekkkapasiteten i fiberringen er større enn tredelens nettoareal ved dybelene. Bruddet vil være vinkelrett på fiberretningen, som er vist i Figur 3.44, og det reduserte arealet påvirkes av dimensjonene og lengden til dyblene vinkelrett på fibrene til treet. Denne bruddformen er ikke inkludert for forbindelser i Ek5, og det anbefales å øke avstanden fra ubelastet kant $a_{4,c}$ og avstanden mellom dyblene på tvers av fiberretningene for å redusere risikoen for strekkbrudd [79, 87, 89]. Dette skyldes at bruddformen er påvirket av størrelsen til dyblene, og ved å øke avstandene øker mengden med tremateriale mellom dyblene. Dette medfører mer styrke og støtte for å motstå strekkraftene [79, 91].



Figur 3.44: Eksempel på strekkbrudd [87, 91].

3.5.7 Flersnittet forbindelse med stålplater

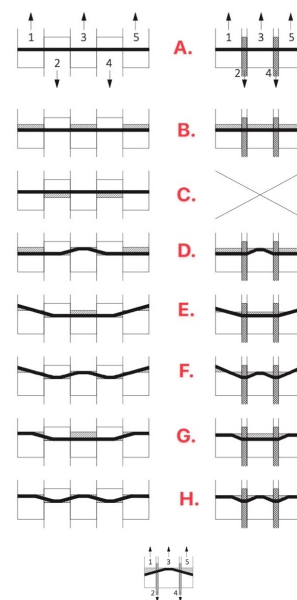
Bruk av flere stålplater i en forbindelse har vært kjent siden 1940, og brukes ofte i store trekonstruksjoner [66]. I Limtreboka [58] blir det presisert at bredden til stavene i et limtrefagverk bestemmes av nødvendig antall stålplater for å overføre kreftene, men i EK5 er det ikke direkte gitt formel for mer enn to skjærplan [66]. Det er heller ikke direkte angitt beregningsregler for innslissede stålplater med dybler i EK5, men i punktene 8.1.2 og 8.1.3 stilles det krav for flersnittet forbindelser [63, 68].

I henhold til punkt 8.1.3 (2) må bruddformene i en trekonstruksjonsforbindelse være kinematiske forenlige, siden tøyningen i det indre og ytre feltet ikke kan være forskjellige [63]. Grunnen kommer av at bruddformene i den ytre og indre delen av forbindelsen er kinematisk avhengig av hverandre [63, 79, 92]. Dermed kan bruddformer som forutsetter små forskyvinger (bruddform c, f og j/l i EK5, se Figur 3.46) ikke kombineres med de andre bruddformene [79, 92].

Bruddformer i flersnittet forbindelse

Figur 3.45 viser forskjellige bruddformene som kan oppstå for tre-mot-tre og stål-mot-tre forbindelser fra boka «Timber Engineering» [57]. Det presiseres at ved bruk av tynn stålplate kan, også bruddformen Figur 3.45-C oppstå. Grunnen er at tynn stålplate ikke gir tilstrekkelig klemeffekt, og feste-elementet kan deformeres hvis tredelene 1 og 5 i delen nederst i Figur 3.45 er tynne [57].

En dybel-forbindelsen som har to symmetrisk tykke innslissede stålplater kan ha seks mulige bruddformer, se Figur 3.45- B til H [57, 93]. Ved bruk av tre symmetrisk innslissede stålplater, vil bruddformene være nokså like bruddformene med to stålplater. Bruddformene er basert på at hullkantfastheten og bøyningen til dyblene har en plastisk stiv oppførsel [93].



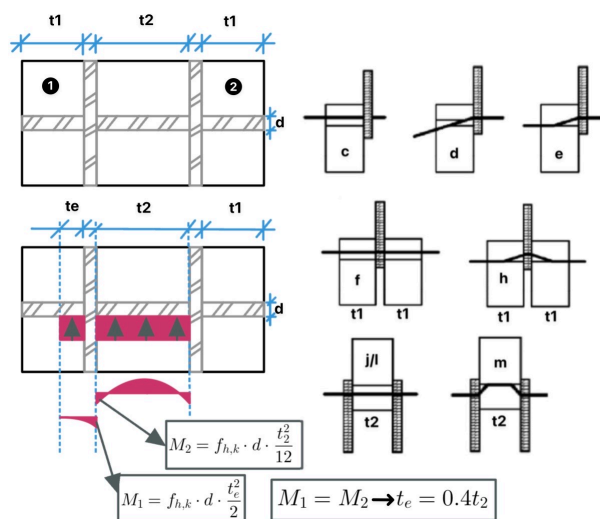
Figur 3.45: Bruddformer for tre-mot-tre og stål-mot-tre [57].

Kinematiske forenlighet i bruddformene

Hvis det innerste feltet i en flersnittet forbindelsen har bruddform, vil kinematiske avhengigheten antyde at bruddforene e, d eller e kan inntreffe i det ytrefeltet [79]. Dette er illustrert i Figur 3.46, hvor de ytre feltene er nummerert med 1 og 2. Mulige bruddformer for en flersnittet forbindelsen fra EK5 er vist til høyre i Figur 3.46. På grunn av kontinuitet over stålplaten i en flersnittet forbindelse, vil bruddformene a, b, g, og h ikke forekomme [63, 92].

I EK5 er det spesifisert at motstanden i hvert snitt av en flersnittet forbindelse bestemmes ved å betrakte at hvert snitt som en del av en rekke av dobbeltsnittende forbindelser [63]. Deretter beregnes kapasiteten ved å velge de laveste kapasitetene fra hvert snitt og summere dem. En rapport fra Statens vegvesen kontroll av fagverksbruer i tre [92], påpeker at det oppstår problemer hvis den laveste kapasiteten til forbindelsen ikke er basert på bruddformene som er kinematisk forenlige. Hvis kravet om bruddformenes forenlighet ignoreres og den minste kapasiteten fra hvert snitt velges, vil dette resultere i en veldig konservativ utforming som kan vært praktisk, men ikke fysisk mulig.

Denne tilnærmingen tar heller ikke hensyn til hvilke bruddformer som kan oppstå, noe som er nødvendig i beregning av blokkutrivning [92].



Figur 3.46: Bruddformene fra EK5 og effektiv tykkelse beregning [79, 94].

Rapporten til Statens vegvesen [92] peker på enn annen tilnærming som innebærer å ta utgangspunkt i bruddformen til den innerste delen av forbindelsen. Siden denne delen utgjør den største delen av kapasiteten og deformasjonen til forbindelsen, vil bruddformen til den ytre delen bestemmes av denne. En situasjon der den indre bruddform j/l gir minst kapasitet, vil for eksempel bruddform c i den ytre delen blir dimensjonerende, selv om bruddform d og e gir mindre kapasitet [92].

Hvis den ytterste delen til er veldig tykk, kan dette føre til en urimelig stor beregnet kapasitet [79, 92]. For å unngå dette foreslår Statens vegvesen å bruke en effektiv tykkelse, kalt t_e , istedenfor den faktiske tykkelsen t_1 , som er vist i Figur 3.46. For å beregne den effektive tykkelsen antas at dybelen i den ytterste delen å være en utkraget bjelke med innspenningsmomentet lik det plastiske momentet ($M_{y,Rk}$) [92]. Dybelen i det ytterste feltet antas å være på kjent av en jevnt fordelt hullkantrykk. Ved å anta at innspenningsmomentet til dybelen i samme stålplate er lik for både det indre og ytre feltet, kan t_e bestemmes som vist i formel 3.22 [79]:

$$t_e = \min\{0.4 * t_2 | t_1\} \tag{3.22}$$

Denne tilnærmingen kan kun antas dersom dybelen er gjennomgående. Dette er illustrert i Figur 3.46 [79, 94].

Påvirkningen av størrelsen på treet og dybler

I en studie av Sawata m.fl. [93] ble dybel-forbindelser undersøkt som inneholdt både to og tre innslissede stålplater med varierende forhold mellom tre tykkelsen og dybel-diameteren $\left(\frac{T}{d}\right)$, og forholdet mellom innerfelt og ytterfelt $\left(\frac{t_2}{t_1}\right)$. Resultatet viste at flytegrensen til en dybel-forbindelse med et $\frac{T}{d}$ -forhold på 4 var uavhengig av $\frac{t_2}{t_1}$ -forholdet. Dersom $\frac{T}{d}$ -forholdet var over 8, påvirket avstanden mellom stålplatene flytegrensen. Dette var tilfellet for både to og tre stålplater [93].

For dybel-forbindelser med to stålplater [93], økte flytegrensen etter hvert som $\frac{t_2}{t_1}$ forholdet økte, og $\frac{T}{d}$ -forholdet var 8. Den høyeste flytegrensen ble oppnådd når $\frac{t_2}{t_1}$ -forholdet var 1.0, og $\frac{T}{d}$ -forholdet økte. Resultatene for tre stålplater hadde også samme tendens [93].

For eksempel [93], når $\frac{T}{d}$ -forholdet var 12 og $\frac{t_2}{t_1}$ -forholdet 1.0, var flytegrensen 15% mindre enn når $\frac{t_2}{t_1}$ forholdet var 2.0. Når $\frac{T}{d}$ forholdet økte til 20 og $\frac{t_2}{t_1}$ var 1.0, ble flytegrensen derimot 7% større enn ved et $\frac{t_2}{t_1}$ forhold på 2.0. Sawata m.fl. konkluderer med at flytegrensen til dybel-forbindelsen blir påvirket av både $\frac{T}{d}$ og $\frac{t_2}{t_1}$ forholdet, og at skjærstyrken til dybel-forbindelsen er tilstrekkelig når hensiktsmessige $\frac{T}{d}$ -forhold og $\frac{t_2}{t_1}$ -forhold brukes [93].

Antall innslissende stålplater og tykkelsen til tre-elementet påvirker også skjærstyrken betydelig [93]. Resultatet til Sawata m.fl. viser at når $\frac{T}{d}$ -forholdet var 5, 10, 15 og 20 i en dybel-forbindelse med to stålplater, var flytegrensen 40%, 60%, 90% og 100% større en flytegrensen til en dybel-forbindelse med en stålplate. For tre stålplater var flytegrensen 40%, 90%, 150% og 190% større enn for en stålplate [93].

Sawata m.fl. [93] påpeker også at når $\frac{T}{d}$ -forholdet er henholdsvis over 14 eller 18 for dybel-forbindelser med to eller tre stålplater, kan flytegrensen til disse beregnes ved å multiplisere flytegrensen til en forbindelse med en enkelt stålplate med antall stålplater i forbindelsen. Dette gir en tilnærmet verdi for flytegrensen til forbindelser med både to og tre stålplater [93].

3.6 Optimalisering av fagverk

I dette underkapittelet presenteres relevant teori som er nyttig for å forstå optimalisering av fagverk. Dette gir en grundigere oversikt og forståelse av teamet som vil være nyttig å ha kunnskap om.

3.6.1 Optimalisering

Optimalisering kan defineres som søken etter den beste løsningen til et problemet under gitte forhold [95]. Denne definisjonen inkluderer alle tekniske designproblemer, og kan derfor brukes innen flere felt i designteknikk. Innen konstruksjonsteknikk handler optimalisering om å finne det beste designet for et gitt system med hensyn til kapasitet, sikkerhet, levetid, miljøpåvirkning, kostnad og materialreduksjon [95-97]. De fleste studier påpeker at designoptimalisering brukes vanligvis for å redusere materialbruken [95, 96, 98, 99]. Flere forskjellige løsninger oppnås ved å vurdere de

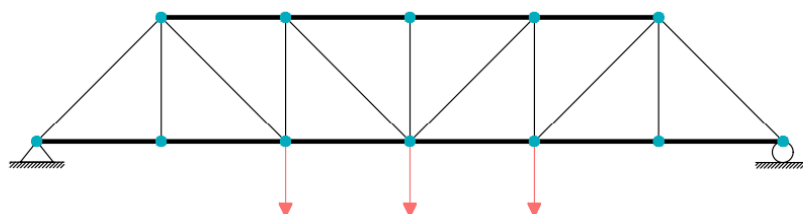
arkitektoniske kriteriene, men for å finne den mest optimale løsningen fra et uendelig utvalg, kreves både tid og iterative beregninger. Digitale verktøy brukes derfor med matematiske og beregningsmessige optimaliseringsalgoritmer for å raske finne optimale løsninger [100].

Ifølge Aydoğdu m.fl. [95] finnes det flere matematiske optimaliseringstyper, og metaheuristiske metoder skiller seg ut fra vanlige optimaliseringstyper. Aydoğdu m.fl. begrunner dette med at egenskapene til metaheuristiske metoder er spesielt fordelaktige for å løse komplekse design problemer. Ettersom iterative søkeprosesser med tilfeldig algoritmer brukes for å gradvis forbedre løsningen. I tillegg brukes det ikke gradientbasert beregninger, siden dette ikke er optimalt for å løse komplekse problemer. Dette gjør at metaheuristiske metoder ikke trenger en høy beregningsinnsats for å finne optimale løsninger. Aydoğdu m.fl. presiserer at metaheuristiske metoder ikke nødvendigvis finner den beste løsningen, men gir likevel gode løsninger for konstruksjonen [95].

Aydoğdu m.fl. [100] påpeker at det er tre kategorier som vurderes ved optimalisering av fagverkt. Dette er optimalisering av størrelsen til tverrsnittet, utforming av strukturen og topologi.

Størrelsesoptimalisering

For å optimalisere et fagverk innebærer dette å gjøre flere vurderinger. Den første er størrelsesoptimalisering, som handler om å finne den optimale størrelsen og velge materialtypen til elementene [100, 101]. I et limtre-fagverk bestemmes tverrsnittstørrelsene av kreftene, spenningene og nedbøyningen som påvirker konstruksjonen [58]. Som nevnt i delkapittel 3.5.1 bestemmes bæreevnen til konstruksjonen av kapasiteten til forbindelsene. Det er derfor viktig at optimalseingen tar hensyn til både forbindelsene og tverrsnittene, siden endring av én faktor kan påvirke den andre. I forbindelse må antall dybler, avstandene mellom dem og ende- og kantavstanden optimaliseres i henhold til de påførte kreftene. Dette vil bestemme størrelsen på stålplatene og dimensjonen til limtrettverrsnittet. Å finne den optimale konstruksjonen, er en helhetlig prosess som må ta hensyn til både forbindelsen og tverrsnittene [102]. I Figur 3.47 illustreres størrelsesoptimalisering av et stål fagverk. I denne figuren har over- og undergurten større tverrsnitt enn stavene, ettersom gurtene bærer det meste av de påførte lastene [58].



Figur 3.47: Størrelsesoptimalisering av fagverk [103].

Formoptimalisering

I denne kategorien blir geometrien til fagverket vurdert. Koordinatene til nodene i fagverket blir brukt som designvariabler, og formålet med optimaliseringen er å forbedre ytelsen til strukturen ved å modifisere grensene. [104, 105]. Generelt er høyden, lengden og vinkelen til stavene i et fagverk designvariablene [58, 101, 103].

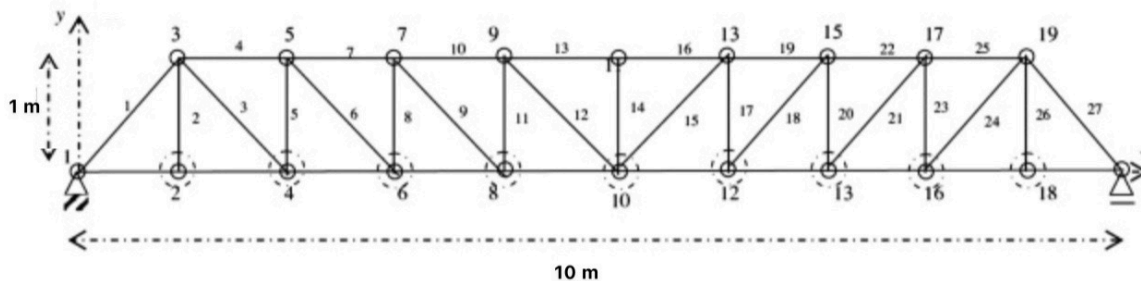
I studien til Mitropoulou m.fl. [105] beskrives metodikken innen formoptimalisering. Det første trinnet er å starte med en grov modell av strukturen. Deretter er det viktig å bestemme optimaliseringsformålet, og hvilke av designvariablene som skal brukes for å oppnå dette målet [105].

I trinn to blir strukturen skapt ved hjelp av de bestemte designvariablene og av formålet med optimaliseringen [105]. Deretter må det utføres en strukturell analyse av elementene, forskyvningen og spenningene i strukturen. Mitropoulou m.fl. påpeker at en adaptivitetsanalyse bør inkluderes i dette stadiet for å øke nøyaktigheten til analysene, ettersom dette vil tilpasse mash-størrelsen til områder som krever høyere oppløsning [105].

Hvis det brukes en gradientbasert optimalisering, handler trinn tre om å utføre en sensitivitetsanalyse av begrensningene og optimaliseringsfunksjonen [105]. Dette utføres ved å gjøre små endringer i designvariablene, og deretter bruke den endelige differansen mellom resultatet. Mitropoulou m.fl. påpeker at en semianalytisk metode kan også brukes [105].

I de fjerde og siste trinnet er formoptimaliseringen ferdig [105]. Designvariablene har blitt optimalisert og strukturen til fagverket ferdig definert. Hvis kriteriene til optimaliseringsalgoritmene er oppfylt, er den optimale løsningen funnet og prosessen er dermed ferdig. Hvis kriteriene ikke er oppfylt, må geometrien korrigeres og prosessen må starte på nytt fra trinn to [105].

I Figur 3.48 illustreres et fritt opplagt stål-fagverk som er blitt formoptimalisert ved hjelp av fire forskjellige optimaliseringsmetoder.



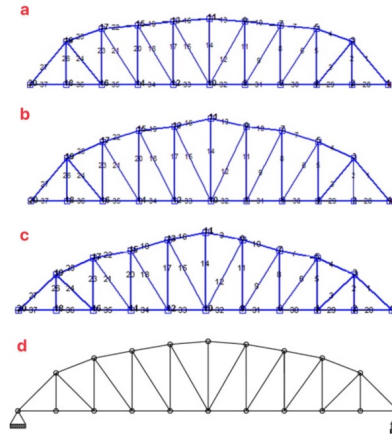
Figur 3.48: Fritt opplagt fagverk [104]

Den første metoden er kalt «Evolutionary node shift method», og ble utført av Wang m.fl. [106] i 2004. Denne metoden optimaliserer nodens koordinater og elementets tverrsnittareal i en enhetlig tilnærming for å redusere vekten til fagverket. Metoden bruker sensitivitetsanalyse for å identifisere og optimalisere de mest effektive variablene systematisk. Videre ble den andre metoden, «Niche Hybrid Genetic Algorithm» (NHGA), brukt av Lyngyun bl.a [107] i 2004. Denne algoritmen bruker genetiske algoritmer (GA) for å optimalisere løsninger. Formålet til NHGA er å forbedre utnyttelseskapasiteten til GA, samtidig som den hindrer tidlige konvergens av strukturen.

Den tredje metoden bruker «particle swarm optimization» (PSO) algoritmen, som ble brukt av Gomes [108] i 2010. PSO- algoritmen er en metaheuristisk optimaliseringsmetode, som er inspirert av flokk av fugler som beveger seg i en gruppe, hvor en fugl leder vei og søker etter mat, mens de andre fuglene deler hver sine opplevelser for å hjelpe flokken med å utføre den beste jakten. Målet

med denne metoden er å finne den globalt optimale løsningen for et bredt spekter av vilkårlige problemer [108, 109]. Til slutt ble den fjerde metoden «Ray optimaization» (RO), utført av Kaveh m.fl. [104] i 2013. RO-modellen modellerer hver komponent som en lysstråle som beveger seg i søkerommet for å finne den globale optimale løsningen.

Resultatene fra de forskjellige metodene for formoptimalisering er illustrert i Figur 3.49 [104]. I Figur 3.49-a vises resultatet fra Wang m.fl., hvor vekten til fagverket ble redusert til 366,5 kg. I Figur 3.49-b vises resultatet fra Lyngyun m.fl., hvor vekten ble redusert til 368,84 kg. I Figur 3.49-c vises resultatet fra Gomes, som oppnådde den høyeste vekten på 377,2 kg sammenlignet med de andre resultatene. I Figur 3.49-d vises resultatet fra Kaveh m.fl., som oppnådde den laveste vekten på 364,04 kg [104].



Figur 3.49: Formoptimaliserings resultatene av fagverket [104, 108].

Topologioptimalisering

Topologioptimalisering (TO) defineres som fordeling av materialer innenfor et bestemt område, uten at funksjonen til objektet reduseres [110]. Noh m.fl. påpeker at dette betyr at hvert rutenettpunkt vil være en designparameter som optimaliserer geometrien til objektet på en piksellignet måte. For at designet skal oppdateres over flere iterasjoner, presiserer Noh m.fl. at en gradient må beregnes basert på design parameterne. Videre påpeker Noh m.fl. at TO ofte er koblet til beregningseffektive «adjoint method», som beskrives som en effektiv måte å beregne gradienter for å oppnå optimal kontroll over geometrien [110, 111].

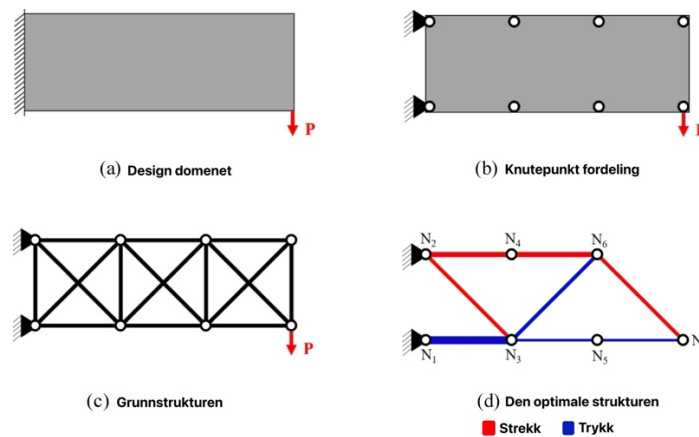
TO bruker dermed matematiske analyser til å optimalisere materialinnholdet innenfor et avgrenset området [8]. Innen konstruksjonsteknikk betraktes denne prosessen som å enten fjerne materialer som ikke har noen konstruktiv effekt på strukturen, eller ved å plassere større mengde av material i svært belastede områder [8].

Ifølge Mam m.fl. [112] finnes det to teknikk klasser innen TO som velges basert på strukturtypen. For diskrete strukturer handler optimalisering om å bestemme det optimale antallet, størrelsen, posisjonen, og forbindelsen av elementene. For kontinuumstrukturer er målet å finne den optimale fordelingen av materialet innen et bestemt område [112].

Et vanlig optimaliseringsproblemet for et fast innspent fagverk, er å finne den optimale plasseringen av stavene i strukturen [113]. Dette problemet involverer vanligvis en bestemt mengde med knutepunkter som er sammenkoblet med mulige elementer, og danner dermed en grunnstruktur. Formålet med optimaliseringen er ofte å redusere massen, volumet, eller stivheten til strukturen, med begrensninger som sikrer at hvert knutepunkt er i statisk likevekt og at spenningene til stavene er innenfor de forhåndsdefinerte kravene. Under optimaliseringsprosessen bestemmes det gunstige

tværsnitts-arealet til alle elementene i strukturen. Arealet til de fleste elementene vil være null og blir dermed fjernet. De resterende elementene som har areal, blir den endelige topologien [113].

Feng m.fl. [113] viser en fire-trinns metode for optimalisering av en grunnleggende fagverkstrukturen, som er illustrert i Figur 3.50. Det første trinnet er å definere utformingen av domenet fra den påførte lasten og festet, som er vist i Figur 3.50-a. Det andre trinnet handler om å jevnt fordele knutepunktene for å adskille design domenet, som vist i Figur 3.50-b. I det tredje trinnet kobles knutepunktene sammen av stenger som danner grunnstrukturen til fagverket, som vist i Figur 3.50-c. I det fjerde trinnet brukes optimaliseringsalgoritmer for å fjerne unødvendige stenger og generere den optimale strukturen for fagverket, som vist i Figur 3.50-d [113].



Figur 3.50: Topologioptimalisering av fagverk [113]

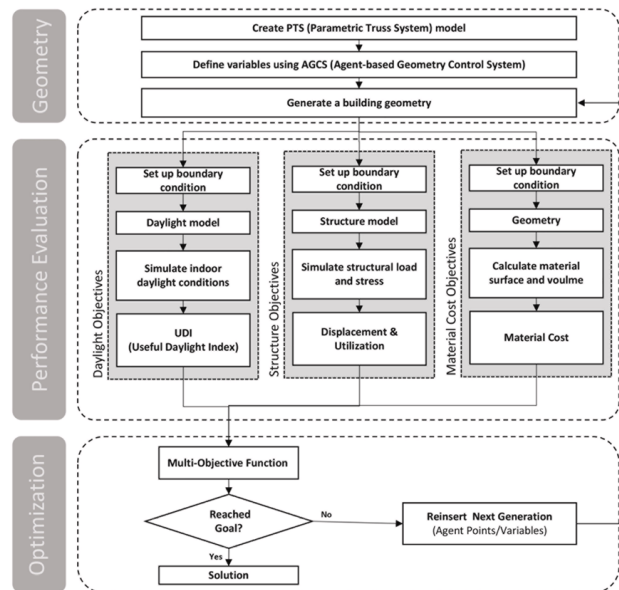
Mam m.fl. [112] understreker at stivheten i limtreforbindelser vil ha en direkte innvirkning på den optimale topologien til strukturen. I undersøkelsen til Mam m.fl. viste resultatet at stive forbindelser fører til økt volum av limtre elementene ved optimalisering av høye limtre bygninger. I tillegg økte antall forbindelser i bygget når forbindele var moment stive, sammenlignet med når de var leddet [112].

3.6.2 Multiobjektiv optimalisering

Optimalisering er viktig innenfor flere fagfelt, men dette kan skape problemer på grunn av motstridende optimaliseringsformål [114]. Det er svært sjeldent av det undersøkes etter en optimalisering som tar hensyn til alle fagområdene. Fagfeltet som undersøker optimaliseringen av energibruk i bygninger, tar sjeldent hensyn til bygningsstrukturen, samtidig vurderes sjeldent energi og dagslys ved optimalisering av strukturen. Yi m.fl. påpeker at en løsning som er optimalt i ett fagfelt, kan være en problematisk i et annet [114].

Yi m.fl. [114] presiser at den teknologiske utviklingen innen beregning, analyse og simulering kan løse dette problemet. I studien til Yi m.fl. ble Grasshopper benyttet for å optimalisere et fagverk innenfor tre områder; dagslys, kapasitet og kostand. Studien presenterer en metode som integrerer flere kriterier for å optimalisere på tvers av fagfelt. Figur 3.51 viser metodikken fra studiet [114].

Metoden baserer seg på en «Parametric Truss System» (PTS) modell, der geometriske variabler blir generert i et «Agent-basert Geometry Control System»



Figur 3.51: Metodikk for multiobjektiv optimalisering [114].

(AGCS) [114]. Geometrien blir så videresendt til neste trinn for å oppdatere systemet til fagverksstrukturen. Et simuleringsverktøy blir brukt til å beregne kapasiteten, og resultatene blir evaluert for å avgjøre om de oppfyller formålet. Hvis ikke, blir en ny geometri generert i AGCS-delen, og prosessen fortsetter i en loop til formålet er oppfylt [114].

Yi m.fl. [114] påpeker at når en kompleks struktur skal optimaliseres, økes antall geometriske variabler som kontrollerer strukturen. Dette gjør det vanskeligere å finne den optimale løsningen. For å løse dette problemet, brukte Yi m.fl. både et hierarkisk og et syntaksbasert system for å kontrollere de underordnede objektene. Dette reduserte antall geometriske variabler, siden et «agent»-punkt kontrollerte posisjonen til de underordnede objektene, i motsetning til at flere variabler kontrollerer dem [114].

3.7 Tidligere litteratur med bruk av Grasshopper

I dette underkapittelet er det innhentet fem fagartikler om bruk av visuell programmering og parametrisk design av fagverk.

3.7.1 Begrensinger ved parametrisk design

I fagartikkelen «Parametric Design- A Drive Towards a Sustainable Future» av Edmonds m.fl. [8], diskuteres begrensninger ved parametrisk design. I dette underkapittelet presenteres disse begrensningene.

Det første punktet som nevnes, er at læringskurven kan være bratt for nye brukere av parametrisk design [8]. Det kan være vanskelig å lære seg alle de ulike plug-ins verktøyene og metodene som er tilgjengelige. Det presiseres at selv erfarne brukere som er kjent med tradisjonelle 2D- og 3D-modelleringsverktøy, vil trenge tid for å lære seg oppbygningen av modeller med parametrisk design. Dette kommer av at det kreves en annen forståelse av designet til modellen, siden designet skapes av gjensidige avhengigheter og sammenhengen mellom alle de ulike variablene. Edmonds m.fl. nevner at den enkleste måten å lære seg parametrisk design er å se igjennom om kurs. Deretter bør

brukeren begynne med enkle modeller, og deretter gradvis øke kompleksiteten og tilleggsmulighetene til modellen [8].

Det andre punktet som nevnes er risikoen for «Black box»- design [8]. Dette oppstår når brukeren har alt for mye tillit til datamaskinens resultat og utfører ikke egne undersøkelser av resultatet. Dette kan føre til mulige feil i designet, som kan både være kostbare og tidskrevende å fikse. Det presiseres at denne risikoen finnes i alle programvarer, men kan spesielt oppstå i parametriske programvarer. Fordi nedlastede tilleggsverktøy kan ha inkompatible funksjoner i det parametriske designsystemet, noe som kan føre til uventede feil, og dermed påvirke resultatet [8].

For å redusere denne risikoen, presiserer Edmonds m.fl. [8] at det er viktig å bruke interiørmessig skjønn og kritisk tenkning i alle aspekter av et gitt prosjekt. Hver løsning som er utviklet parametriske bør undersøkes, og det bør være enkelt å identifisere kreftene som påvirker konstruksjonene. Videre anbefales det å utføre håndberegninger for å bekrefte kreftene og konseptene til den optimalisert geometri. I tillegg bør geometrien fra optimaliserings analysen, undersøkes i mer avanserte programvarer som er mer egnet til utføring av beregningsanalyser [8].

Det tredje punktet som Edmonds m.fl. [8] nevner, er utfordringen med å finne det ønskede resultatet når modellen inneholder mange variabler. Ettersom det kan oppstå en endeløs iterasjon i det parametriske designet programmet. For å unngå dette er det viktig å bestemme hva som er viktig for prosjektet. For eksempel kan dette være å sette søkelys på kun et geometrisk formål eller å redusere materialmengden mest mulig. Ved å bestemme begrensinger for resultatet vil det bli enklere å fjerne løsninger som ikke oppfyller disse begrensingene. Dette gjøre det enklere å finne den mest optimale løsningen for prosjektet [8].

I det fjerde punktet nevnes byggbarheten av strukturen som er generert i et parametriske design program [8]. Det poengteres at selv om strukturen er analysert og designet i henhold til lastene, betyr ikke dette nødvendigvis at det endelige designet kan bygges på en effektiv og sikker måte. Parametriske design kan gi uendelig mange geometriske alternativer, men den programmerte og optimale strukturen er ikke alltid den mest kostnadseffektive og bærekraftige løsningen. Det kan derimot føre til motstridene effekter, som Edmonds m.fl. påpeker, kan dette medføre mer arbeid til å stabilisere strukturen under byggefasen. I tillegg kan en komplekst geometriske løsning oppfylle det estetiske kravet, men samtidig øke arbeidstiden for å bygge denne strukturen, og dermed øke kostandene [8].

For å unngå å laget et design som ikke er byggbart, presiserer Edmonds m.fl. [8] at er det viktig å vurdere konstruksjonsprinsipper og begrensinger tidlig i prosjektet. Dette betyr å legge inn begrensingene i designet etter hvert som det utvikler seg, noe som kan sikre at det endelige designet er byggbart og kan effektivt bygges [8].

I det femte punktet, nevner valget av de riktige parameterne som skal variere designet [8]. Edmonds m.fl. påpeker at det er viktig å stille de riktige spørsmålene før iterasjonsprosessen begynner. Det presiseres at dette kan være utfordrende, siden svarene må tilpasses hvert prosjekt. For eksempel er det viktig å forstå prosjektbegrensingene, konstruksjonsmetodene, de geotekniske risikoene, og budsjettbegrensingene. I tillegg må varierende parametere vurderes i henhold til hvordan disse

påvirker designet. Det anbefales å gjennomføre samarbeidsvurderinger med flere fagfelt før modellen utvikles. Dette gjør det mulig å identifisere de viktige variablene og deretter justeres dem slik at de skaper mest mulig verdi for strukturen [8].

I det sjette punktet påpekes hvordan parametrisk modellering og design kan være en potensiell ulempe for den kreative designprosessen [8]. I både konseptuelle og optimaliseringsdesign vil parametrisk design produsere iterasjoner og gi designere flere alternativer å velge mellom. Dette kan potensielt begrense den historiske tenkemåten som har utviklet noen av historiens beste designeksempler. Både innen ingeniørfaget og arkitektur brukes grunnleggende designprinsipper i alle prosjekter. Edmonds m.fl. presiser derfor, at korrekt bruk av parametrisk modellering og design kan forbedre den kreative designprosessen. Det genererte designet bør alltid testes mot grunnleggende prinsipper, og kritisk vurderes for å utfordre det. Dette vil sikre at det endelige resultatet er utformet best mulig for prosjektet, i motsetning til å velge et av alternativene fra design optimaliseringen [8].

I det syvende punktet påpekes det at parametrisk design ikke er et verktøy som alltid vil løse eventuelle designproblemer [8]. Selv om det ofte betraktes som en måte å løse problemer i designprosessen, vil dette ikke alltid være tilfellet. Enkelte komplekse design kan være av kvalitative art, og i slike tilfeller vil ikke programmeringen av et parametrisk designscript nødvendigvis gi den optimale løsningen. Det kan derimot øke kompleksiteten til designet og dermed øke kostander i prosjektet. Edmonds m.fl. påpeker at igjen er det avgjørende å ha erfaring som kan vurdere om parametrisk design er det riktige verktøyet for å løse problemet. Dette gjør at bruken kan vurdere om det er en mer kvalitativ designprosess, som heller benyttes [8].

I det åttende og siste punktet, nevner risikoen med å i bruk programvarer for tidlig i designprosessen [8]. Det presiseres at selv i den tradisjonelle designprosessen kan tidlig modellering føre til mislykkede resultater. Dette oppstår dersom forarbeidet og de grunnleggende prinsippene ikke er fullstendig gjennomtenkt. Edmonds m.fl. påpeker at parametrisk design kan forsterke denne risikoen. Hvis begrensningene til et prosjekt ikke er fullstendig forstått, eller at de valgte parameterne for optimalisering ikke er godt nok gjennomtenkt, kan det parametriske designet ikke gi det ønskelige resultatet. Det er derfor viktig å ha grundig kunnskap om konseptet til designet og de grunnleggende prinsippene før programvarene brukes. I tillegg må brukeren ha en fullstendig forståelse av de varierende faktorer som vil påvirke prosjektet før det parametriske designet starter [8].

3.7.2 Design av stålkonstruksjoner med gjenbrukbare elementer

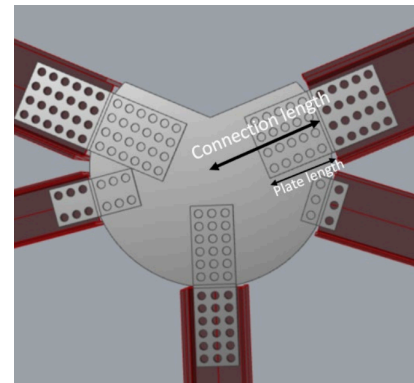
I masteroppgaven til Boer [5] ble Grasshopper brukt til visuell programmering av bolteforbindelser med stålplater. Forbindelsene ble utformet i henhold til beregningskrav. Selv om studien bruker stål og designer en forbindelse som ikke er lik en dybel-forbindelse i et limtre-fagverk, kan framgangsmåten likevel hjelpe oss med å visuelt programmere forbindelser for et limtre-fagverk. I dette kapitlet presenteres relevant informasjon fra studien til Boer [5].

Studien omhandler bruk av parametriske verktøy for design av stål strukturer for økt gjenbrukbarhet og påvirkningen forbindelsen har på gjenbruken av strukturen [5]. Boer har utviklet et designverktøy som visuelt designer stålforbindelser, og som skal være gjøre det enklere å demontere

konstruksjonen, uten å skade den-. Forbindelsen har en sirkulær utforming med standard bolteavstander og er designet i henhold til kravene for stålfester i NS-EN 1993 1-8 Eurokode 3 [115] (EK3) for å sikre tilstrekkelig kapasitet. Utformingen er tilpasset I-profiler og er designet for overføring av aksialbelastning.

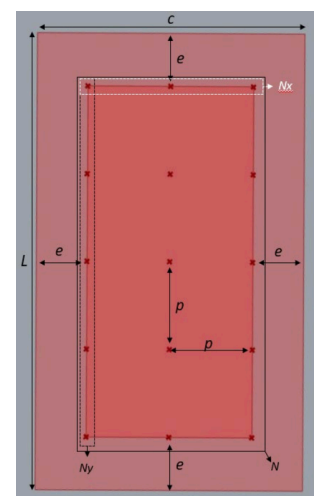
For å teste verktøyet ble et stål-fagverk valgt som case [5]. Studien brukte plugg-in verktøyet Karamba3D for strukturell analyse av strukturen. Karamba3D beregner kapasiteten til hvert element av strukturen i henhold til kravene i Eurokode 3 (NS-EN 1993 1-8). Boer brukte komponenten tverrsnitts-optimalisering for å finne den optimale dimensjonen til elementene under gitte betingelser. Kriterier som ble brukt var minimumshøyden eller arealet til I-profiler, og Karamba3D søkte etter den strukturen som hadde minst vekt i henhold til utnyttelsene av moment-, skjær-, normalkraft og de ulike knekkingstyper. Etter Karamba3D hadde optimalisert strukturen, ble forbindelsene designet basert på dataene til Karamba3D [5].

Boer [5] påpeker at forbindelseslengden («*Connction length*») i Figur 3.52) var en viktigfaktor for å ta hensyn til størrelsen på forbindelsen. Denne størrelsen ble anslått ut ifra vinkler, profilstørrelser og krefter i forbindelsen. Både vinkel- og profilstørrelsen bestemte avstanden fra midtpunktet av sirkelen til midten av de rektangulære stålplatene. Dette er vist i Figur 3.52. Antall bolter og plasseringen av dem ble bestemt basert av kreftene i forbindelsen og profilstørrelsene. Lengden til stålplatene ble bestemt som et resultat av dette, og deretter bestemmes lengden til forbindelsen av stålplatene [5].



Figur 3.52: Stålforbindingen designet av Boer [5]

Boer [5] programmerte et skript i Grasshopper som automatisk designet både den sirkulære og de rektangulære stålplatene i forbindelsene, basert på de påførte kreftene. De rektangulære stålplatene er designet for å overføre kreftene til den sirkulære stålplaten. Skriptet bestemte det nødvendige antallet av bolter ved å dele den totale kraften i forbindelsen på kapasiteten til en bolt. Deretter ble boltene fordelt i x- og y-retning, hvor bredden til den rektangulære stålplaten ble bergenes ut ifra høyden mellom flesene til I-profilet. Antall bolter i x-retningen ble beregnet basert på avstandskravene fra EK3, og i y-retningen ble antall bolter beregnet ved å dele det totale antallet av bolter på boltene i x-retningen. Lengden til stålplaten blir deretter beregnet ut ifra avstandskravene og antall nødvendige bolter i y-retningen. I Figur 3.53 illustreres de definerte lengdene og hva som er definert som x- og y-retning. Dermed ble arealet til den rektangulære stålplaten beregnet og videre blir den speilet, slik at en del er i I-profilet og den andre delen er i den sirkulære stålplaten vist i Figur 3.52 [5].



Figur 3.53: Oversikt over parametere av rektangulære stålplaten [5].

Platetykkelsen er avhending av boltevalget, men i de elementene hvor spenningene var store, bestemmes tykkelsen av formler fra EK3 [5]. Det samme gjelder for tykkelsen til den sirkulære stålplaten, men lengden bestemmes på en annen måte. Boer har utviklet en beregning som tar hensyn til lengden til de rektangulære stålplatene og minimumsavstandene som skal forhindre kryssing av elementene i forbindelsen [5].

Boer [5] påpeker at den strukturelle kapasiteten til forbindelsen var tilstrekkelig, men resultatet viste at mindre profiler var mer utsatt for relativ store strekkrefter, noe som potensielt kan føre til brudd i nettoseksjonen av forbindelsesplatene. Boer designet forbindelsene for skjær- og bærekapasitet i kombinasjon med kravet om at boltene skulle svikte av skjærkreftene. De andre feilmekanismene ble vurdert, men ble ikke antatt som et alvorlige problem som kunne føre til brudd i forbindelsen. Boer konkluderte med at det ville vært fornuftig å utføre en Finite Element Analyse av forbindelsen for å kunne forbedre designet [5].

3.7.3 Parametrisk design og analyse av trekonstruksjoner ved bruk av Beaver

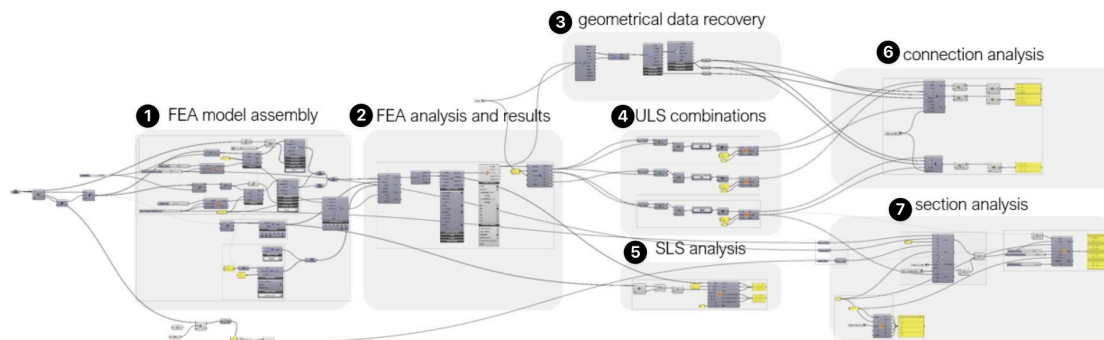
I studien fra Pini m.fl. [6] presenteres utviklingen av plug-in verktøyet Beaver for Grasshopper og Rhino3D. Beaver gjør det mulig å parametrisk analysere og designe trekonstruksjoner i henhold til EK5, ved hjelp av den strukturelle dataen fra Karamba3D. Beaver kan undersøke bestemte trematerialer i ULS og SLS for analyse av elementene og forbindelsene i strukturen. Det spesifiseres at Beaver sammenkobler designprosessen fra arkitektonisk utforming og strukturell detaljering i en programvare [6]. I dette underkapittelet presenteres arbeidsflyten mellom Beaver og Karamba3D, beregningskomponentene og resultatene fra et av de presenterte case-studiene i studien.

Arbeidsflyt og beregninger med Beaver komponenter

I Figur 3.54 illustreres arbeidsflyten mellom Beaver og Karamba3D fra studien til Pini m.fl. [6]. Prosessen starter med oppbyggingen av selve modellstrukturen (se Figur 3.54-1), etterfulgt av en FEA-analyse basert på materialet, tverrsnittet og lastene som utføres av Karamba3D. Resultatet av denne prosessen overføres deretter til Beaver-komponentene som utfører den strukturelle analyse av trekonstruksjonen. Geometridataene gjenoprettes (se Figur 3.54-3) og de indre kreftene i nodene til strukturen genereres og sendes videre til forbindelsesberegningen (se Figur 3.54-6). I ULS kombinasjonen genereres de relevante lastkombinasjonene basert på EK0. Inputene inkluderer de karakteristiske lastverdiene, klimaklassen og de definerte last-typene. Outputene er ULS kombinasjonene, K_{mod} verdiene og informasjon om de forskjellige lastkombinasjonene med lastfaktorer. I SLS-analysen analyseres alle mulige kombinasjoner av øyeblikkelige og langtidslast nedbøyninger.

I seksjonsanalysen (Figur 3.54-7) utføres utnyttelsesberegningene for strekk, trykk, torsjon, skjær- og bøyekrefter for tre-elementene [6]. Inputene er lastkombinasjonene fra ULS-analysen, dimensjonen til tverrsnittet og materialdata. Outputene er utnyttelsen både i y- og z-retning for alle nødvendige beregninger i henhold til EK5 [6].

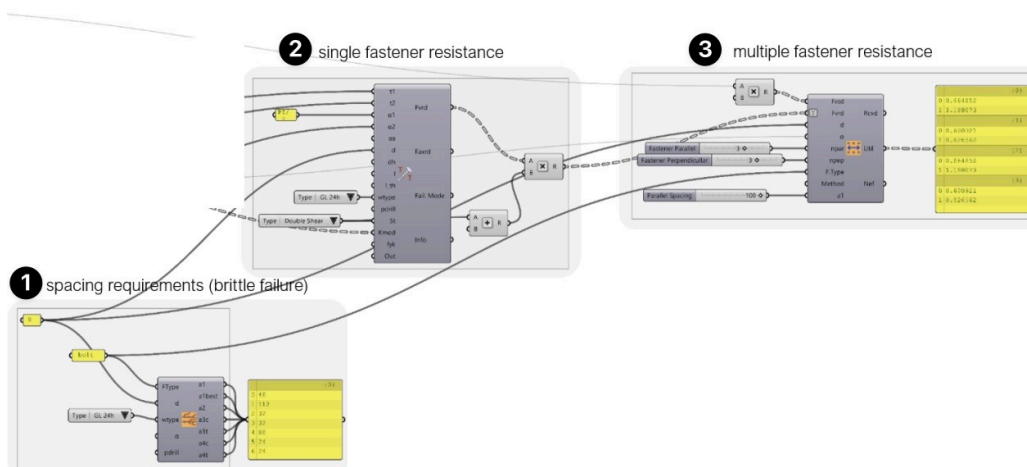
I forbindelses-gruppen vist i Figur 3.54-6, blir kapasiteten til bolter, dybler, skruer eller spikre vurdert etter Johansens feilmekanismer for tre-mot-tre og stål-mot-tre forbindelser [6].



Figur 3.54: Arbeidsflyten for parametrisk analyse med Bever og Karamba3D [6].

Figur 3.55 viser kolbingen mellom de forskjellige komponentene som utfører de nødvendige analysene for trekonstruksjonsforbindelser. Pini m.fl. forklarer at sprø brudd kan unngås ved at tilfredsstille minimumskravet fra EK5. Dette undersøkes i komponenten vist i Figur 3.55-1. For å undersøke den duktile bruddformen, må mostanden til ett enkelt festemiddel først vurderes (Figur 3.55-2). Deretter vurderes den totale motstanden til festene (Figur 3.55-3) med avstandene som gir den effektive kapasiteten til hele forbindelsen. Pini m.fl. nevner at denne prosessen kan være utfordrende, siden forbindelsen er avhengig av de geometriske begrensingene til strukturen. Det er også flere mulige konfigurasjoner av forbindelsen og ulike festemidler som skaper utfordringer når det gjelder å finne den optimale løsningen [6].

I Figur 3.55-3 vises komponenten som beregner kapasiteten for ett enkelt festemiddel. Denne komponenten bruker input-parametere, som festemiddeldiameter, stålmotstanden, lengde til festemiddel, type trematerial, antall skjærplan og vinkelen på kraften i henhold til fiberretningen [6]. Det presiseres at det kun kan brukes en eller to skjærplan, og at resultatene bestemmes av den minste verdien av de mulige bruddformene som kan oppstå. Outputen fra denne komponenten er festemidlets skjær- og aksialmotstand, som er basert på hullkantfastheten og det karakteristiske flytemomentet til forbindeeren. Disse beregnes av inputene og det valgte festemiddelet. Hvis motstandsverdien for flere skjærplan skal beregnes, må dette korrigeres i henhold til kapasitetsverdien for et skjærplan [6].

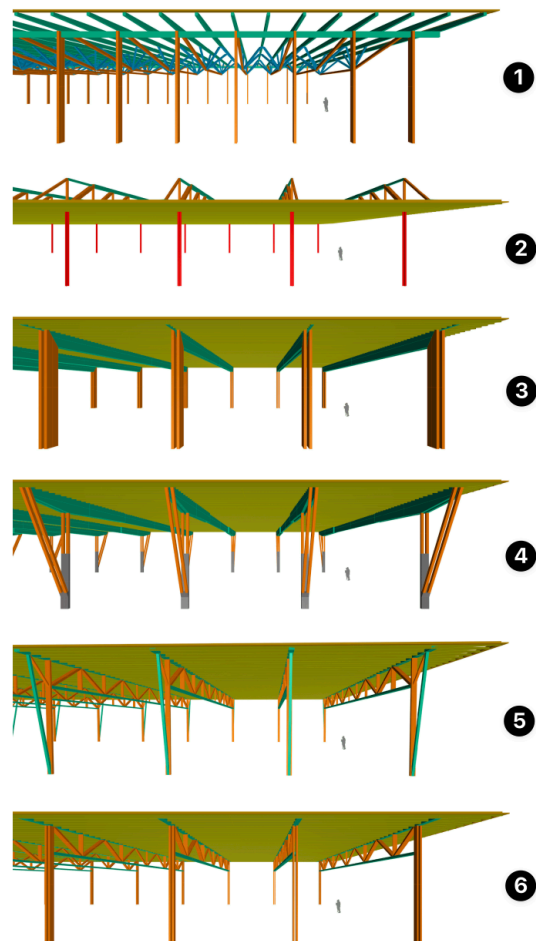


Figur 3.55: Arbeidsflyten for forbindelsesanalyse [6]

Pini m.fl. [6] nevner to mulige metoder for beregning av kapasiteten til en forbindelse med flere festemidler. Den første metoden er å beregne den samlede mostanden for hele forbindelsen, mens den andre metoden er å beregne kapasiteten til den mest påkjente forbindeeren. Pini m.fl. påpeker at den andre metoden er fordelaktig i momentstive forbindelser. Siden lasten og retningen på kreftene vil variere på festene, og siden motstanden til hvert individuelle festemiddel baseres på det effektive og det totale antallet av festemidler [6].

Optimalisering av limtrefagverk med Beaver

I dette caset skulle en hestearena i Sao Paulo, bygges et 40 meter langt flatt tak. ITA Construtora var en bedrift med over 40 års erfaring med trekonstruksjoner, og var involvert i prosjektet. Ved å kombinere denne erfaringen med en parametriske prosess, kunne flere strukturelle konsepter analyseres på kort tid. De analyserte konseptene vises i Figur 3.56. Hele modellen ble parametriske utviklet i Grasshopper, og avstanden mellom fagverkene, fagverkshøyden, bjelketverrsnittene og utformingen på fagverket ble valgt som varierende parametere, for optimalisering av strukturen. Strukturen vist i Figur 3.56-6 var den valgte strukturen for prosjektet. Det presiseres at målet var å finne en struktur som både hadde sikre strukturelle egenskaper og redusert bruk av trematerialer. Det ble også satt krav som skulle påvirke optimaliseringen av tverrsnittene, slik at tverrsnittshøyden til både diagonalene og undergurten ble redusert. Bjelkene fikk dermed en lav tverrsnittshøyde og en stor bredde, noe som førte til en estetisk lett struktur. Det påpekes også at forskyvinger i strukturen var kritiske, ettersom det påvirket dreneringen av et tak med en stigning på 1%.



Figur 3.56: Forskjellig genererte strukturer fra casen [6].

Optimaliseringsverktøyet som ble brukt i dette case var Galapagos [6]. Resultatet viste et tre forbrukt på 4.75 m^3 per kvadratmeter, noe som påpekes å være relativt lavt for et tak med et spenn på 40 meter. Det presiseres også at dette resultatet er basert på et brasiliansk scenario, der verken snølast og jordskjelvberegninger ikke er med i analysen. I tillegg var det brukt en høykvalitets «Eucalypts Hardwood Glulma» fra ITA Construtoras. Dette trematerialet kan sammenlignes med den europeiske LVL-standard med en gjennomsnittlig elastisitetsmodul på ca. $1700 \frac{\text{KN}}{\text{cm}^2}$ [6].

For å evaluere resultatet fra Karamba3D og Bevaer, ble den optimaliserte strukturen eksportert til programvaren RFEM Timber Pro add-on [6]. Det presiseres at resultatet fra Beaver stemte overens

med RFEM Timber Pro, og modellen ble også godkjent i den andre evalueringen. Det nevnes også at Karamba3D-resultatet ikke samsvarte med RFEM resultatet. Grunnen til dette kommer av at Karamba3D utfører raske FEA-analyser, som gjør det mulig å utvikle optimaliseringsprosesser. I tillegg påpeker Pini m.fl. at det kan være en forskjell mellom Karamba3D og RFEM i «finite beam element» - modellen. Ettersom begge bruker Timoshenko-bjelketeori og vurderer lineær og kvadratisk interpolering av formfunksjonen på samme måte [6].

Det konkluderes dermed at RFEM Timber Pro var ga mer presise resultater i forhold til Karamba3D og Beaver [6]. Det påpekes derimot, at Karamba3D og Beaver var bedre egnet til strukturell optimalisering og var mer effektiv til å utforske flere strukturer, i forhold til den tradisjonelle designprosessen [6].

3.7.4 Generativ design til BIM-arbeidsflyt for vekt optimalisering

I forskingsartikkelen til Malgit m.fl. [99] var målet å utvikle en arbeidsflyt som gjør det mulig for optimaliseringsverktøy modellere og finne den minste totalvekten til en fagverksmodell. Dette ble gjort ved å integrere generativ design, strukturanalyse og BIM verktøy. I dette delkapittelet blir optimaliseringsverktøyene, metoden og resultatet fra studien presentert [99].

Metoden og optimaliseringsverktøyene

Malgit m.fl. [99] brukte flere forskjellige optimaliseringsverktøy og påpeker at det eksisterer to optimaliserings metoder. Den første metoden er optimalisering av ett objekt, som eksempelvis kan være reduksjon av totalvekten eller å øke kapasiteten til modellen. Den andre metoden er multi-objektiv optimalisering, der det er flere enn et optimaliseringsmål. Dette kan for eksempel være å både redusere totalvekten og å øke kapasiteten til modellen. De valgte optimaliseringsverktøyene for studien var Opossum, Goat og Galapagos som er innenfor den første metoden. Strukturanalysen og optimaliseringen ble utført i Rhino3D og Grasshopper sammen med Karamba3D. Fagverksmodellen ble først parametrisk generert i Grasshopper, før den ble overført til Karamba3D for strukturell analyse. Det presiseres at kombinasjonen av Karamba3D og ulike optimaliseringsverktøy genererte flere alternativer for fagverket, og den modellen med minst totalvekten ble valgt. Deretter ble modellen overført til Dynamo og transformert til Revit [99].

For å lage en modell som kan optimaliseres, presiserer Malgit m.fl. [99] at parametere som danner strukturen må først defineres. Videre er det viktig å definere nødvendig informasjon for den strukturelle analysen til Karamba3D. Dette inkluderer å definere oppleggene, materialet, lastene, valg av tverrsnitt og dimensjonene til dem. Når disse er definert, kobles optimaliseringsverktøyene sammen med både Karamba3D og de definerte design parameteren. Karamba3D vil vurdere kapasiteten til de genererte strukturene, som kommer av at optimaliseringsverktøyene varierer på design parameterne. Når analysen var ferdig, blir den mest optimale modellen overført til Dynamo, for å generere objektene i Revit [99].

Resultatene fra studien

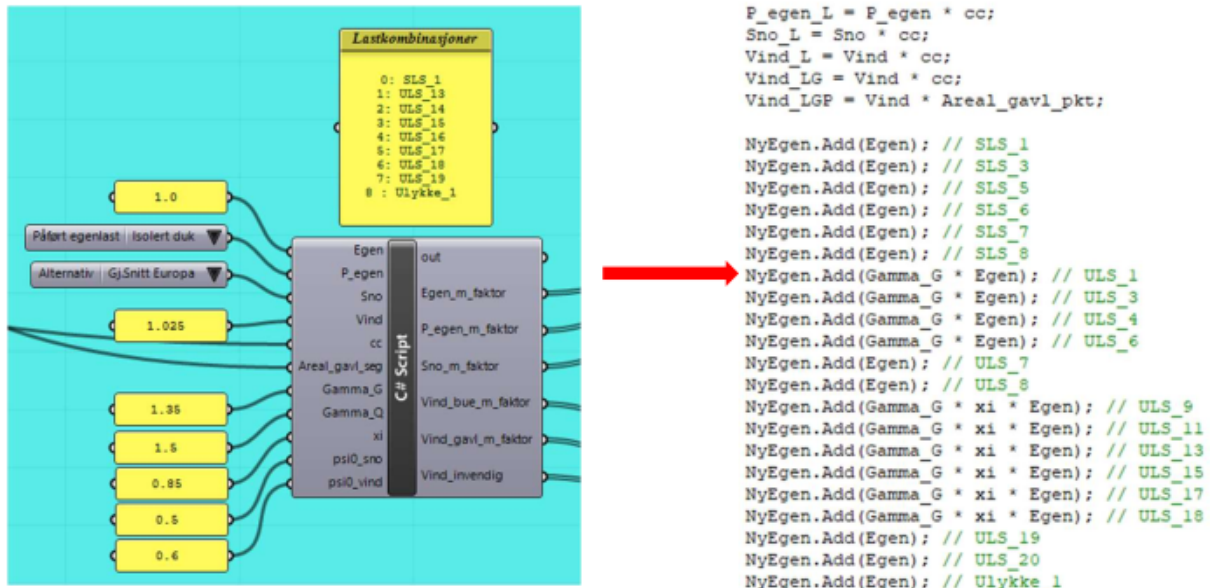
Resultatene fra de tre optimaliseringsverktøyene var relativt like [99]. Forskjellen mellom dem er presentasjonen av resultatene. Både Opossum og Galapagos gir lister over de mest optimale strukturene, mens Goat presenterer kun det beste resultatet. Galapagos genererte den høyeste

vekten på 1624 kg, mens både Oppssum og Goat hadde et identisk resultat på 1611 kg. Det presiseres også at det optimale resultatet fra Galapagos var likt resultatet til både Oppssum og Goat. Forskjellen var at Galapagos genererte en 0,5 meter høyere fagverkshøyde. Grunnen til forskjellen i svarene kan også komme fra algoritmene som brukes av hvert verktøy. Galapagos brukte evolusjonsalgoritmen, mens Oppssum brukte «*Radial basic function*» (RBF), og Goat brukte en begrenset optimalisering ved lineær tilnæringsalgoritme. Det konkluderes med at Oppssum og Goat er mer effektive optimaliseringsverktøy enn Galapagos [99].

3.7.5 Parametrisk design av fagverksbuer

En masteroppgave fra NTNU med tittel «Optimalisering av fagverksbuer for store frispennkonstruksjoner» [4], har som mål å bygge ett gjenbrukbart parametrisk system for fagverksbuer. Videre skal oppgaven anbefale en konstruksjonsutforming basert på konstruktive egenskaper, brukbarhet og indirekte kostnadsbesparelser. I tillegg har masterstudentene satt seg som delmål å kontrollere utnyttelsen av konstruksjonen [4].

Studentene opplever at beregningsprogrammet «Karamba» er til å stole på, da resultatene samsvarer godt med Robot modell og manuelle beregninger [4]. Det er verdt å bemerke seg metoden de benyttet for å definere lastkombinasjoner. Her har de etablert et C#-script som tar inn lastfaktorer, lastverdier og lastareal, hvor hver enkelt last-type blir omgjort til en liste med verdien fra lasten multiplisert med tilhørende lastfaktor. Et utklipp som presenterer C#-komponenten og selve scriptet, er presentert i Figur 3.57. [4].



Figur 3.57 Metode for å benytte C#-script for definering av lastkombinasjoner [4]

For å kunne anbefale en ideell konstruksjonsutforming, ble optimaliseringsverktøyene Galapagos og Octopus benyttet sammen med karambas «optimize cross section»-komponent [4]. Men ettersom verktøyene bare har mulighet for å tilfredsstillere ett mål om gangen, utviklet masterstudentene ett C#-skript for å benytte flere mål samtidig. Scriptet summerte kontrollene for både forskyvning, utnyttelse og knekking. Tallet fra hver av kontrollene ble satt til å være utrolig stort dersom maks utnyttelse overskrides, mens det ble satt lik null dersom kravet ble overholdt.

Optimaliseringsverktøyenes mål ble dermed å minimere den summerte verdien fra de tre kontrollene [4].

Studentene konkluderte med at alle optimaliseringsverktøyene har sine begrensninger [4]. Galapagos viser bare ett resultat og bare kan tilfredsstille ett mål om gangen. Octopus kunne derimot tilfredsstille to krav og vise resultatene som ble testet undervegs, men kunne bare endre én og én parameter. I tillegg hadde «Optimize Cross Section» komponenten begrensninger, da den ikke tok hensyn til forskjellige knekk lengder [4].

4 Forskerspørsmål

Oppgaven skal gi svar på følgende forskerspørsmål:

Hvordan utfordrer visuell programmering i Grasshopper den tradisjonelle metoden for prosjektering av limtrefagverk?

For å bygge opp under forskerspørsmålet, er det også utarbeidet tre underspørsmål. Med dette skal følgende spørsmål besvares:

- 1) Hvordan samsvarer dimensjoneringsresultatene fra Grasshopper med tradisjonelle beregninger?
- 2) Hvordan kan bruken Grasshopper være fordelaktig for optimalisering av teknisk utforming av limtre-fagverk?
- 3) Hvordan kan visuell programmering effektivisere dimensjoneringsprosessen for takstoler i tre?

4.1 Spesifikasjoner

Det vil ikke utføres «tradisjonelle beregninger» i oppgaven. Sammenligningsgrunnlaget baseres på oppdragsgivers utførelse av dimensjoneringsprosessen. Utskrift av beregninger for bruddgrensetilstand og knutepunkter, Robot-modell og muntlige beskrivelser av prosess-utførelse, er tildelt fra Sweco.

Tredje underspørsmål skal besvares basert på opplevelser og observasjoner. Det konkrete timeantallet ved utførelse av de forskjellige dimensjoneringsprosessene vil derfor ikke sammenlignes.

4.2 Avgrensninger

Det skal bygges et system i Grasshopper som bygger et limtrefagverk ved bruk av parametrisk design. Systemet skal utføre lineær elastisk analyse, samt benytte kreftene til å utføre kontroller i henhold til eurokode. Systemet vil imidlertid ikke ta hensyn til følgende:

- Eurokodens kontroller for bruksgrensetilstand
- Skjøtede forbindelser i over- og undergurter (alle gurter vil betraktes som hele elementer).
- Brannprosjektering

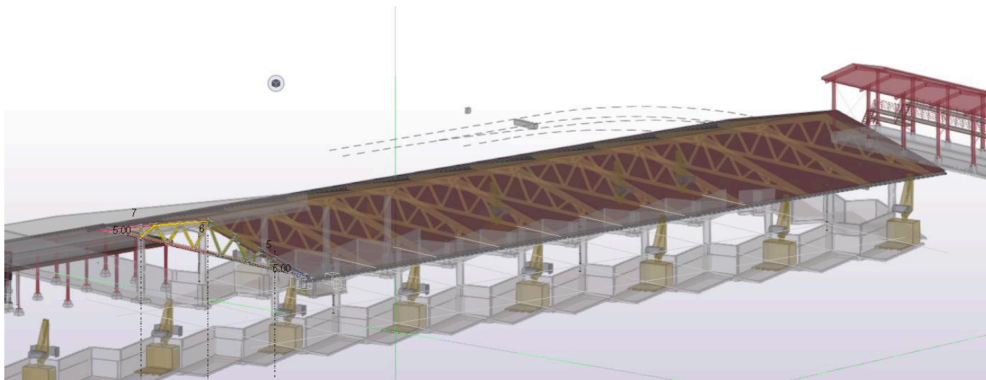
I tillegg vil kontroll av stålplater i henhold til Eurokode 3 være en nedprioritert del av oppgaven. Beregningene vil utføres med utgangspunkt i limtrebokas forenklete anbefalinger for stålplatekontroll.

5 Case/Materialer

I dette kapitlet presenteres caset som ble valgt for å besvare forskerspørsmålet, og materialene som er benyttet for å løse casen.

5.1 Case

I 2021 utførte Sweco en konstruksjons teknisk prosjektering av en gjenvinningsstasjon for Avfall Sør-Husholdning AS. Figur 5.1 viser skissen av prosjektet fra bergeneringsrapporten til Sweco [116], der det markerte området viser fagverket som utgjør caset for denne oppgaven. Figur 5.2 illustrerer det ferdigstilte prosjektet. Fagverket ble prosjektert i Robot Structural Analysis og knutepunktene ble beregnet i Excel. I denne oppgaven skal resultatet fra beregningsrapporten til Sweco sammenlignes av det genererte resultatet fra Grasshopper.



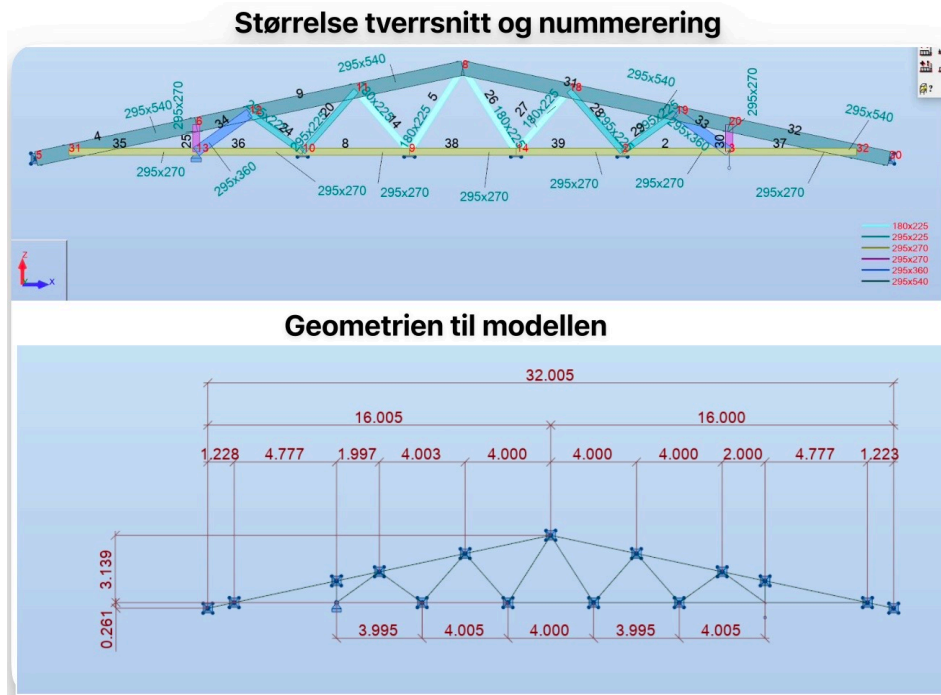
Figur 5.1: Valgt case er markert med flere farger.



Figur 5.2: Gjenbruksstasjonen av Avfall Sør-Husholdning.

5.1.1 Geometri

Limtre-fagverket er designet slik at horisontale krefter på langs av bygget, overføres fra taksskive til avstivningsstålet som er lokalisert i aksene med betongsøylene. Horisontale krefter som virket tvers på limtre-fagverket overføres fra fagverket til toppen av betongsøylene. Lengden til limtre-fagverket er 32 meter langt og høyden er fra 0 til 3.6 meter i midten av fagverket. Figur 5.3 viser dimensjonene til alle tverrsnittene, geometrien og nodene til modellen.



Figur 5.3: Geometri av limtrefagverket

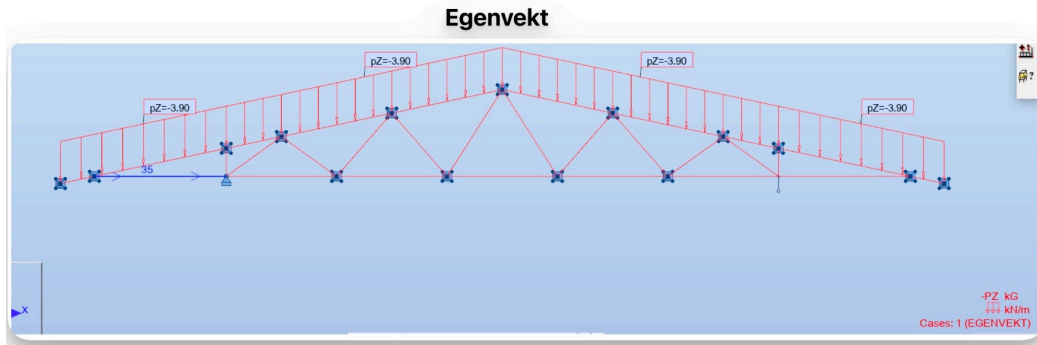
5.1.2 Randbetingelser

Overgruten er generelt sett kontinuering, men er splittet i toppunktet til node 6, 20 12 og 19. I disse nodene er det momentinnspenninger og knutepunktene er beregnet for momentene som oppstår i disse nodene i overgruten. Undergruten er splittet på midten og har ikke noen betydelige momentbelastninger.

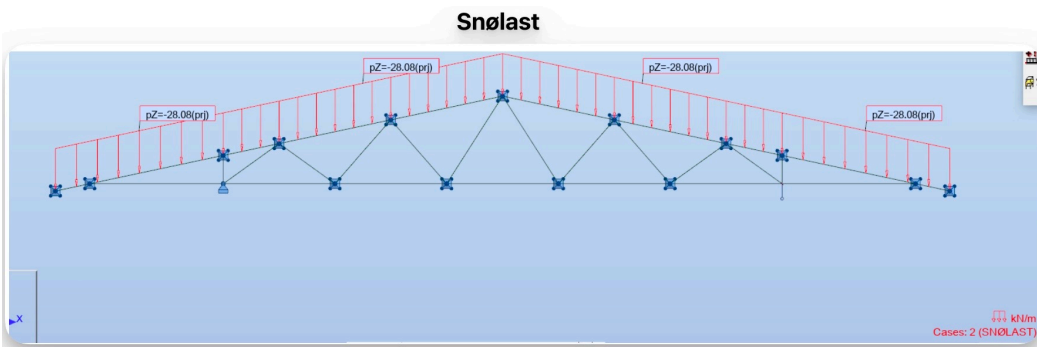
Alle stavene i fagverket er betraktet som leddet, spesielt for stavene som er i nærheten av opplegget ved betongsøylene for store knutepunkt, ettersom disse kan gi noe momentinnspenning. Knutepunktene i disse er blitt beregnet med lav utnyttelse fra aksiallasten, slik at momentinnspenningene blir tatt hensyn til.

5.1.3 Laster

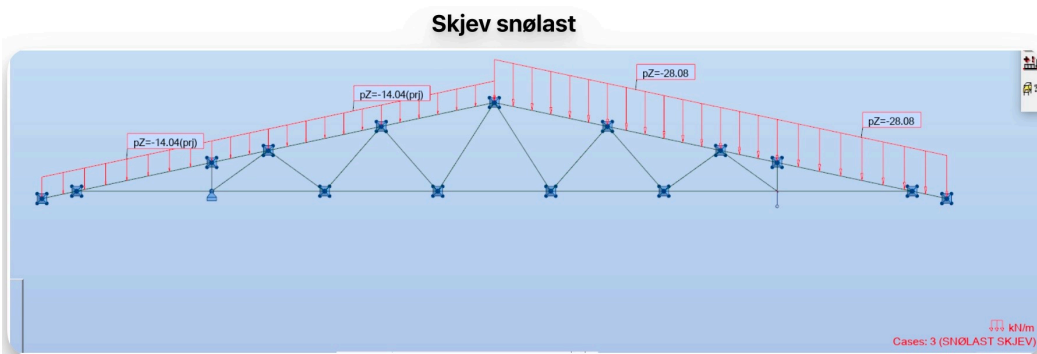
I figurene under presenteres alle lastene fra beregnings rapporten til Sweco. Lastene som er vurdert er egenlast (se Figur 5.4), snølast (se Figur 5.5), skjev snølast (se Figur 5.6), sentrert snølast (se Figur 5.7), vindlast oppløft (se Figur 5.8), vindlast jevnt trykk (se Figur 5.9), vindlast skjev trykk (se Figur 5.10) og vindlast på tvers (se Figur 5.11). Til slutt vises opplagerkreftene for hver lastkombinasjon fra Robot-filen til Sweco, se Figur 5.12.



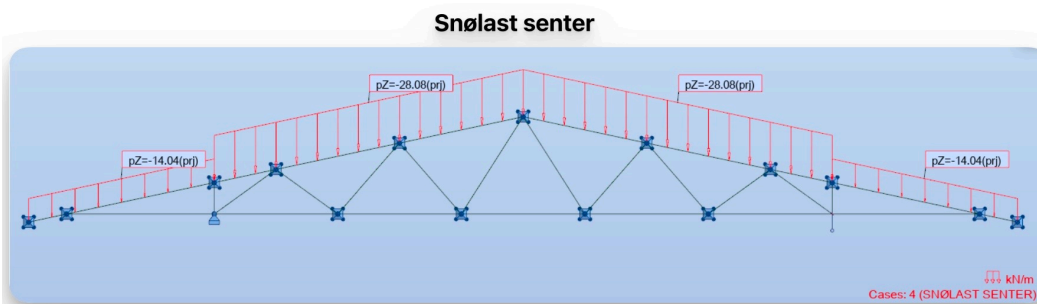
Figur 5.4: Egenvekt saltak



Figur 5.5: Snølast

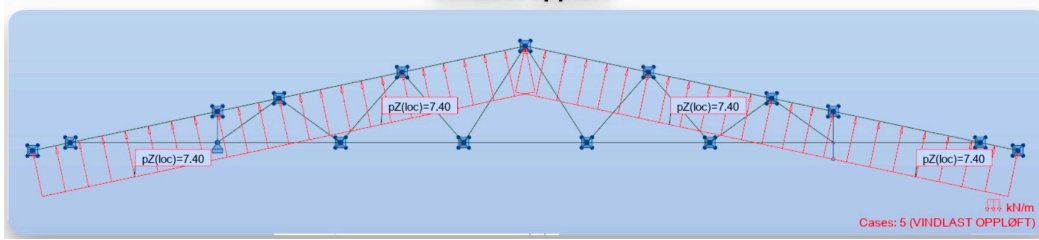


Figur 5.6: Skjev snølast



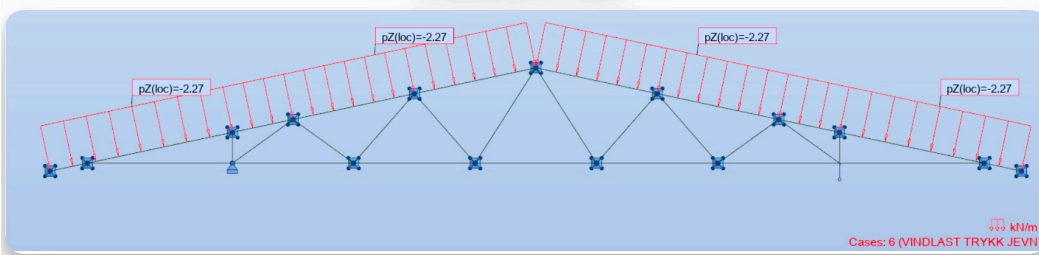
Figur 5.7: Sentrert snølast

Vindlast oppløft



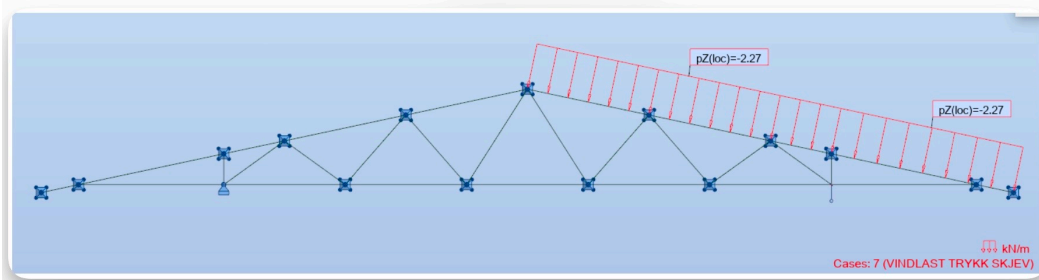
Figur 5.8: Vindlast oppløft

Vindlast jevnt trykk



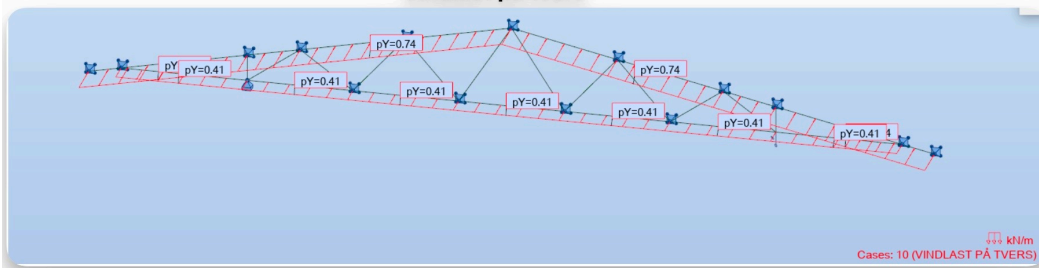
Figur 5.9: Vindlast jevnt trykk

Vindlast skjev trykk

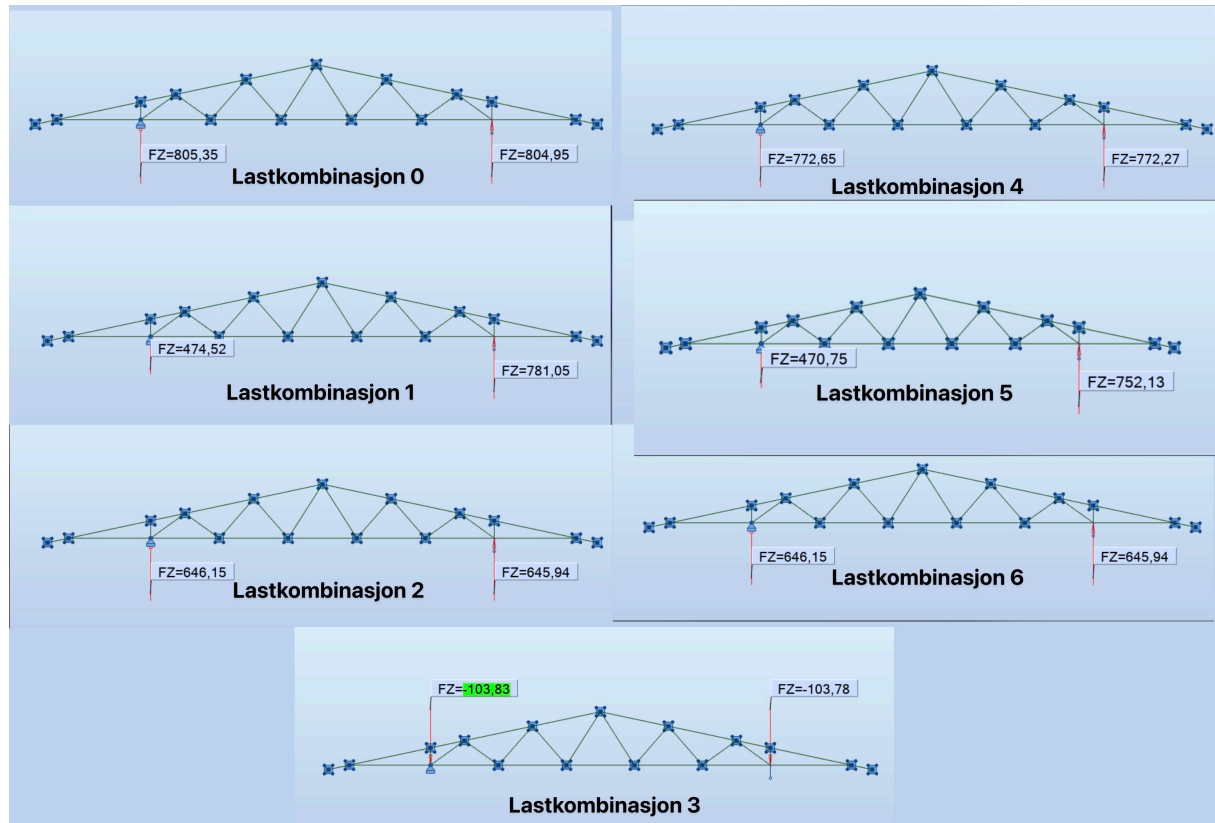


Figur 5.10: Skjev Vindlast

Vindlast på tvers



Figur 5.11: Vindlast på tvers



Figur 5.12: Oppleggs-krefter for hver lastkombinasjon fra Robot

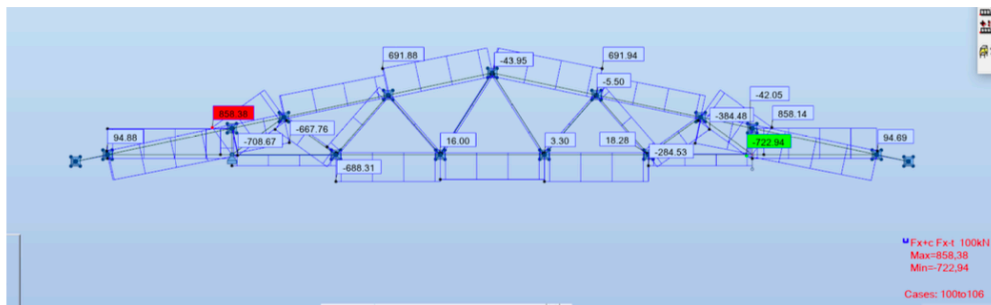
5.1.4 Materialkvalitet

Stavene, over- og undergruten av samme limtrekvalitet GL28c. Det er også forutsatt klimaklasse 2, ettersom limtrekonstruksjonen er beskyttet mot regn. Stålplatene er av kvalitet S355, mens dyblene har en flytespenning på 200 MPa og en bruddspenning på 700 MPa. Disse materialkvalitetene er gitt av Sweco [116].

5.1.5 Resultater

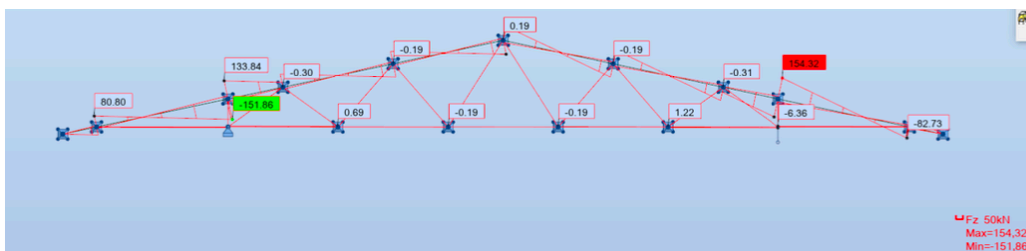
I figurene under presenteres ULS resultatene fra Robot beregningen til Sweco. Dette er resultatene for normalkraft, skjærkraft (F_z og F_y), og moment (M_y og M_z). Resultatene for belastningskrefter, utnyttelse av både elementer og knutepunkter er vist i kapittel 7, siden resultatene skal sammenlignes med resultatene fra Grasshopper [116]. Resultatene for belastningskreftene er fra Robot-modellen som ble tilsendt fra Sweco, og resultatene fra elementanalysene og knutepunktene er fra beregningsrapporten til fagverket. Denne ble også tilsendt fra Sweco. Disse blir kun referert til, og er ikke gitt som vedlegg.

Normalkraft – N



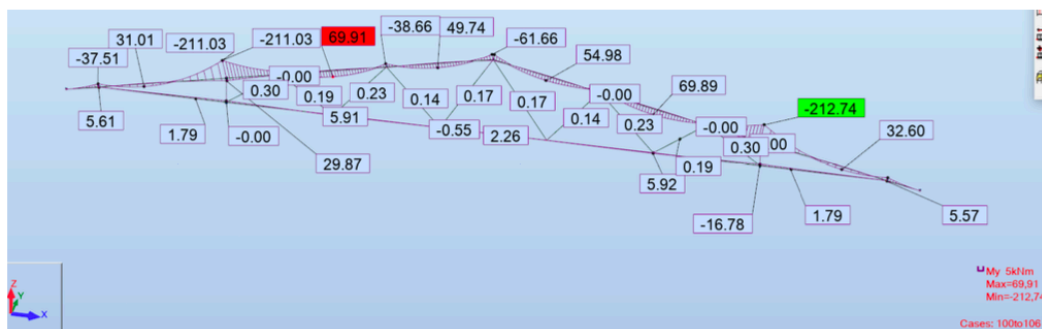
Figur 5.13: Normalkraft diagram

Skjærkraft – Fz



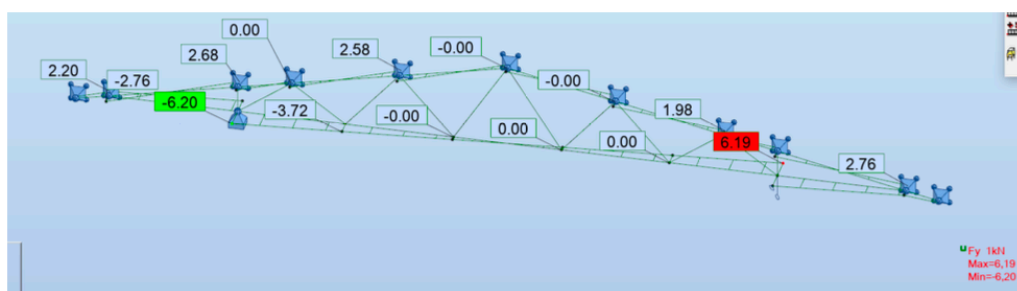
Figur 5.14: Skjærkraft diagram z-retning

Moment – My



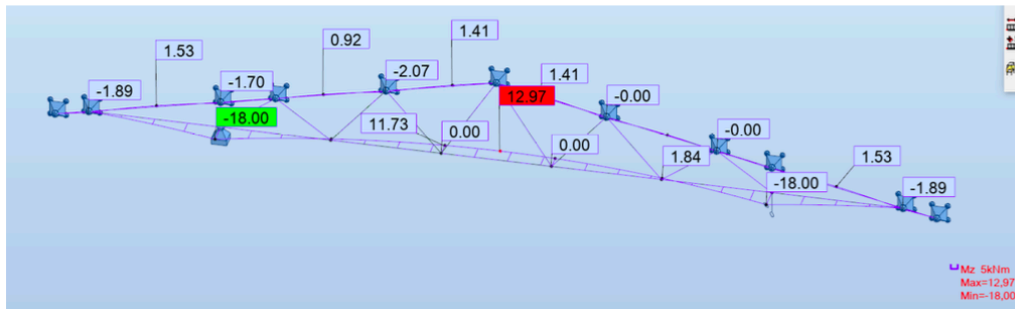
Figur 5.15: Moment diagram y-retning

Skjærkraft – Fy



Figur 5.16: Skjærkraft diagram y-retning

Moment – Mz



Figur 5.17: Momentdiagram z-retning

5.2 Materialer

Her presenteres de digitale verktøyene som er brukt for å løse forspørsmålet til oppgaven.

5.2.1 Rhinoceros 7 SR28

Rhinoceros også forkortet til Rhino3D eller Rhino, er et dataassistert konstruksjonsprogram (DAK) eller på engelsk Computer Aided Desing (CAD) [117]. Programvaren visualiserer geometri som enten er direkte modellert, eller som kommer fra tilleggsverktøy [118]. Dette gjør det mulig å opprette komplekse modeller og geometrier både i 2D og 3D-modeller. På grunn av dette kan programvaren brukes innenfor alle bransjer [119].

Rhino er utviklet av Robert McNeel & Associates, og ble først utgitt i 1998 [118]. Det er et modelleringsverktøy som benytter Non-Uniform Rational Basis Spline (NURBS) [119]. NURBS beskrives som matematiske representasjoner av 3D-geometri som nøyaktig kan gjengi 2D-linjer, kurver, eller overflater i datagrafikk [119, 120]. På grunn av fleksibiliteten og presisjonen NURBS-modellen tilbyr, kan den derfor benyttes i alle prosesser. Eksempelvis til illustrasjon, animasjon og til produksjon [120]. Programvaren kan også brukes i flere andre programvarer, som for eksempel Grasshopper, Dynamo, Revit og Tekla.

5.2.2 Grasshopper

Grasshopper er forklart i kapittel 3.1.4, ettersom forståelse om bruken av Grasshopper var viktig å gjennomgå på en grundig måte. Nedlastede verktøy for Grasshopper modellen nevnes i delkapittel 6.3.1.

5.2.3 Autodesk Robot

Robot Structural Analysis Professional er en programvare som brukes av konstruksjonsingeniører for strukturell belastningsanalyse [121]. Programvaren er en del av BIM-programmene til Autodesk og tillater dermed en BIM koordinert arbeidsflyt. Belastningsanalysene utføres etter verifiserte standarder, og hjelper ingeniører til å frembringe robuste og konstruktive design på en nøyaktig måte. Programvaren kan lage både 2D og 3D modeller, av forskjellige materialer som stål, betong og tre [121].

5.2.4 Microsoft Office Excel

Microsoft Excel er en programvare som kan løse matematiske problemer på en effektiv, enkle og oversiktlig måte. Dette gjøres ved bruk av regneark, som gjør det mulig å analysere og beregne data etter spesifikke behov [122]. Programvaren er spesielt fordelaktig for itererende beregninger, og brukes ofte av ingeniører til beregningskontroll av strukturerer.

6 Metode

Formålet med studien er å undersøke bruken av visuell programmering for å effektivisere dimensjoneringsprosessen for limtrefagverk. Derfor bygges det ett system basert på parametrisk design. Gjennom oppgaven vil det bli utviklet visualiserings- og beregningskomponenter som kan gjenbrukes ved visuell programmering av andre limtrefagverk. Dette skal også gi muligheten for å optimalisere geometrien for spesifikke formål.

Det benyttes i hovedsak kvantitativ metode for å besvare forskerspørsmålet i denne oppgaven. Metodekapittelet inneholder etablering av bakgrunnskunnskap, datainnsamling, tilleggsverktøy til Grasshopper, etablering av modell i Grasshopper og metode for optimalisering.

6.1 Etablering av bakgrunnskunnskap

For oss begge var visuell programmering og parametrisk design nye konsepter. I starten av prosjektet var det dermed nødvendig gjøre med mer grunnleggende litteratursøk for å tilegne kunnskap og informasjon om emnet til oppgaven. Vi fant mye informasjon generelt om visuell programmering og dens bruk i byggebransjen. Det var derimot begrenset informasjon om visuell programmering med bruk av programvaren Grasshopper for oppbygging av fagverk, spesielt ved bruk av limtre. Det eksisterte derimot, nok informasjon til å få en overordnet forståelse av bruken og mulighetene for oppbygging av fagverk i Grasshopper.

Det neste vi utførte var å se igjennom tidligere masteroppgaver som brukte Grasshopper til parametrisk design av fagverk, noe som ga oss innsikt i mulighetene med Grasshopper [4, 118, 123]. Vi tok også kursene «*Learning Grasshopper*» [124], «*Grasshopper: Generative Design for Architecture*» [125] og så noen videoer fra «*Grasshopper: Tips, Tricks, and Techniques*» [126] fra LinkedIn Learning for å lære mer om bruken av programvaren Grasshopper.

Vi så også noen videoer fra YouTube-kanalene som Rhino Grasshopper [127], Parametric Engineer [128], Diego Apellaniz [129], DCO Architecture [130], BIM Corner [131], og Structured Parametrics [132] for å lære mer om visuell programmering av enkle strukturer i Grasshopper. Disse kanalene er laget av ingeniører og arkitekter som illustrerer og forklarer programmeringen av både enkle og komplekse strukturer. I tillegg forklares bruken av enkelte plug-ins i Grasshopper, som skal hjelpe med oppbyggingen av strukturene.

Kapittel 3.1 beskriver visuell programmering, BIM, Grasshopper, parametrisk design og modellering. Bortsett fra BIM var alle disse teamene helt nye for oss. Det var dermed viktig å gjøre en grundig gjennomgang av disse nye temaene, ettersom de er essensielle for å løse forskerspørsmålet. Dette var med på å oppbygge en forståelse av prosessen for å visuelt programmere parametriske modeller i Grasshopper. Gjennomgangen av sentrale parametere og komponenter fra Grasshopper var viktig å få en oversikt over, og ble derfor gjennomgått grundig i kapittel 3.1.4.

Kapittel 3.2 blir tre som byggematerialet beskrevet. Mange av disse temaene har blitt gjennomgått i utdanningen, men det var likevel nyttig å for oss å ha en grundig gjennomgang av materialet. Tre er et levende materiale og oppfører seg på en annen måte i forhold til både stål og betong, noe som man må ta hensyn til ved konstruksjon med tre som byggematerialet. De mest brukte kildene i dette

delkapittelet var bøkene: Limtreboka, Dimensjonering av trekonstruksjoner, og Timber Engingering [22, 57, 58].

I Kapittel 3.3 blir kunnskap om fagverket gjennomgått. Siden en stor del av oppgaven handler å parametrisk designe og analysere et fagverk, var det viktig for oss å forstå de forskjellige fagverksmodellene og hensynene som må gjøres. Limtreboka, Dimensjonering av trekonstruksjoner og Eurokode 5 var sentrale kilder i oppbyggingen av forståelse av dette [22, 58, 63].

I kapittel 3.4 blir dette gjennomgått essensielle faktorer for dimensjonering av trekonstruksjoner i henhold til EK5 [63]. Dette er temaer som er gjennomgått både i bachelor- og masterutdanningen, men det var likevel nyttig for oss å få en mer grundig forståelse av dem. Ettersom det visuelt skal programmeres komponenter som utførere utnyttelses kontroller av elementene i fagverket.

Kapittel 3.5 beskriver forbindelser og bruddformene som kan oppstå i knutepunktene. Dette er noe som er gjennomgått både i bachelor- og masterutdanningen, men siden forbindelser er et kritisk område for trekonstruksjoner, var det viktig for oss å ha en grundig gjennomgang av dette temaet. Det skal programmeres komponenter som utfører analyse av knutepunktene, og det var dermed nyttig å ha en forståelse av formlene for å kontrollere knutepunktene. Kildene som ble bruk for dette var Eurokode 5, boka Dimensjonering av trekonstruksjoner, Limtreboka, Timber Engingering, Kontroll av fagverksbruer av Vegvesenet, Masteroppgaven om store dybelforbindelser og flere av kildene denne oppgaven bruker, samt flere forskningsartikler som ble funnet i Google Scholar [22, 57, 58, 63, 64, 68, 76, 79, 92].

Kapittel 3.6 gjennomgår optimalisering. Dette var et nytt tema for oss, og det ble derfor gjennomgått grundig for å oppbygge en ny kunnskap om dette temaet. Det var viktig for oss å forstå de forskjellige metodene, mulighetene og algoritmene som kan brukes for å optimalisere strukturen.

I Kapittel 3.7 blir det gjennomgått fem studier om parametrisk design og visuell programmering. Disse studiene skal brukes for å sammenligne resultatet til oppgaven og eventuelt hjelpe til med programmeringen av fagverket i Grasshopper. Hver studie inneholder forskjellige temaer innenfor parametrisk design og visuell programmering. Den første studien påpeker svakheter med parametrisk design, og tips som kan forbedre resultatet. Den andre studien viser programmering av en stålforbindelsene. Vi fant ingen studier som hadde programmert både beregninger og visualisering av stålplater med dybler i trekonstruksjons forbindelser. Dette var den eneste studien vi fant som utforsket visuelt programmering av forbindelser. Den tredje studien viser bruken av et verktøy for beregningen av trekonstruksjoner i Grasshopper, og var dermed sentral for i denne studien. Den fjerde studien bruker både Karamba3D og forskjellige optimaliseringsverktøy for optimalisering av et stål fagverkt. Det eksisterte ikke noen studier angående optimalisering med bruk av både Karamba3D, optimaliseringsverktøy som kan brukes i Grasshopper, og med tre om materiale. Denne studien ble valgt ettersom den oppfyller to av de tre kravene. Den femte studien er en masteroppgave som brukte Kramba3D, og hadde relevant programmering for lastkombinasjoner.

6.2 Datainnsamling

Innsamling av data er en viktig og kritisk del av forskningsprosessen. Ettersom det er i denne prosessen forskeren eller forskerne innsamler fakta og informasjon som er nødvendige for besvarelse av forskerspørsmålet [133]. Kriterier for datainnsamling ble derfor valgt og bestemt tidlig i prosjektet. Det var viktig for oss å ha kriterier som var relevante både for forskerspørsmålet og formålet med studien. I dette underkapittelet presenteres de utvalgte kriteriene og hvordan vi har brukt dem i datainnsamlingsprosessen.

6.2.1 Krister for valg av litteratur

For det første ble det valgt å benytte søkemotorene Google og Google Scholar som et kriterium for datainnsamlingen. Dette kommer av at Google har raske og mer relevante søkeresultater i forhold til andre søkemotorer [134, 135]. Google forstår også søksformålet bedre og gir dermed mer nøyaktige og relevante resultat [135].

Det andre kriteriet var å innsamle så mye data som mulig fra akademisk litteratur. Grunnen til dette er at denne litteraturen gjennomgår en streng godkjenningssprosess fra fageksperter før den publiseres, noe som gir en høy pålitelighet [136]. Den akademiske litteraturen inkluderte fagfellevurderte artikler, publikasjoner, standarder eller rapporter. Elektroniske artikler ble ofte foretrukket for å kunne hente mer data fra referanselistene. Dette bidro til å redusere søksmengden og øke påliteligheten ved å bekrefte informasjonen fra de henviste litteraturene. Vi brukte også relevante bøker fra bachelor- og masterstudiet, samt relevant litteratur som ble gitt av veiledere.

Det tredje kriteriet omhandlet innhenting av data fra ikke-akademisk litteratur. Ved bruk av disse kildene, måtte vi utøve mer omfattende kildekritikk for å vurdere om kvaliteten var god nok til å benyttes. Dette inkluderte å oppsøke flere uavhengige kilder som nevner det samme, for å bekrefte informasjonen. I tillegg ble forfatter, utgiver, språk, og referanser i kilden vurdert til å øke påliteligheten [137].

Det fjerde kriteriet innebærer å utføre søkene hovedsakelig på det internasjonale hjelpespråket, engelsk. Engelsk er et offisielt språk i mange land, og blir også brukt som undervisnings- og administrativt språk i flere deler av verden [138]. Dette gjør at det blir flere søkstreff og mer omfattende resultater enn ved å søke på norsk. For eksempel kan et søk på «*Visual programming Grasshopper*» gi omtrent 24 100 resultater, mens det norske søket «*Visuell programmering Grasshopper*» gir omtrent 162 resultater. Det norske søket gir også resultater på svensk, dansk, tysk og engelsk. De norske resultatene er som oftest master- eller bacheloroppgaver, mens i det engelske søket er flere av treffene akademisk vurderte forskningsartikler, noe som øker påliteligheten til dataene.

Det femte kriteriet handler om å bestemme tidsperiode for søket. Dette er viktig for å finne data som er mest relevante for oppgavens formål. Valget av tidsperiode ble basert på emnet som ble undersøkt og antall resultater fra søket. For eksempel ble tidsperioden for søket av litteratur i kapittel 3.7 i utgangspunktet satt til tre år, for å få tak i den mest oppdaterte litteraturen om bruken av Grasshopper. Dette ble utvidet til fem år for å øke antallet på relevante artikler. I kapittel 3.5 ble det derimot ikke satt en bestemt søkeperioden, siden forskning på dybel-forbindelser er blitt utført i

flere tiår. Selv om noen av studiene kan være over 20 år gamle, er de fortsatt relevante og brukes som grunnlag for nyere forskningsartikler. Undervisningsbøkene «Dimensjonering av Trekonstruksjoner», «Limtreboka» og «Timber Engineering - Principles for Design» er også veldig relevante i dag, selv om de ble publisert for mer enn seks år siden [22, 57, 58].

Det sjette kriteriet handler om å bruke nøkkelord ved søk i Google Scholar, for å finne de mest relevante artiklene og fjerne de irrelevante. Nøkkelord karakteriserer innholdene i en fagartikkel [139], og gir en veiledning til søket som Google Scholar skal utføre. For å utføre et mer avansert søk ble søkefunksjonen i Google Scholar brukt. Et eksempel på dette er når vi søkte etter tidligere forskning med bruk av Grasshopper for oppbygging av limtrefagverk. I dette søket ble følgende nøkkelord brukt: «Optimization Grasshopper plug-in glulam» med den eksakte setningen: «Dowel connection». Dette ga kun tre treff. Siden det eksisterte svært få fagartikler om fagverkt med trematerialer og dybel-forbindelser, ble søket utvidet ved å søke på stål konstruksjoner med bolteforbindelser, noe som økte treffet. Vi utvidet også søket ved å søke etter mer relevante emner, som optimalisering av fagverk og bruk av Karamba3d som plug-in. Dessverre eksisterte det veldig få artikler som visuelt programmerte limtrefagverk i Grasshopper.

Det sjuende kriteriet involverer bruk av tidligere master- og bacheloroppgaver for å øke påliteligheten av oppgaven. Vi forsøkte å unngå å bruke slik oppgaver, men i enkelte tilfeller ble det nødvendig å inkludere dem når fagartiklene ble for tekniske og vanskelige å forstå. Vi skrev inn «master» og «bachelor» i *uten ordene* funksjonen til Google Scholar for å fjerne disse fra resultatet. I de tilfellene master- og bacheloroppgaver ble brukt, gikk vi igjennom fagartiklene fra referanselisten for å forsikre oss om at informasjonen var korrekt. På grunn av mangel på fagfelleverderte artikler om emnet til denne oppgaven, ble de få tilgjengelige masteroppgavene benyttet for å sammenligne resultatene fra vår oppgave.

Det åttende kriteriet handler om vurderingen av relevansen til den utvalgte litteraturen. Dette ble gjort ved å lese gjennom sammendraget og konklusjonen til litteraturen for å øke forståelse av innholdet og vurdere relevansen. Ordsøk ble også utført i artiklene for å effektivt finne relevant informasjon, siden ikke all informasjon i en artikkel var relevant for oppgaven. Denne metoden ble også brukt i tilfeller når søket ga mange treff og når det var nødvendig å innhente spesifikk informasjon om et bestemt tema.

6.2.2 Induktiv analyse

Induktiv resonering ble brukt som en metode for å forme og strukturerer forskningsspørsmålet. Denne metoden er gunstig når formålet til studien ikke er definert eller når det mangler kunnskap om emnet som skal undersøkes [140]. Induktiv resonering er en prosess som skaper konklusjoner fra de innsamlede dataene, ved å koble ny informasjon til teori. For å identifisere betydningsfulle emner som svarer på forskningsspørsmålet, ble artiklene analysert med et åpent sinn [141]. Denne prosessen brukes ofte for å organisere kvalitative data, og ble derfor brukt ved innhenting av informasjon i starten av oppgaven. Relevante fagartikler ble gjennomgått med to kriterier; forståelse av innholdet og for avdekke betydningsfulle temaer. Måten disse kriteriene ble utført på var å direkte skrive notater inn i teksten imens artikkelen ble gjennomlest. Videre ble notatene organisert og klassifiseres til bestemte grupper. Hensikten med dette var å øke forståelsen og oppbygge kunnskap

om temaer. Ved å klassifisere gruppene var det mulig å bestemme hva som skulle inkluderes i hver gruppen [140]. Dette skape en oppbygging til bakgrunnskunnskapskapittelet og potensielle underspørsmål til forskerspørsmålet kunne bestemmes.

6.2.3 Deduktiv analyse

En deduktiv analyse kan være nyttig når formålet er å teste tidligere teori i en annen sammenheng [140, 141]. Denne metoden ble benyttet flere ganger i oppgaven, både i innhenting av litteratur og i diskusjonskapittelet. I innhenting av litteratur blir deduktiv analyse brukt etter den induktive analysen, slik at man kunne være mer kritisk til de innhentede artiklene. Forståelsen og kunnskapen som ble oppbygget igjennom den induktive analysen, gjorde at vi kunne være mer kritiske i valg av litteratur. I tillegg benyttes denne metoden i diskusjonskapittelet, hvor resultatet fra denne oppgaven diskuteres mot både resultatene til Sweco, og resultatene til den innhentede litteraturen i kapittel 3.7.

6.3 Tilleggsverktøy til Grasshopper

For å bygge opp limtre-fagverket i Grasshopper, ble flere plug-ins lastet ned fra food4rhino nettsiden [142]. Nettstedet inneholder over 1400 plug-ins som er designet for å utvide verktøyene i Grasshopper. McNeel Europe er utvikler og vedlikeholder av nettstedet, og er en avdeling av Robert McNeel & Associates som er utvikleren av både Rhino3d og Grasshopper [142].

Tilleggsverktøyene kan utvikles av alle som kan programmere med et .Net-programmeringsspråk [143, 144]. For å undersøke kvaliteten på verktøyene, ble antall nedlastinger, rangeringen, og anmeldelsene fra kommentarfeltet for hvert enkelt verktøy vurdert. Dette var noe som LinkedIn-kurset til Zesk påpekte var viktig [125]. Verktøyene som ble nedlastet og brukt i studien, ble funnet gjennom anbefalinger fra LinkedIn-kurs og fra YouTube-videoer. I tillegg ble egne søk og nettsiden grasshopperdocs benyttet, som viser anbefalte plug-ins basert på bruken av Grasshopper [145].

6.3.1 Nedlastede plug-ins

I dette delkapittelet presenteres de nedlastede verktøyene, hva de utfører og hvem som anbefalte dem. Det ble nedlastet flere plug-ins, men disse ble ikke brukt til å komme frem til resultatet av studien, og blir derfor ikke omtalt.

Pufferfish V3.0

Pufferfish er en populær plug-in som inneholder et sett med 330 komponenter for å endre formen på visuelt programmerte objekter [146]. Verktøyet benytter hovedsakelig parametere og faktorer som input, for å gi en mer tilpasset kontroll over operasjoner som utføres i Grasshopper. Verktøyet er utviklet av ekimroyrp og er et gratis å bruke, med over 250 000 nedlastinger. Pufferfish ble anbefalt av nettsiden grasshopperdocs som et grunnleggende verktøy for visuell programmering i Grasshopper [145].

Heteroptera 0.7.3.5

Heteroptera er en gratis plug-in utviklet av Helioripple studio, og har over 90 000 nedlastinger [147]. Dette verktøyet brukes hovedsakelig til datastruktur og manipulering av elementer, med

komponentene som utfører matematiske oppgaver. Heteroptera ble anbefalt av både LinkedIn-kurset «*Grasshopper: Tips, Tricks and Techniques*» og fra grasshopperdocs-nettstedet [126, 145].

Karamba3D 2.2.0

Karamba3D er et verktøy som er beskrevet i kapittel 3.1.5, og det har blitt nedlastet over 80 000 ganger [148]. Verktøyet er mest egnet for bruk av stål, men kan også brukes for både tre og betong. Veilederen vår anbefalte Karamba3D, og det har også blitt brukt flere studier med bruk av Grasshopper. Det finnes en gratis versjon av verktøyet, men den fungerte kun med 1.3.3 versjonen av Karamba3D, og kunne kun analysere en struktur med maks 20 elementer. Derfor var det nødvendig å kjøpe studentversjonen som kostet 30 euro, noe som gjorde det mulig å bruke den nyeste versjonen av Karamba3D (2.2.0 versjonen) [149].

LunchBox

Lunchbox er et gratis verktøy utviklet Nathan Miller, og den siste versjonen ble utgitt 02.11.2020 [150]. Verktøyet har blitt nedlastet over 700 000 ganger og formålet til verktøyet er å utforske matematiske formler, strukturerer og arbeidsflyten i Grasshopper. Det tilbyr også flere generelle maskinlæringskomponenter som utfører regresjonsanalyse, gruppering og nettverksanalyse. Lunchbox ble anbefalt av YouTube-kanalen Rhino Grasshopper og av nettsiden grasshopperdocs som et grunnleggende verktøy til visuell programmering i Grasshopper [145, 151].

TT Toolbox 2.0.3

TT Toolbox er et gratis verktøy utviklet av COREstudio med over 95 000 nedlastinger [152]. Verktøyet lar brukeren overføre data direkte fra Grasshopper til Excel, og gir dermed mulighet for dokumentasjon av beregninger som utføres i Grasshopper. Det er anbefalt av nettsiden grasshopperdocs og er kategorisert som et grunnleggende verktøy for bruk av Grasshopper [145, 152].

Beaver 1.0.2

Beaver Structures har utviklet Beaver, et verktøy som er spesialisert på trekonstruksjoner og har blitt lastet ned over 1 000 ganger [153]. Verktøyet kan utføre parametrisert strukturell analyse og design i henhold til Eurokode 5 og fungerer sammen med Karamba3D. På nettstedet food4Rhino er det mulig å laste ned eksempelfiler som viser hvordan analysen oppbygges, samt et Excel-dokument som inneholder materialeegenskapene for C14-C50, D30-D70, GL20h-GL32h, GI20c-GL32c, Kerto-S LVL og Kerto-Q, LVL (se vedlegg 1). Verktøyet er gratis, men for å bruke alle funksjonalitetene kreves en lisens, som for øyeblikket kan fås gratis ved å sende en forespørsel til utviklerne via e-post. Det tok oss en uke å få lisensen etter å ha sendt forespørselen. Verktøyet ble funnet gjennom litteratursøk i Google Scholar, der fagartikkelen fra Pini m.fl. [6] viste bruken av Beaver. I tillegg var det anbefalt som et analyseverktøy for trekonstruksjon med Karamba3D på Mcneel-nettstedet sitt forum [153, 154].

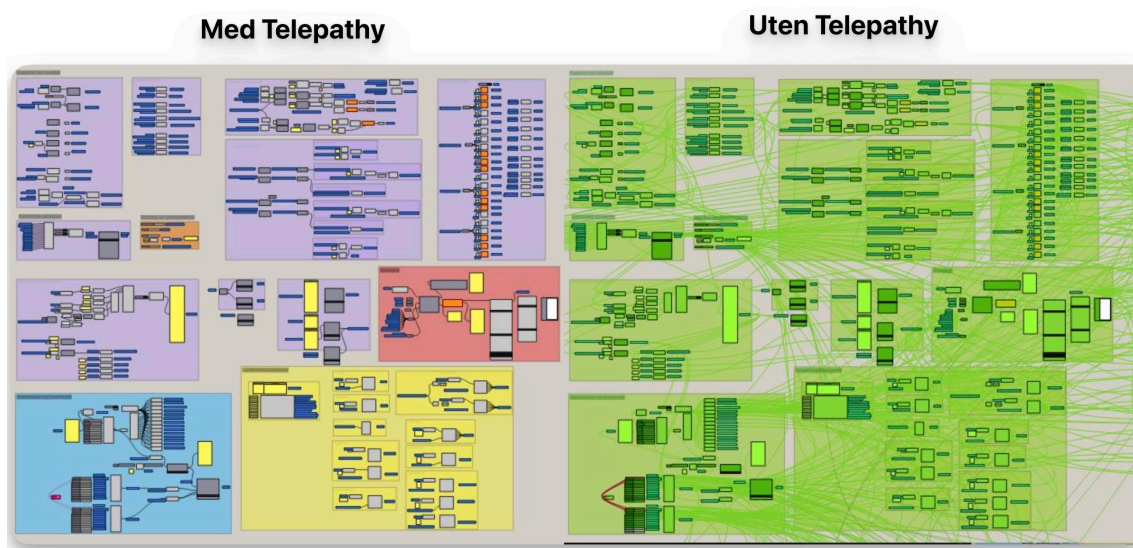
SnappingGecko 1.1.2

Snappinggecko er en objektsnappingsfunksjon utviklet av weArk, og den er blitt lastet ned over 19 000 ganger [155]. Verktøyet er designet for å hjelpe brukere med å organisere parameterne og komponentene i Grasshopper ved å lage linjer som viser posisjonene til objektene som er i nærheten

av det valgte objektet. Dette verktøyet ble anbefalt av YouTube-kanalen BIM Corner for å hjelpe til med å rydde opp i rotet som kan oppstå når det jobbes med flere parametere og komponenter i Grasshopper. Snappinggecko er også gratis for alle brukere [155, 156].

Telepathy 1.0.0.0

Dette verktøyet er et gratis verktøy utviklet av marcsyp som er nedlastet over 9 000 ganger [157]. Formålet til verktøyet er å rydde opp i rotet som oppstår i Grasshopper, ved å erstatte alle ledningene mellom komponentene og parameterne med trådløse sender- og mottakerenheter som kan navngis. Tilkoblinger mellom enhetene opprettes automatisk basert på det gitt navnet. Figur 6.1 viser hvordan Telepathy rydder opp i Grasshoppermodellen. Verktøyet ble anbefalt av YouTube-kanalen BIM Corner og fra en av veilederne fra Sweco [156].



Figur 6.1: Bruk av Telepathy i Grasshopper

Bifocals

Denne verktøyutgivelsen er utviklet av NBBJ digital Practice og har blitt lastet ned over 120 000 ganger [158]. I standard Grasshopper vises enten navnene eller logoen til hver enkelt komponent. Formålet med verktøyet er å vise både navnet og logoen til hver enkelt komponent samtidig. Ved å legge verktøyet til i Grasshopper, vil alle navnene på de brukte komponentene vises i tekstform. Dette gjør det enklere å følge kurs og videoer som bruker Grasshopper, da noen brukere foretrekker å se navnene på komponentene, mens andre foretrekker å se logoen. Verktøyet er gratis og ble anbefalt av YouTube-kanalen Rhino Grasshopper [151].

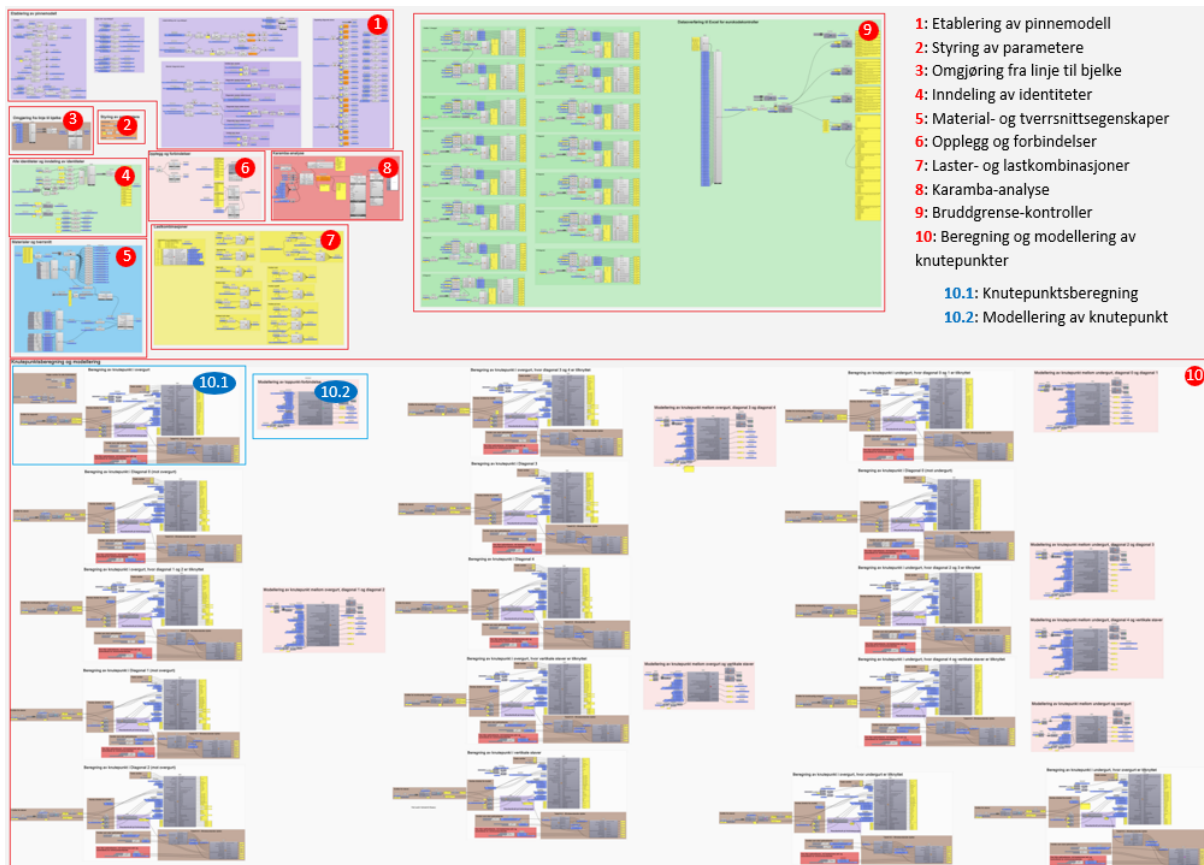
Opossum 3.0.2

Opossum er et optimaliseringsverktøy for Grasshopper, som kan utføre multioptimaliseringsformål eller enkelt-objektiv optimalisering. Den brukeren kan bruke flere forskjellige optimaliserings algoritmer, men den mest brukte algoritmen er RBFOpt (Radial Basis Function) som også er kjent for å være derivatfri optimalisering [159]. Denne algoritmen er et Python-bibliotek for «black-box»-optimalisering (BBO). BBO analyserer og vurderer designet til en algoritme som skal løse problemer der objektet-funksjonen « f » og begrensningene som bestemmer variablene « x » er ukjent [160]. Dette gjør at man ikke vil få samme svar, selv om samme variabler brukes [159]. RBF interpolasjon beskrives som en avansert metode innen tilnærmingsteori, og skal lage nøyaktige interpolasjoner av

ustrukturert data [161]. RBF verdien baseres på avstanden mellom variabel-inputen. Den mest brukte RBF er Gaussian RBF, som har samme form som Gaussian sannsynlighetsfunksjon [162]. Dette verktøyet ble brukt i forskningsartikkelen i kapittel 3.7.4 [99], og har over 11 500 nedlastinger på food4rhino nettsiden [163]. Verktøyet er også gratis, men det trengs en linsens for å bruke det. Linsens ble gitt ved å sende e-post til utviklere av Opossum. E-post adressen er gitt på food4rhino nettsiden, men kan også finnes ved å dobbeltklikke på komponenten i Grasshopper [163].

6.4 Etablering av modell i Grasshopper

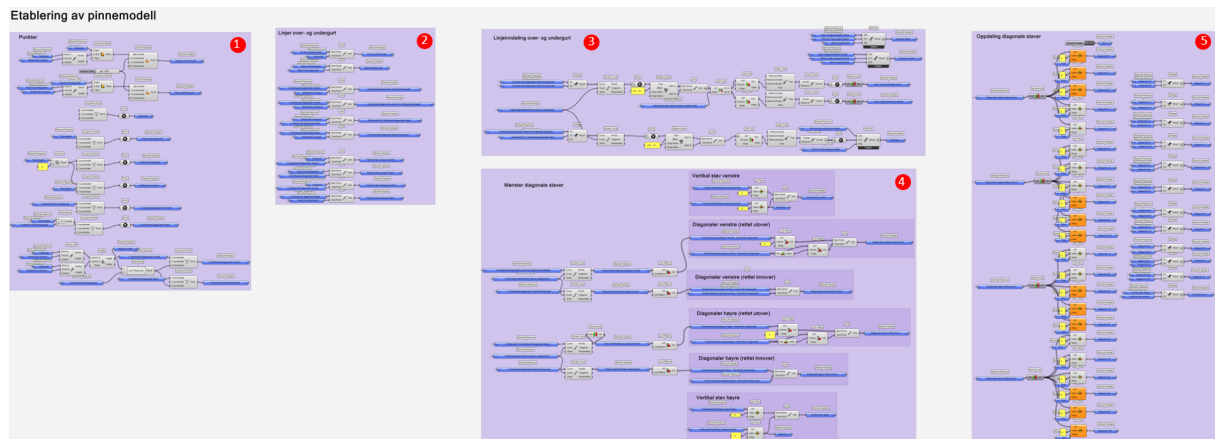
En stor del av oppgaven har bestått av å etablere limtre-fagverket i Grasshopper. Med etablering av modell menes oppbygging av limtre-fagverket på en måte som gjør det mulig å utføre belastnings-, element-, knutepunks- og optimaliseringsanalyse av fagverket. I tillegg var formålet å lage komponenter som kan gjenbrukes i andre prosjekter. Figur 6.2 viser et oversiktsbilde av den ferdige modellen, og er gitt i vedlegg 5.



Figur 6.2 Oversiktsbilde av Grasshopper modell

6.4.1 Oppbygging av pinnmodell

I førsteomgang ble fagverket bygget som en pinnmodell med punkter og linjer. Ettersom dette var første steg i programmeringsprosessen, satte det krav til mye opplæring underveis. Basert på videoer fra YouTube og utallige forsøk med prøving og feiling, endte vi med løsningen som er presentert i Figur 6.3. Oppbyggingen av pinnmodellen ble organisert i grupper med lilla farge.

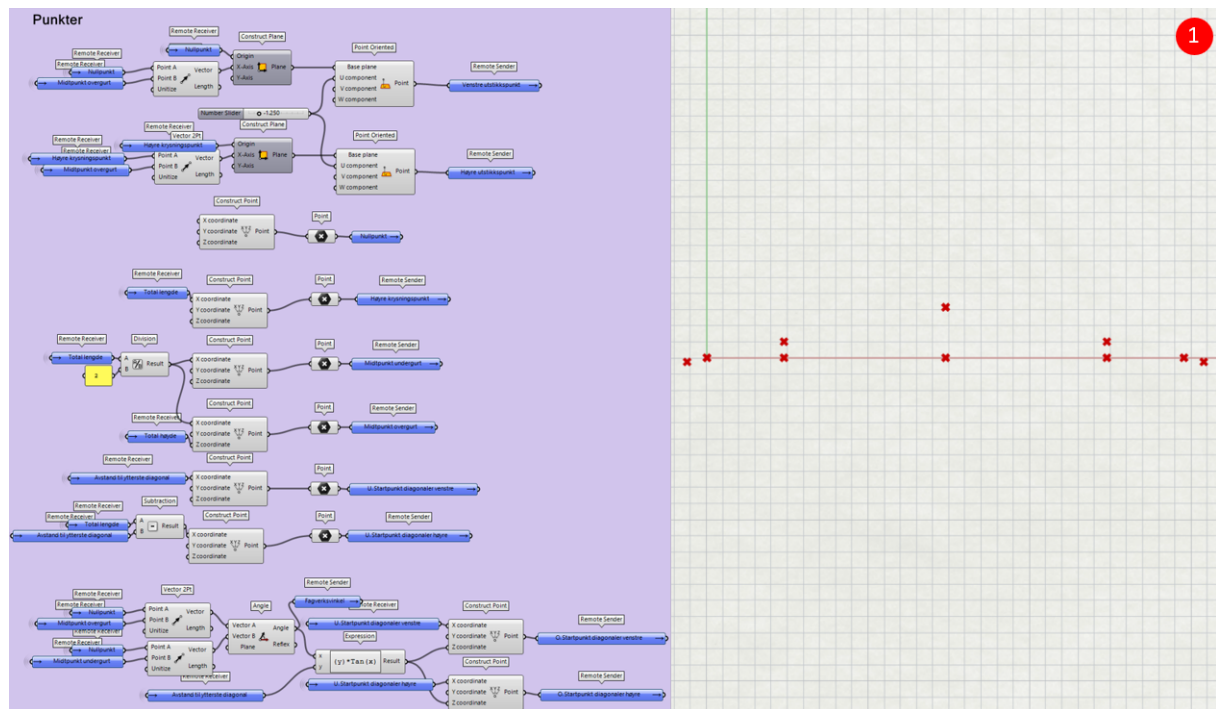


Figur 6.3 Oversiktsbilde av gruppene fra Grasshopper-modellen som etablerer pinnemodell

I gruppe nr. 1 ble hovedpunktene i modellen dannet. Figur 6.4 presenterer grasshopper-modellen av gruppen til venstre, samt Rhino-visningen av punktene til høyre. Komponenten «construct point» danner et punkt på grunnlag av valgt x-, y- og z-koordinat. Denne komponenten ble benyttet både for venstre og høyre krysningspunkt mellom over- og undergurt. Her ble venstre krysningspunkt plassert i origo $[0,0,0]$ og fikk navnet «Nullpunkt», mens «høyre krysningspunkt» ble plassert med en variabel i positiv x-retning som definerer avstanden fra origo. Videre ble den samme komponenten benyttet til å etablere midtpunktene i over- og undergurt. Disse midtpunktene er avhengig av den samme variabelen i positiv x-retning som «høyre krysningspunkt», men avstanden blir i dette tilfellet dividert med to. Samtidig har midtpunktet i overgurt en annen variabel i positiv y-retning som definerer høyden til fagverket.

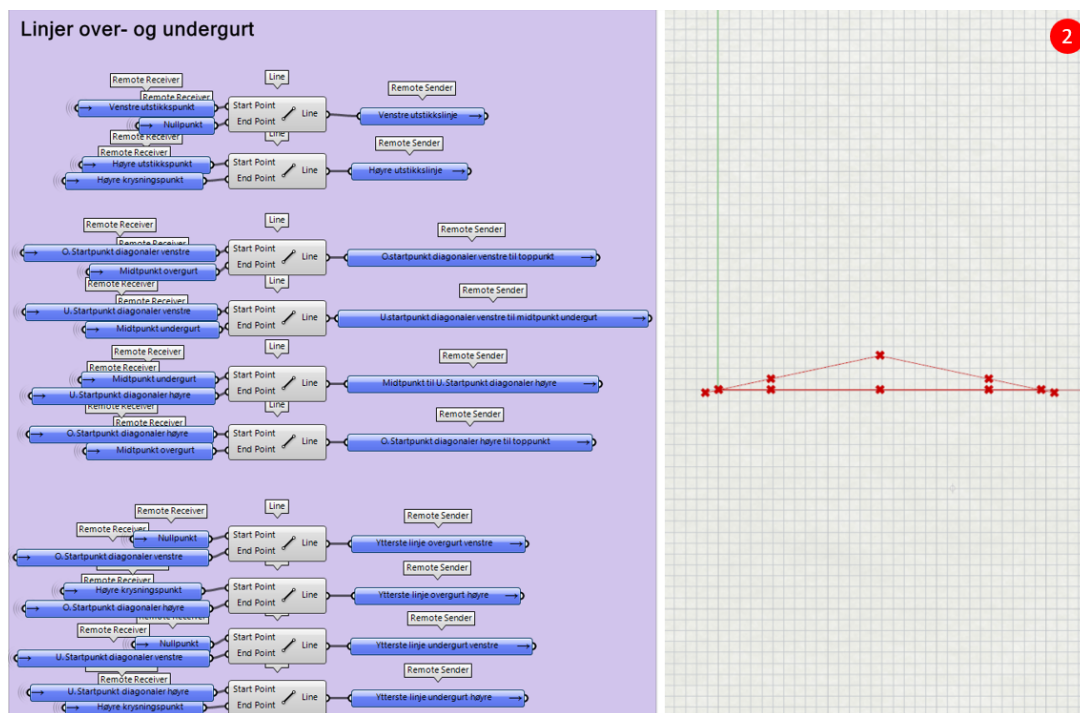
I tillegg ble «construct point»-komponenten benyttet til å etablere start- og endepunkter for fagverkets første- og siste stav. Disse punktene ble navngitt som «Startpunkt diagonaler», hvor navnet også inneholder beskrivelse av hvorvidt punktene befinner seg på høyre eller venstre side av fagverket. I tillegg inneholder navnet initialene O og U, som definerer om punktene ligger på henholdsvis over – eller undergurt. Startpunktene baseres på en tredje variabel i positiv x-retning, med navn «Avstand til ytterste diagonal». Denne definerer avstanden fra startpunktene til krysningspunktene mellom over- og undergurt. For punktene i undergurt ble dette utført på samme måte som punktene beskrevet ovenfor, mens punktene i overgurt krever litt mer. Det er ønskelig å styre startpunktene i over- og undergurt med samme variabel, men ettersom startpunktene i overgurt har en vinkel i forhold til x-retning, er det også nødvendig å definere punktets avstand i z-retning (høyde). Derfor måles fagverkets vinkel mellom to vektorer som etableres mellom nullpunktet og de to midtpunktene. Denne vinkelen benyttes sammen med variabelen «Avstand til ytterste diagonal» i den trigonometriske funksjonen for tangens, og danner lengden på vinkelens motstående katet. Denne blir deretter benyttet for å fastsette punktenes z-koordinat.

For å oppnå samme struktur som Case-fagverket er det ønskelig at overgurt springer ut over krysningspunktet mellom over- og undergurt. Her ble komponenten «construct plane» benyttet for å etablere to plan med u-komponent i overgurtens lengdeakser. I tillegg ble det spesifisert at planene skulle ha origo i fagverkets høyre- og venstre krysningspunkt. Med dette planet kunne komponenten «point oriented» benyttes for å danne punkter med en definert avstand fra krysningspunktet. Disse punktene ble navngitt: «venstre utstikkpunkt» og «høyre utstikkpunkt».



Figur 6.4 Etablering av pinnemodell del 1 – Hovedpunkter

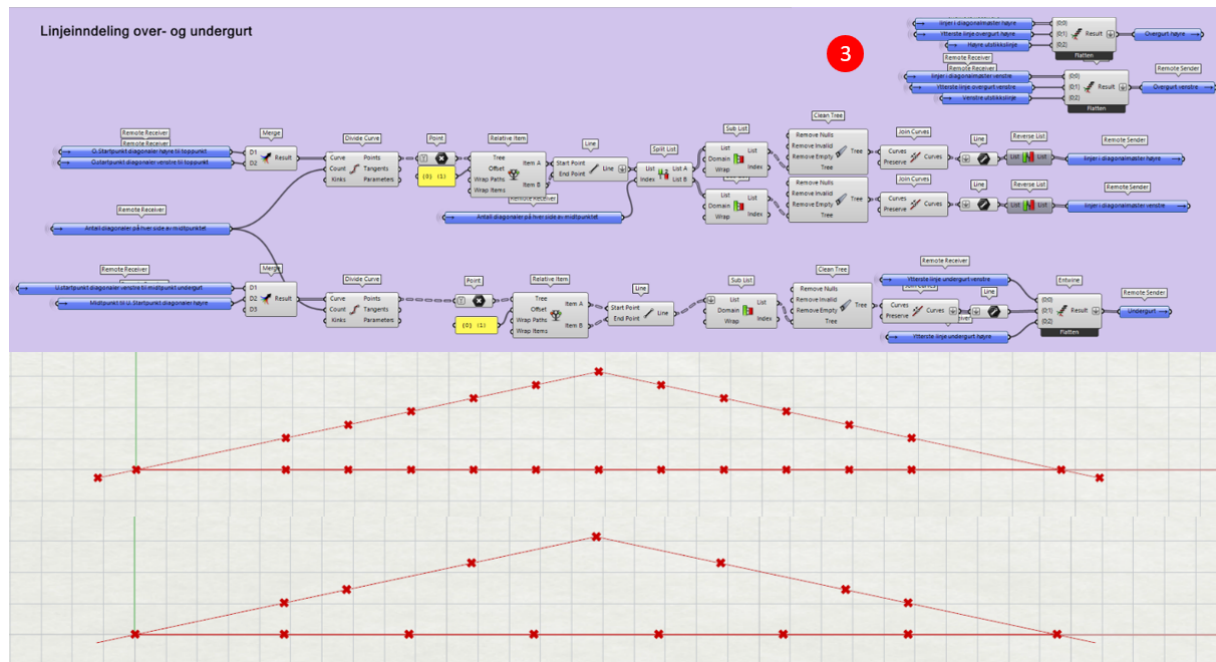
Videre benyttes punktene fra gruppe 1 til å etablere hovedlinjene i over- og undergurt (gruppe 2). Figur 6.5 presenterer grasshopper-modellens gruppe nr. 2 til venstre, samt Rhino-visningen av linjene til høyre. Punktene fra gruppe 1 ble dermed brukt som start- og endepunkt i komponenten «line», for å danne linjer i over- og undergurtens lengderetning. Linjene danner et viktig utgangspunkt for videre modellering.



Figur 6.5 Etablering av pinnemodell del 2 – Hovedlinjer

Figur 6.6 presenterer grasshopper-modellens gruppe 3, samt Rhino-visning av hvordan linjeinndelingen endres som et resultat av operasjonene som blir utført i gruppen. Formålet er å dele opp linjene mellom «startpunkt diagonaler» og midtpunktene i over- og undergurt. Inndelingen er avhengig av hvor mange diagonale staver som ønskes mellom dem. I første operasjon (lest fra figurens venstre side) samles linjene som skal deles, hvor over- og undergurt samles hver for seg. Linjene føres deretter inn i komponenten «Divide curve», som deler linjene i x-antall like deler. Antall linjeinndelinger bestemmes av en variabel som definerer antall diagonale staver på hver side av midtpunktet. For eksempel har case-fagverket fem diagonale staver på hver side, og dermed blir også linjene i over- og undergurt delt i fem like deler på hver side av midtpunktet. Alle punktene mellom de inndelte linjene blir automatisk samlet i en liste, hvor rekkefølgen følger gurtens lengdeakse. Denne punktlisten benyttes videre i komponenten «Relative item». I første output fra «Relative item» etableres en punktliste som er identisk med den som kommer inn. I neste output blir rekkefølgen derimot forskjøvet med ett punkt. På denne måten skaper komponenten start- og endepunkt for hver av de inndelte linjene, som gir muligheten for å etablere linjer mellom hver av punktene.

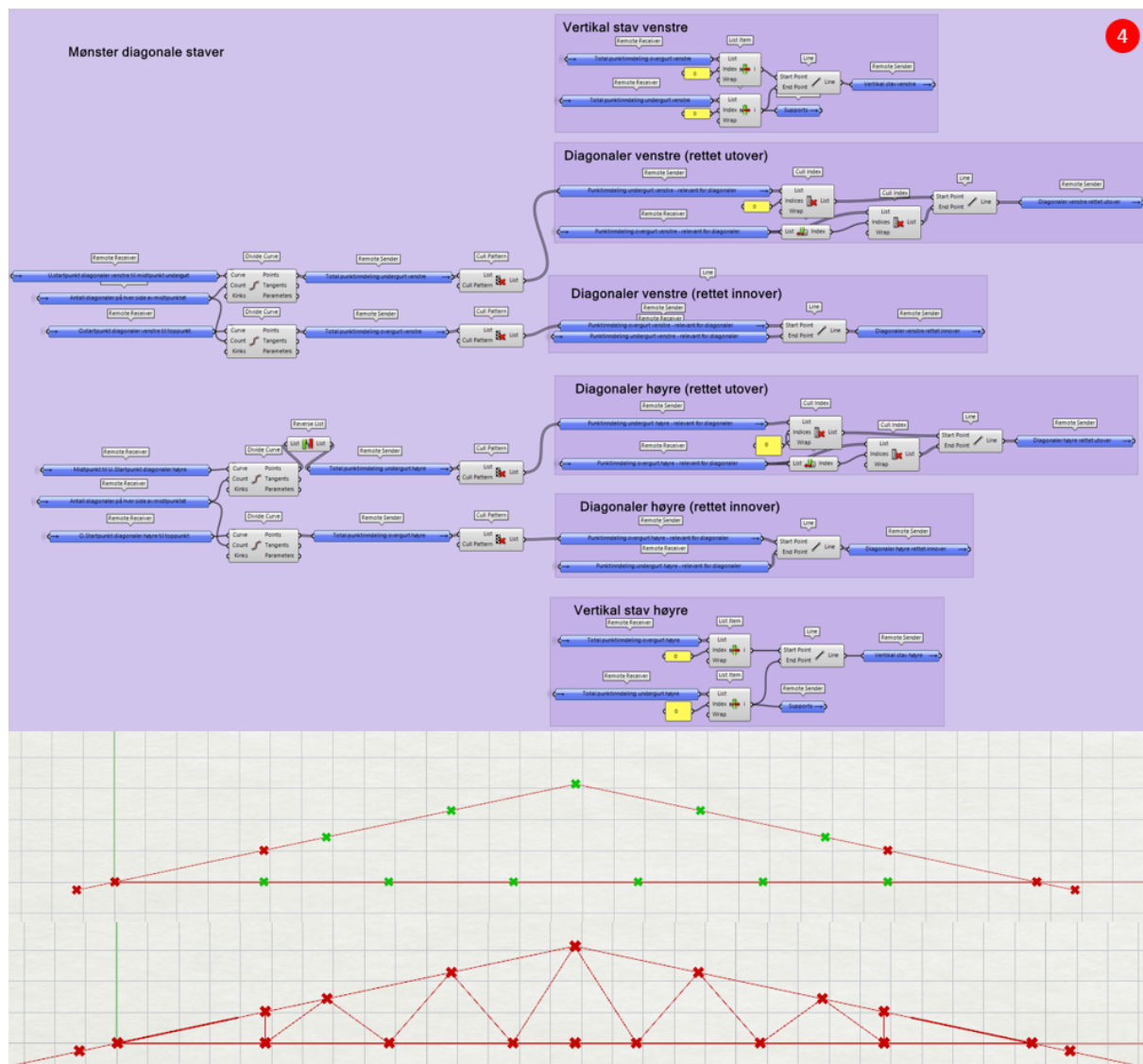
Det skal være muligheter for å legge forskjellige laster på venstre- og høyre side av overgurt, og derfor splittes denne i to deler med komponenten «split list». Også her benyttes variabelen som definerer antall staver på hver side av midtpunktet, slik at listen deles på riktig sted dersom variabelen endres. På dette stadiet fungerer linjeinndelingen som vist i den øverste Rhino-visningen i Figur 6.6. Videre skal linjene mellom stavene som er diagonale kobles sammen, mens de ytterste linjene (ut mot vertikal stav) skal være som de er. Derfor benyttes «sub list» for å slå sammen linjene som ønsket, og dermed «join curve». Etersom linjeinndelingen fra «sub list» er tilpasset for å kunne øke antall diagonaler, etableres noen «tomme» grener i komponentens output. Derfor benyttes komponenten «clean tree» for å fjerne disse grenene. Med dette fungerer listeinndelingen som vist i den nederste Rhino-visningen i Figur 6.6. Til slutt vendes rekkefølgen til linje-listene i overgurtene slik at første linje alltid starter ved toppunktet, mens linjene i undergurtene følger fra venstre mot høyre. Gruppen inneholder i tillegg sammenslåing av alle linjene i hver av gurtene, inkludert utstikkslinjer og linjene før- og etter stavmønster.



Figur 6.6 Etablering av pinnmodell del 3 – Linjeinndeling i over- og undergurt

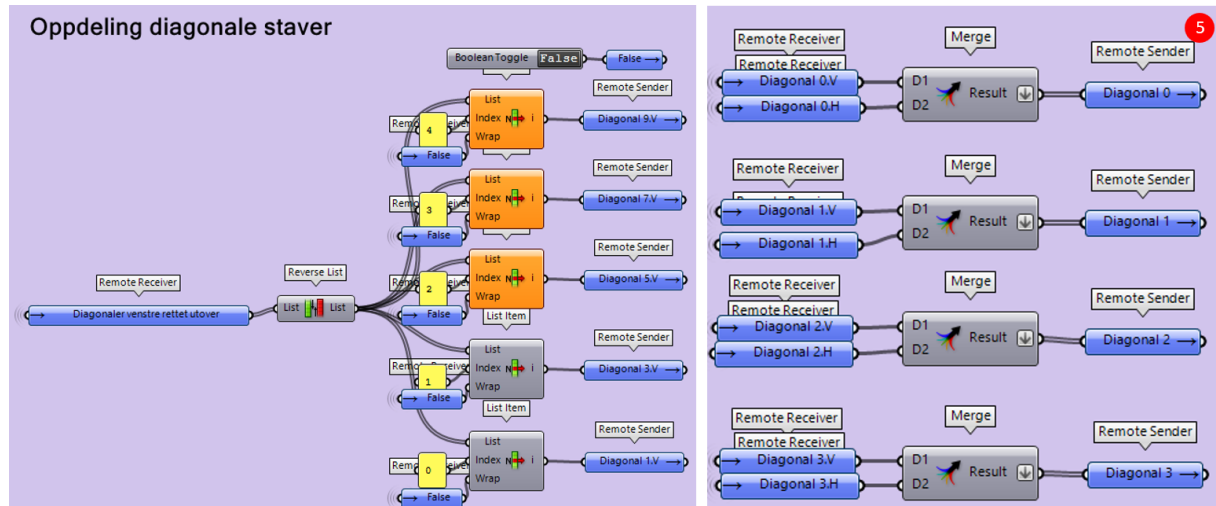
I gruppe 4 ble hovedlinjene fra gruppe 2 benyttet til å etablere stavene mellom over- og undergurt i ønsket mønster. Komponenten «divide curve» benyttes på akkurat samme måte som i gruppe tre, for å dele opp linjene mellom «startpunkt diagonaler» og midtpunktene i over- og undergurt. Deretter benyttes komponenten «cull pattern» for å sortere ut punktene hvor det er ønskelig at stavene skal treffe gurtene. Mønsteret settes ved kommandoene «True» og «False», hvor «True» beholder punktet i listen og «False» fjerner. Alle punktlistene starter ved fagverkets ytterkant og følger deretter innover mot midtpunktene. I overgurten benyttes derfor mønsteret «False, True, False, True», mens det i undergurten benyttes «True, False, True, False». På denne måten blir punktene sortert ut som vist i Figur 6.7s øverste Rhino-visning.

Videre benyttes de sorterte punktene til å etablere stavmønsteret. Det å danne stavene som er rettet innover, gjøres rett frem ved å velge punktinnndelingen i overgurt som linjens startpunkt og punktinnndelingen i undergurt som linjes endepunkt. Dette gjelder både høyre og venstre side av fagverket. Videre dannes stavene som er rettet utover ved å fjerne første punktet fra undergurten og siste punktet fra overgurten, og dermed velges punktene som gjenstår som linjenes start- og endepunkter. Ved etablering av vertikale staver benyttes punktlistene som ble dannet før utsorteringen ved «cull pattern», som inneholder alle punktene fra linjedelingen. Her velges første punkt fra punkt i overgurt som linjens startpunkt og første punkt i undergurt som linjes endepunkt. Med dette fungerer modellen som vist i den nederste Rhino-visningen i Figur 6.7.



Figur 6.7 Etablering av pinnemodell del 4 – Staver i mønster

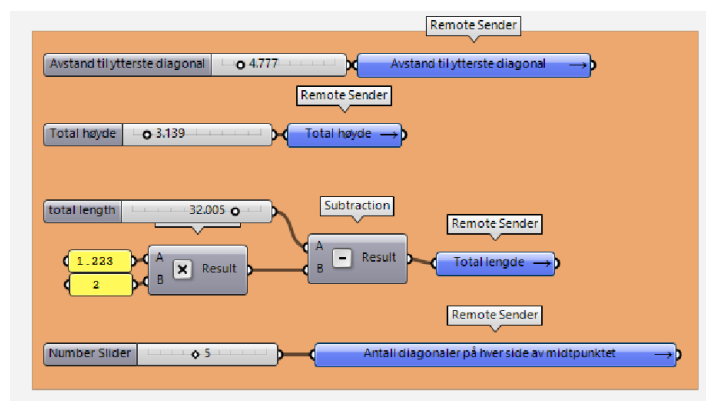
Til slutt blir stavene sortert i riktig rekkefølge og navngitt i gruppe 5. Denne gruppen er presentert i Figur 6.8. Her benyttes komponenten «list item» til å velge ut ett etter ett element fra listene. Første diagonal fra midten av fagverket benevnes «Diagonal 0», og deretter følger «Diagonal 1», «Diagonal 2» osv. utover mot fagverkets ytterkant. Ettersom systemet skal akseptere flere diagonaler i mønsteret, er dette sorterings- og navngivningssystemet etablert for inntil 11 diagonaler på hver av sidene. Dersom variabelen for antall diagonaler er satt til f.eks. 5 (som i figurene ovenfor), vil ikke systemet for de resterende diagonalene fungere. Komponentene vil da få oransje farge og ikke føre informasjon videre i systemet. For at systemet skal fungere som dette, er det viktig at inputen «Wrap» bli satt til «false» i alle komponentene. Dersom det f.eks. er to elementer i listen med indeks 0 og 1 når «Wrap» er satt til «true», vil indeks nr. 2 hente ut elementet med indeks 0. Grunnen til dette er at elementene i listen gjentas uendelig mange ganger. Siste operasjon i gruppe fem er at elementene som er like på hver side av fagverket samles, og benevnes med like navn.



Figur 6.8 Etablering av pinnemodell del 5 – Sortering og navngivning av staver

6.4.2 Styring av parametere

Komponentene som styrer variablene i fagverket er samlet i den oransje gruppen, som illustrert i Figur 6.9. Dette gir bedre oversikt over mulighetene for variasjon av fagverkets geometri. Først og fremst er det mulig å justere avstanden fra opplegget under de vertikale staverne til krysningspunktet mellom over- og undergurt. Videre kan både total høyde- og total lengde på fagverket justeres. «Total lengde» justerer avstanden fra krysningspunktet mellom over- og undergurt på venstre side til krysningspunktet mellom over- og undergurt på høyre side. I case-fagverket er den totale avstanden mellom utstikkslinjens ytterste punkter satt til 32,005 m. Derfor er denne avstanden subtrahert med avstandene til utstikkslinjene. I tillegg kan antallet diagonaler på hver side av midtpunktet justeres, som tidligere nevnt.

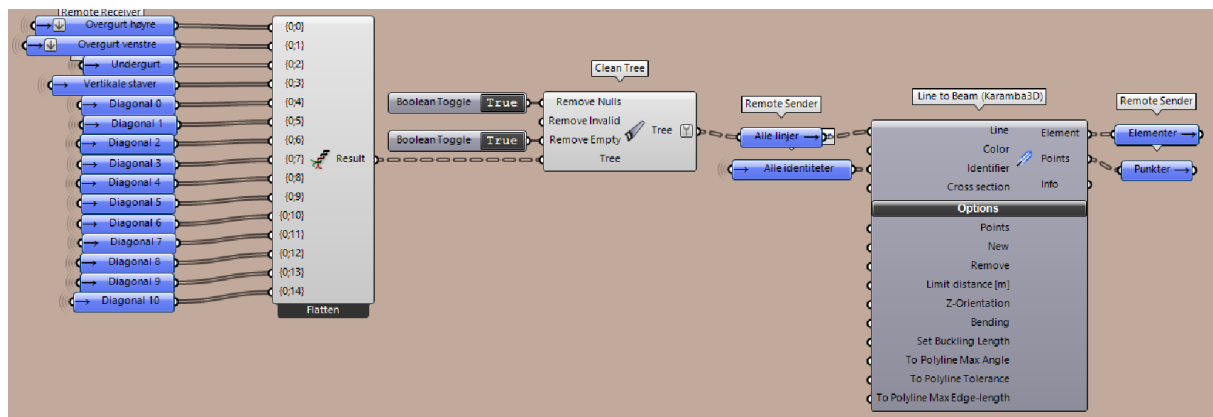


Figur 6.9 Styring av parametere

6.4.3 Fra linje- til bjelkeelementer

Videre skal elementene klargjøres for å kunne analyseres i FEM-analyse med plug-in «Karamba 3D». I første steg omgjøres linjene fra forrige kapittel til bjelke-elementer. Denne gruppen er markert med brun farge i modellen, som vist i Figur 6.10. Alle elementer samles med komponenten «Entwine» som danner grener for hver av inputene. Dersom systemet er satt til å inneholde 5 diagonaler på hver side av midtpunktet, vil første- og andre gren inneholde fem overgurt-linjer. Videre vil tredje

gren inneholde 7 undergurts-linjer, mens de resterende grenene inneholder to linjer hver (like staver fra venstre og høyre side av fagverket). Fagverket vil i dette tilfellet kun fylle ut grenene med linjeinformasjon frem til «Diagonal 4», mens de resterende grenene blir fylt med «tomme» diagonaler. Derfor benyttes komponenten «clean tree» for å fjerne grenene uten informasjon. Listen med alle linjer blir deretter ført videre inn i Karamba-komponenten «Line to Beam».



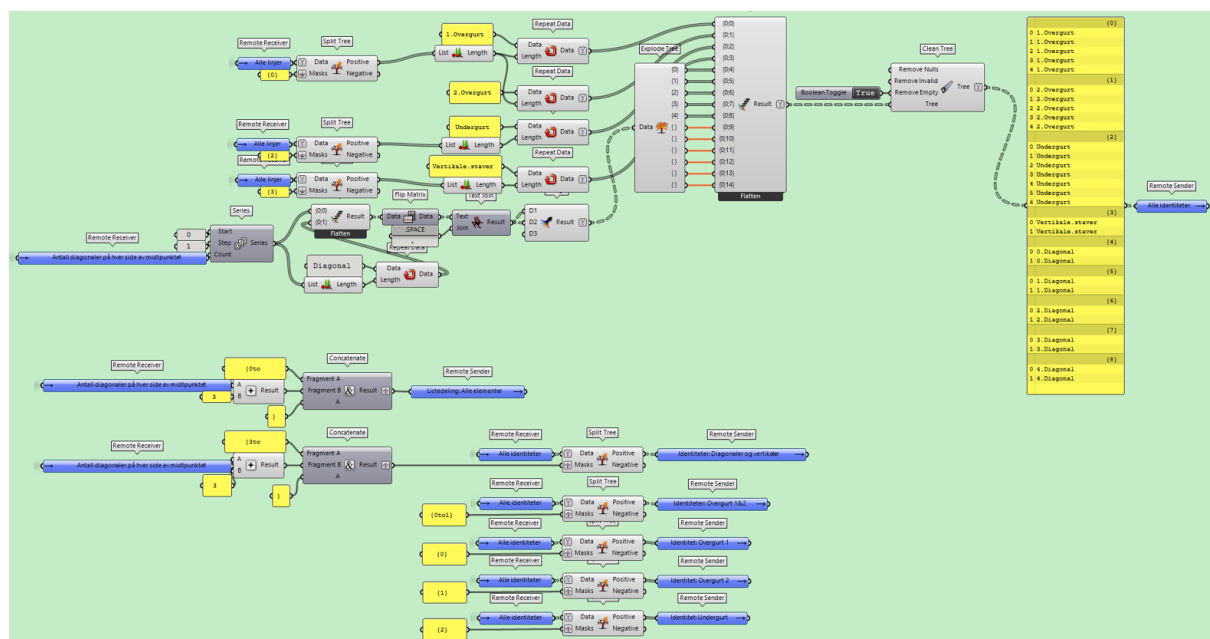
Figur 6.10 Fagverkets linjer omgjøres til bjelke-elementer

For at FEM-analysen skal fungere, er det svært viktig at elementene får tildelt en identitetsliste som har identisk oppbygging som bjelke-elementlisten. Derfor er det bygget et system som etablerer identitetslister som forandres etter hvert som antall diagonaler endres. Systemet er markert med grønn farge i modellen, som vist i Figur 6.11. Høyre- og venstre side av overgurtene, navneses henholdsvis som «1.Overgurt» og «2.Overgurt». Hver av overgurtssidene, får tildelt hver sin gren i identitetslisten. Antallet linjer i overgurtene endres etter hvert som antall diagonaler endres (som forklart i kapittel 6.4.1 gruppe 3). Derfor repeteres identitetsnavnet i treet basert på antall linjer i hver av gurtene. Dette utføres med komponentene «split list» og «list length». Som omtalt ovenfor, er det etablert en samling av alle linjer med grener for hver del. Denne er navngitt «Alle linjer», og benyttes i «split list»-komponenten for å hente informasjon om antall linjer i hver av grenene. Linjene i høyre- og venstre overgurt vil alltid befinne seg i henholdsvis nulte- og første gren, og derfor vil komponenten «list length» alltid observere riktig antall linjer. På samme måte er linjene i undergurt og vertikale staver alltid i henholdsvis andre og tredje gren, og kan dermed identifiseres på samme måte som overgurtene.

For de diagonale stavene er systemet noe mer komplisert, da antallet grener i listen endres ved variasjon i antall staver. Det er ønskelig å identifisere stavene som «0.Diagonal», «1.Diagonal», «2.Diagonal» og så videre. Derfor etableres først en serie med tall som starter ved null, og lengden på serien baseres på variabelen som bestemmer antall staver. Tallserien blir så ført videre inn i «Entwine»-komponenten og utgjør første gren i denne. I den andre grenen etableres en liste med teksten «Diagonal», som blir repetert like mange ganger som tallserien. Deretter benyttes komponenten «flip matrix», som henter informasjon fra første indeks i hver av grenene og danner en ny gren. Det samme blir gjort med de resterende indeksene. Med dette samles tallet 0 og teksten «Diagonal» i første gren, tallet 1 og teksten «Diagonal» i andre gren og så videre. Deretter benyttes komponenten «Text join», for å binde sammen de to indeksene i hver gren. I tillegg fastsettes det at bindeleddet mellom indeksene skal være et punktum. Med dette inneholder første gren

«0.Diagonal», andre gren «1.Diagonal» og så videre. Men ettersom hver av disse identitetene representerer to linjer (både fra høyre- og venstre side av fagverket), benyttes komponenten «merge» for å repetere innholdet i hver av grenene én gang. Til slutt benyttes «explode tree» for å dele opp grenene hver for seg. Siden det er mulighet for å benytte inntil 11 diagonaler, er det like mange utganger fra denne komponenten. Dersom et mindre antall diagonaler blir benyttet, sendes dette over som «tomme grener», og fjernes etterpå med komponenten «clean tree».

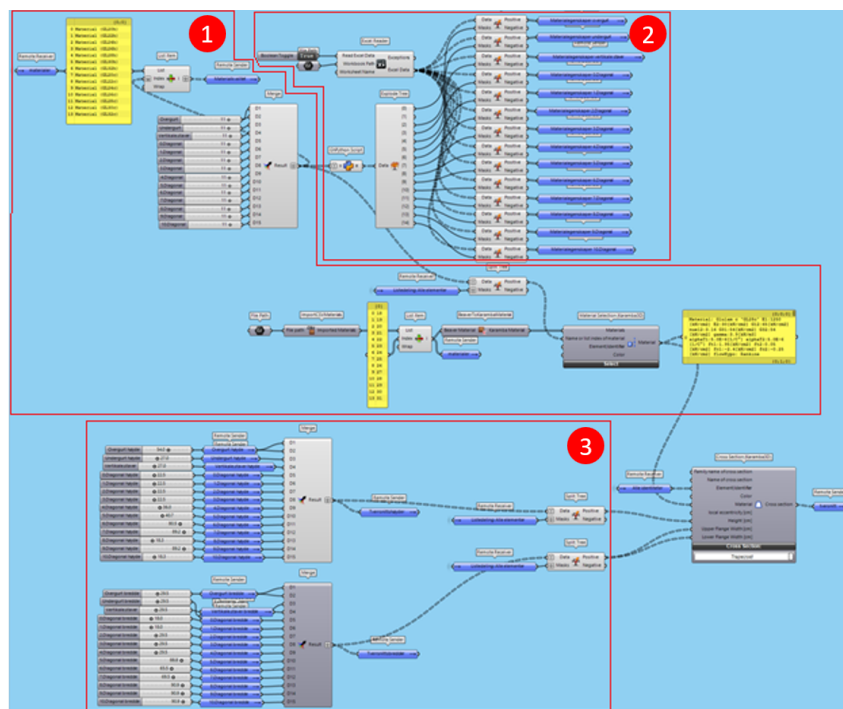
I nederste del av gruppen ble det dannet et system for å hente ut deler av identitetene. Dette blir benyttet videre i modellen dersom det f.eks. bare skal benyttes identitetene for diagonaler og vertikaler, eller bare identitetene for høyre- og venstre overgurt. Her benyttes komponenten «split list», hvor man bestemmer grenen/grenene hvor man ønsker å hente identiteter fra. For å hente ut identitetene til bare de diagonale stavene, er komponenten «Concatenate» benyttet for å beskrive hvilke grener som skal hentes ut. Denne binder sammen tekst, og gir derfor rom for å legge inn variasjonen i antall diagonaler. Diagonalene vil alltid starte ved gren nr. 3, og derfor er denne satt som fast verdi. Deretter bestemmes siste gren ved å benytte variabelen for antall diagonaler og addere denne med 3. På denne måten vil identitetslisten oppdateres ved variasjon i antall diagonaler.



Figur 6.11 Alle identiteter og inndeling av identiteter

6.4.4 Material- og tverrsnittsegenskaper

Videre må informasjon om material- og tverrsnittsegenskaper oppgis, for at FEM-analysen i Karamba skal gi riktige resultater. Denne gruppen er markert med blå farge i modellen, og er vist i Figur 6.12. Karamba-komponenten «Cross Section» benyttes for å hente inn all informasjon om både tverrsnitt og materiale. Denne gir muligheter for å velge forskjellige tverrsnitt, hvor «Trapezoid» egner seg for rektangulært tverrsnitt. Egenskapene blir tilknyttet hvert element ved hjelp av identitetslisten som ble beskrevet i forrige kapittel. Derfor er det viktig at listene med egenskaper har identisk oppbygging som identitetslisten.

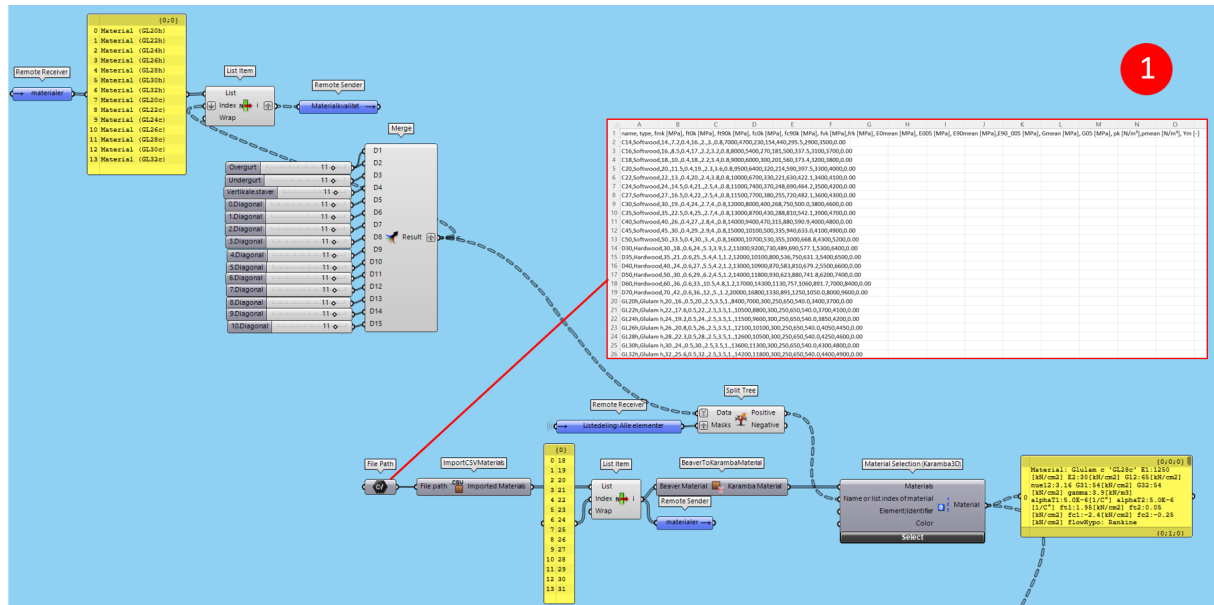


Figur 6.12 Oversiktsbilde for definering av material- og tverrsnittsegenskaper

I den blå gruppens første inndeling, som vist i Figur 6.13, hentes materialene for hvert av elementene som skal benyttes videre i FEM-analysen. Materialene hentes inn via en Karamba komponent med navn «Material Selection». Denne gir mulighet for å velge materiale fra Karambas bibliotek, og i tillegg gir den mulighet for å definere materialtypen selv. Vi har valgt å definere materialtypen selv, ved å benytte en metode fra Beaver-plugin. Grunnen for dette metodevalget vil bli diskutert i kapittel 8.

Beaver har dannet en CSV fil med materialegenskaper for både limtre og standard konstruksjonsvirke. Et utklipp fra CSV filen er presentert i Figur 6.13, og ligger også vedlagt som «Vedlegg 1». Ved hjelp av Beaver-komponenten «ImportCSVMaterials» hentes materialene inn i Grasshopper-modellen, og etablerer en liste med alle materialtypene. I dette tilfellet er det bare ønskelig å benytte limtre, og derfor hentet vi ut limtrematerialene (indeks nr. 18 til nr. 31) ved bruk av «list item». Deretter benyttes Beaver-komponenten «BeaverToKarambaMaterial» for å overføre materialegenskapene til riktig Karamba-format.

Ønsket materiale for hvert av elementene samles i en liste. Her er outputen fra «ImportCSVMaterials» benyttet for å hente ut en oversikt over hvilken materialtype som tilsvarer hvilken indeks. I dette tilfellet skal GL28c benyttes for alle elementene, og derfor benyttes indeks nr. 11 for alle. Systemet er laget for inntil 10 diagonaler, og derfor må listen justeres etter hvert som antall diagonaler endres. Dette gjøres ved å benytte listeinndelingen fra forrige kapittel, for å splitte listen ved riktig diagonal. På denne måten samsvarer listeinndelingen med identitetslisten som går inn i «Cross section»-komponenten.



Figur 6.13 Definerings av materialtype ved bruk av Beaver-komponenter

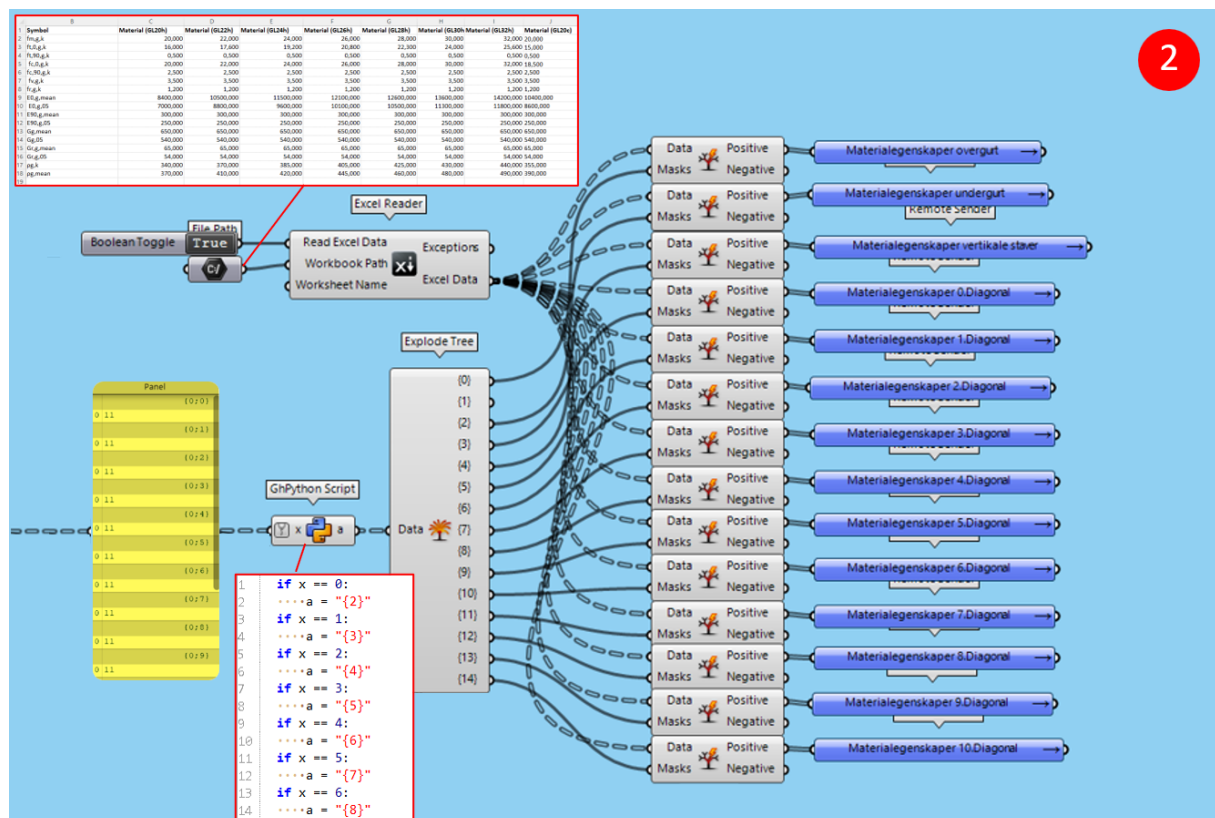
Tabell 6.1 presenterer en oversikt over fasthet- og stivhetsdata som blir definert i CSV-filen fra Beaver, samt hvilke av dataene som blir overført videre til Karamba. Materialtypen GL28c brukes som eksempel. Ikke alle verdier blir overført videre til Karamba, og det er heller ikke alle verdiene som er riktige. Verdien for rulleskjær er sannsynligvis satt til det doble av strekkfastheten normalt på fiberretning $f_{t,90,k}$ (jf. EC5 (punkt 6.1.7) [22]). Den behøves imidlertid ikke å rettes på i henhold til riktige tabellverdier for limtre, da den uansett ikke brukes videre. I tillegg defineres en koeffisient for termisk utvidelse, uten at denne informasjonen er hentet fra CSV-filen til Beaver. Verdien vil likevel ikke bli benyttet dersom temperaturen ikke blir spesifisert [49]. Derfor kan man se bort fra denne verdien, og heller ta hensyn til termisk utvidelse i eget system for utnyttelse i henhold til eurokode.

Tabell 6.1 overføring av fasthet- og stivhetsdata fra Beaver til Karamba

	FASTHET- OG STIVHETS-DATA DEFINERT AV BEAVER		FASTHET- OG STIVHETS-DATA OVERFØRT VIDERE TIL KARAMBA	
Materialnavn	GL28c		GL28c	
Bøyefasthet	$f_{m,k}$	28 MPa		
Strekfasthet	$f_{t,0,k}$	19.5 MPa	1,95 kN/cm ²	f_{t1} Tensile strength dir 1
	$f_{t,90,k}$	0.5 MPa	0,05 kN/cm ²	f_{t2} Tensile strength dir 2
Trykkfasthet	$f_{c,0,k}$	24 MPa	2,4 kN/cm ²	f_{c1} Compressive strength dir 1
	$f_{c,90,k}$	2.5 MPa	0,25 kN/cm ²	f_{c2} Compressive strength dir 2
Skjærfasthet (skjær og torsjon)	$f_{v,k}$	3.5 MPa	Karamba har en inngang for denne « t_{12} », men den blir likevel ikke overført	
Rulleskjærfasthet	$f_{r,k}$	1 MPa (bør være 1,2)		
Elastisitetsmoduler	$E_{0,mean}$	12500 MPa	1250 kN/cm ²	E_1 Young's modulus dir1
	$E_{0,05}$	10400 MPa		
	$E_{90,mean}$	300 MPa	30 kN/cm ²	E_2 Young's modulus dir2
	$E_{90,05}$	250 MPa		
Skjærmoduler	G_{mean}	650 MPa	65 kN/cm ²	G_{12} In plane shear modulus
	G_{05}	540.0 MPa	54 kN/cm ²	G_{31} transverse shear modulus dir1
G_{32} transverse shear modulus dir2				
Poisson's ratio			$\frac{E_1}{2 G_{12}} - \sqrt{\frac{E_2}{E_1}} = 3,16$	ν_{e12} in – plane lateral contraction coefficient
Densiteter	ρ_k	3900	3,9 kN/cm ²	Γ spesific weight
	ρ_{mean}	4200 N/m ³		
	Y_m [-]	0.00		
Termisk utvidelse			$5 * 10^{-6}$ [1/C°]	Alpha T ₁ coefficient of thermal expansion dir 1
			$5 * 10^{-6}$ [1/C°]	Alpha T ₂ coefficient of thermal expansion dir 2

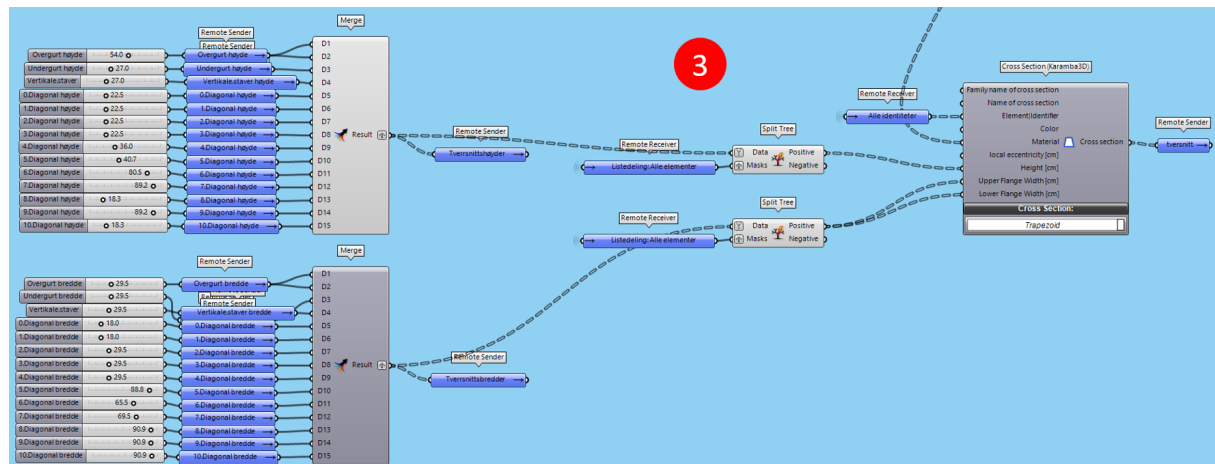
Ved innhenting av materialer med Beaver som forklart ovenfor, vil en ikke ha tilgang til å bruke de konkrete Fasthet- og stivhetsdataene som trengs for kontroll av elementer og forbindelser i henhold til Eurokode. Det vil kun være de omgjorte egenskapene som er tilgjengelig, som vist til høyre i

Tabell 6.1. Derfor er det etablert et eget system som gir mulighet for å hente inn verdier som $f_{m,k}$, $f_{t,0,k}$, $f_{t,90,k}$ og så videre. Dette blir utført i den blå gruppens andre inndeling, som vist i Figur 6.14 Innhenting av materialeegenskaper for Eurokode-beregninger. Materialtabeller fra limtreboka ble derfor ført inn i et Excel-ark (se Vedlegg 2), og hentet inn til Grasshopper ved hjelp av komponenten «Excel Reader». Komponentene etablerer en liste med alle egenskaper, hvor materialtypene blir representert i hver sin gren. Derfor må det hentes ut riktig gren, basert på hvilket materiale som velges i Beaver-materialsystemet. For dette, er det laget et python-script, som oppgir hvilken gren som må hentes ut når et spesifikt materiale er valgt i Beaver-systemet. For eksempel vil tallet to i Beaver-systemet tilsvare materialtypen GL24h, og materialeegenskapene for GL24h ligger i gren nummer fire. Dermed benyttes «split tree» for å hente ut gren nummer fire fra listen. Systemet gir dermed fremdeles mulighet til å benytte forskjellig materialtype i de ulike elementene, og oppdateres automatisk etter hvert som det gjøres endringer i Beaver-systemet.



Figur 6.14 Innhenting av materialeegenskaper for Eurokode-beregninger

Nederste del av gruppen (del 3) definerer elementenes høyde og bredde, som vist i Figur 6.15. Elementenes høyder samles i en liste, mens elementenes bredder samles i en annen. Systemet er laget for inntil 10 diagonaler, og derfor må listen justeres etter hvert som antall diagonaler endres. Dette gjøres ved å benytte listeinndelingen fra forrige kapittel, for å splitte listen ved riktig diagonal. På denne måten samsvarer listeinndelingen med identitetslisten som går inn i «Cross section»-komponenten. I tillegg må elementenes bredde være input for både «Upper Flange Width» og «Lower Flange Width», ettersom elementene har et rektangulært tverrsnitt (ikke trapes).



Figur 6.15 Definerings av tverrsnitts-dimensjoner

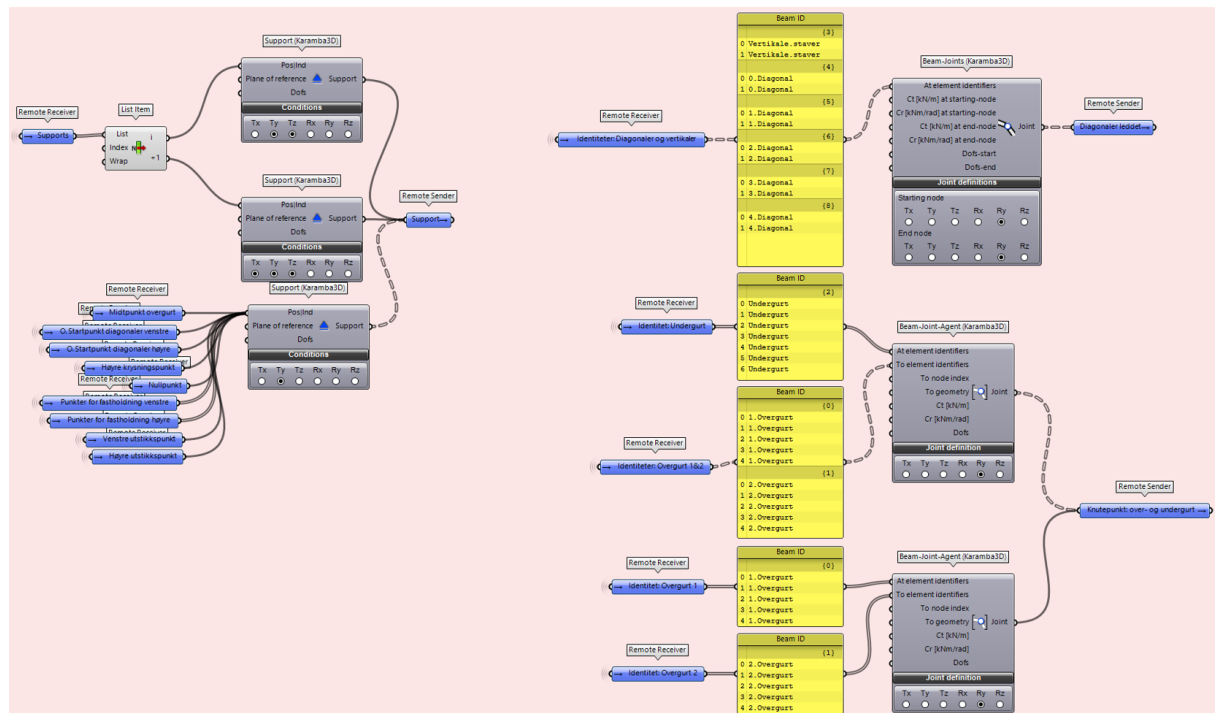
6.4.5 Opplegg og forbindelser

Systemet for opplegg og forbindelser er markert med rosa farge i modellen, som vist i Figur 6.16. Strukturen har seks frihetsgrader (DOFs) i et tredimensjonalt rom. Den kan bevege seg i tre translasjoner (rettlinjet uten rotasjon) langs x-, y-, og z-aksen. I tillegg kan strukturen roteres om de samme aksene [49]. Systemet etablerer oppleggene som vist til venstre i Figur 6.16. «Support»-komponentene vil automatisk ta utgangspunkt i det globale koordinatsystemet, om ikke annet spesifiseres. Komponentens tre første avkryssingssirkler bestemmer fastholdning for translasjonene, mens de tre siste bestemmer fastholdning for rotasjonene. Når sirklene er avkrysset, blir strukturen fastholdt og frihetsgraden settes til null. Dermed fremkalles en reaksjonskraft for å motstå bevegelsen i den spesifikke retningen [49].

Oppleggene er plassert under hver av de vertikale staver, og dannes ved hjelp av punkter som allerede er etablert ved oppbygging av pinnemodell. Strukturen skal være fritt opplagt, og derfor fastholdes høyre opplegg mot translasjon i y- og z-retning. Hvorimot venstre opplegg fastholdes for translasjon i alle retninger. I tillegg er det satt fastholdning for translasjon ut av planet i alle overgurtens punkter. Dette er en forenkling, ettersom fastholdningen i realiteten er noe tettere.

Videre må det spesifiseres om forbindelsene skal være leddet eller momentstive. Dette bestemmes ved bruk av «Beam-Joints»- og «Beam-Joint-Agent»-komponentene til Karamba. Forbindelsene tilknyttes ønsket element ved hjelp av identitetslistene som ble beskrevet i kapittel 6.4.3. Øverst til høyre i Figur 6.16, defineres forbindelsene for diagonalene. Alle diagonale staver blir regnet som leddet om y-aksen. Derfor fylles sirklene for rotasjon om y-aksen (R_y). For disse benyttes «Beam-Joints»-komponenten, da denne spesifiserer fastholdning for hver av endepunktene til staverne.

Det samme utføres for knutepunktet i toppunktet og i knutepunktene mellom høyre- og venstre overgurt, da disse også skal regnes som leddet. Disse knutepunktene defineres i de to komponentene nederst til høyre i Figur 6.16. For disse trenger man imidlertid bare å spesifisere fastholdning for en av elementets ender, og derfor benyttes «Beam-Joint-Agent» istedenfor «Beam-Joints».



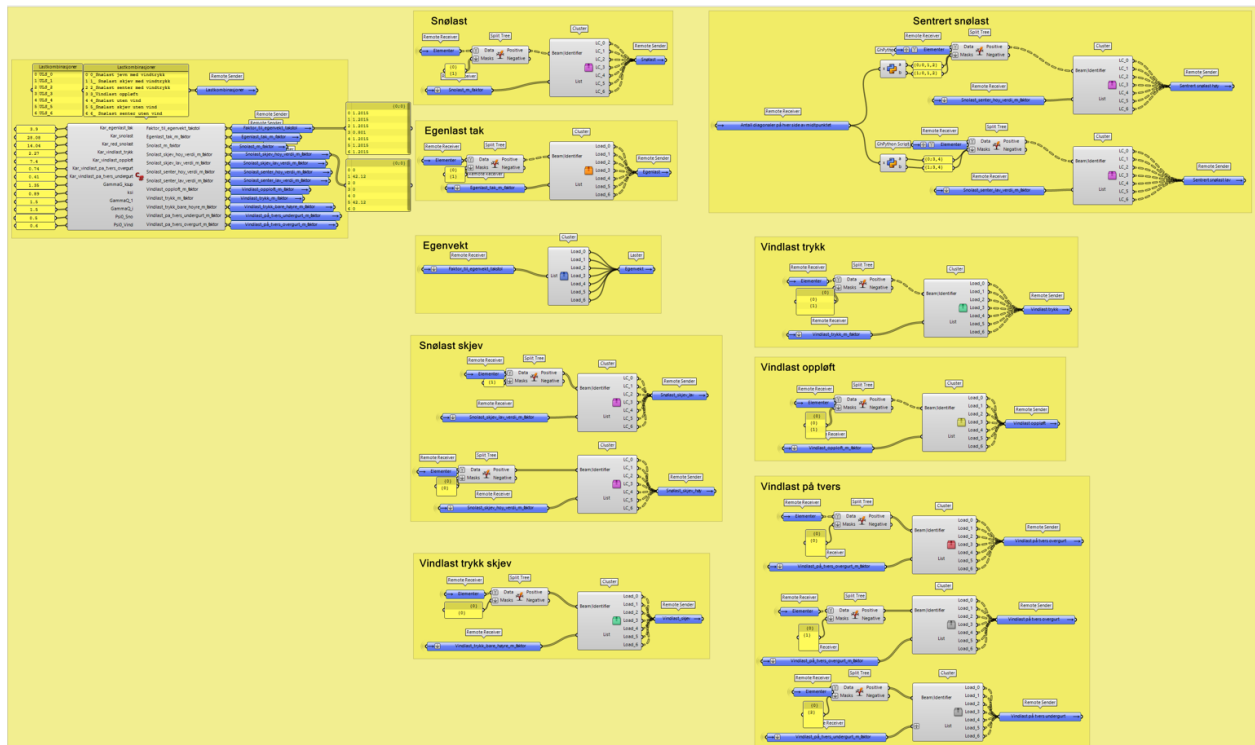
Figur 6.16 Oversiktsbilde av gruppen fra Grasshopper-modellen som etablerer opplegg- og forbindelser

6.4.6 Laster

Som vist i Figur 6.17, er systemet for laster- og lastkombinasjoner samlet i en gruppe som er markert med gul farge. Til venstre i figuren er det laget et script som danner lister for hver av de ulike last-typene (Jevn snølast, egenlast tak, egenvekt osv.). Listene inneholder sju indekser, som tilhører de sju forskjellige lastkombinasjonene som ønskes for systemet. Hver indeks fra listen inneholder lastens karakteristiske størrelse multiplisert med tilhørende lastfaktor. Dersom f.eks. jevn snølast ikke involveres i lastkombinasjon to, vil indeks to i listen tilsvare tallet null. Denne metoden er hentet fra en tidligere masteroppgave fra NTNU med navn «Optimalisering av fagverksbuer for store frispennkonstruksjoner» [4].

Videre føres listene med krefter og lastfaktor til Karambas «Beam load»-komponent for hver av de ulike last-typene. Her spesifiseres også retning, kraft, last-type og plassering i forhold til bjelke-elementene. Dette utføres i clusterene, fordi hver av lastkombinasjonene må håndteres hver for seg. Etttersom retning og last-type er forskjellig for de ulike lastene, er det benyttet forskjellige clusterene, hvor clusterene med lik oppbygging har samme farge på cluster-ikonet.

Videre i kapittelet presenteres metoden for å bygge opp scriptet først, og deretter metoden for å bygge opp de ulike clusterene.



Figur 6.17 Oversiktsbilde av gruppene fra Grasshopper-modellen som etablerer laster- og lastkombinasjoner

Script for laster- og lastkombinasjoner

Det er tatt utgangspunkt i lastene- og lastkombinasjonene som er benyttet for case-fagverket, og det er derfor ikke utført egne analyser for hvilke lastkombinasjoner som vil være dimensjonerende. Alle lastkombinasjonene, bortsett fra lastkombinasjon nummer tre, er hentet fra ligning 6.10.b i Tabell NA.A1.2(B), NS-EN-1990: Grunnlag for prosjektering av konstruksjoner [164] Ligning 6.10.b er presentert i formel 6.1:

$$\sum_1 \xi \cdot \gamma G_{j,sup} \cdot G_{kj,sup} + \sum_2 \gamma Q_1 \cdot Q_{k1} + \sum_3 \gamma Q_i \cdot \psi_{0i} \cdot Q_{ki} \quad (6.1)$$

Hvor Σ_1 benyttes for permanente laster, Σ_2 for den dominerende variabel, Σ_3 for øvrige variabler. Den permanente lasten settes som ugunstig, for å finne det mest kritiske tilfellet. I henhold til NA.A1.2(B) [164] settes dermed:

- $\xi = 0,89$
- $\gamma G_{j,sup} = 1,35$
- γQ_1 og $\gamma Q_i = 1,5$
- $\psi_{0,vind} = 0,6$.

For lastkombinasjon nummer tre, benyttes derimot ligning 6.10 i Tabell NA.A1.2(A), NS-EN-1990: Grunnlag for prosjektering av konstruksjoner [164]. Her velges permanent last som gunstig, fordi man ønsker at minst mulig last legger vekt på konstruksjonen når man undersøker den mest kritiske kombinasjonen i motsatt retning (vindlast oppløft). I tillegg er både vindlast oppløft og vindlast på tvers satt som dominerende variabel last. Ligning 6.10.a er presentert i formel 6.2:

$$\sum_1 \gamma G_{j,inf} \cdot G_{kj,inf} + \sum_2 \gamma Q_1 \cdot Q_{k1} + \sum_3 \gamma Q_i \cdot \psi_{0i} \cdot Q_{ki} \quad (6.2)$$

Med:

- $\gamma G_{j,inf} = 0,9$
- γQ_1 og $\gamma Q_i = 1,5$
- $\psi_{0,vind} = 0,6$.

Tabell 6.2 og Tabell 6.3 presenterer alle lastkombinasjonene som blir benyttet, både identitetsnavn og formel.

Tabell 6.2 Oversikt over lastkombinasjoner

	NAVN PÅ LASTKOMBINASJONER
LC 0	Jevn snølast med vindtrykk
LC 1	Skjev snølast med vindtrykk
LC 2	Sentrert snølast med vindtrykk
LC 3	Vindlast oppløft
LC 4	Jevn snølast uten vind
LC 5	Skjev snølast uten vind
LC 6	Sentrert snølast uten vind

Tabell 6.3 Den totale ligningen for hver av lastkombinasjonene

	LIGNINGER FOR LASTKOMBINASJONER
LC 0	$\xi \cdot \gamma G_{j,sup} \cdot (Gk_{Egenvekt} + Gk_{Egenast_tak}) + \gamma Q_1 \cdot Qk_{S_jevn} + \gamma Q_i \cdot \psi_{0,V} \cdot (Qk_{V_trykk} + Qk_{V_tvers})$
LC 1	$\xi \cdot \gamma G_{j,sup} \cdot (Gk_{Egenvekt} + Gk_{Egenast_tak}) + \gamma Q_1 \cdot Qk_{S_skjev} + \gamma Q_i \cdot \psi_{0,V} \cdot (Qk_{V_skjev} + Qk_{V_tvers})$
LC 2	$\xi \cdot \gamma G_{j,sup} \cdot (Gk_{Egenvekt} + Gk_{Egenast_tak}) + \gamma Q_1 \cdot Qk_{S_senter} + \gamma Q_i \cdot \psi_{0,V} \cdot Qk_{V_tvers}$
LC 3	$\gamma G_{j,inf} \cdot (Gk_{Egenvekt} + Gk_{Egenast_tak}) + \gamma Q_1 \cdot Qk_{V_oppløft} + \gamma Q_1 \cdot Qk_{V_tvers}$
LC 4	$\xi \cdot \gamma G_{j,sup} \cdot (Gk_{Egenvekt} + Gk_{Egenast_tak}) + \gamma Q_1 \cdot Qk_{S_jevn}$
LC 5	$\xi \cdot \gamma G_{j,sup} \cdot (Gk_{Egenvekt} + Gk_{Egenast_tak}) + \gamma Q_1 \cdot Qk_{S_skjev}$
LC 6	$\xi \cdot \gamma G_{j,sup} \cdot (Gk_{Egenvekt} + Gk_{Egenast_tak}) + \gamma Q_1 \cdot Qk_{S_senter}$

Scriptet har som formål å dele opp leddene i ligningene som vist i Tabell 6.3, og dermed lage lister for hver av de ulike last-typene. Listene for hver av last-typene vil dermed inneholder lastens karakteristiske størrelse multiplisert med tilhørende lastfaktor. Ettersom $Gk_{Egenvekt}$ vil regnes ut automatisk av Karambas last-komponent, er det bare nødvendig å ha med lastfaktorene for egenvekten i listen. Ved skjev- og sentrert snølast må scriptet danne lister for både den høyeste og laveste lastverdi. Dermed skal scriptet danne tolv lister, med oppbygning som vist i Tabell 6.4.

Tabell 6.4 Hvordan scriptet deler opp ligningene for lastkombinasjonene og etablerer lister for hver enkelt last-type

Egenvekt	Egenlast_tak	S_jevn	S_skjev (lav last-verdi)
$\xi \cdot \gamma G_{j,sup}$	$\xi \cdot \gamma G_{j,sup} \cdot Gk_{Egenast_tak}$	$\gamma Q_1 \cdot Qk_{S_jevn}$	0
$\xi \cdot \gamma G_{j,sup}$	$\xi \cdot \gamma G_{j,sup} \cdot Gk_{Egenast_tak}$	0	$\gamma Q_1 \cdot Qk_{S_skjev_lav}$
$\xi \cdot \gamma G_{j,sup}$	$\xi \cdot \gamma G_{j,sup} \cdot Gk_{Egenast_tak}$	0	0
$\gamma G_{j,inf}$	$\gamma G_{j,inf} \cdot Gk_{Egenast_tak}$	0	0
$\xi \cdot \gamma G_{j,sup}$	$\xi \cdot \gamma G_{j,sup} \cdot Gk_{Egenast_tak}$	$\gamma Q_1 \cdot Qk_{S_jevn}$	0
$\xi \cdot \gamma G_{j,sup}$	$\xi \cdot \gamma G_{j,sup} \cdot Gk_{Egenast_tak}$	0	$\gamma Q_1 \cdot Qk_{S_skjev_lav}$
$\xi \cdot \gamma G_{j,sup}$	$\xi \cdot \gamma G_{j,sup} \cdot Gk_{Egenast_tak}$	0	0

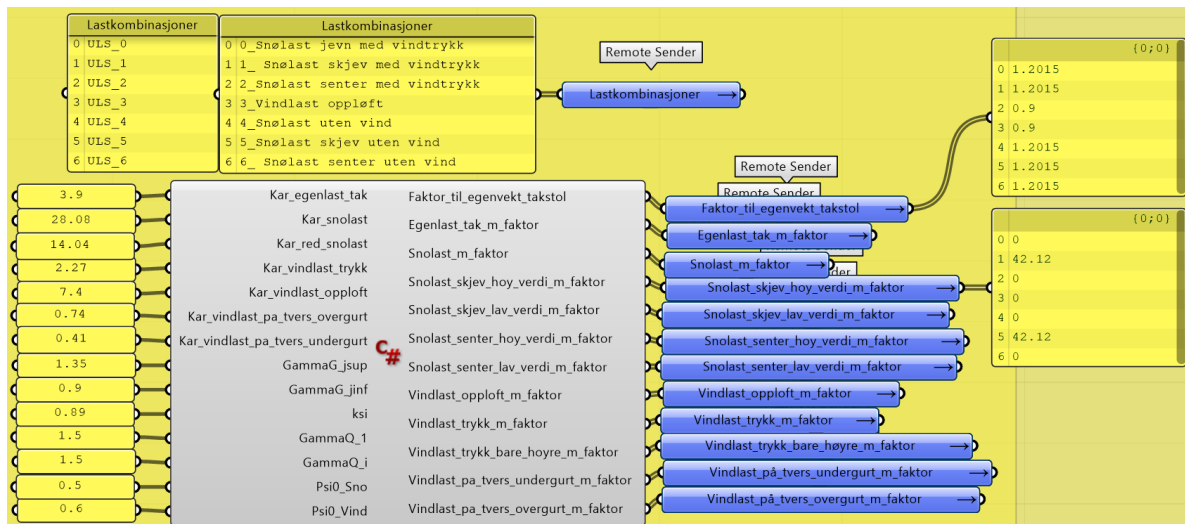
S_skjev med (høy last-verdi)	S_senter (lav last-verdi)	S_senter (høy last-verdi)	Vindlast oppløft
0	0	0	0
$\gamma Q_1 \cdot Qk_{S_skjev_høy}$	0	0	0
0	$\gamma Q_1 \cdot Qk_{S_senter_lav}$	$\gamma Q_1 \cdot Qk_{S_senter_høy}$	0
0	0	0	$\gamma Q_1 \cdot Qk_{V_oppløft}$
0	0	0	0
$\gamma Q_1 \cdot Qk_{S_skjev_høy}$	0	0	0
0	$\gamma Q_1 \cdot Qk_{S_senter_lav}$	$\gamma Q_1 \cdot Qk_{S_senter_høy}$	0

Vindlast trykk	V_trykk skjev	Vind på tvers overgurt	Vind på tvers undergurt
$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tr}$	0	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_0}$	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_U}$
0	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_skjev}$	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_0}$	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_U}$
0	0	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_0}$	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_U}$
0	0	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_0}$	$\gamma Q_i \cdot \psi_{0,v} \cdot Qk_{V_tvers_U}$
0	0	0	0
0	0	0	0
0	0	0	0

Figur 6.18 viser hvordan C#-komponenten blir presentert i Grasshopper-systemet.

Lastkombinasjonene som er dannet er i tillegg presentert ved bruk av paneler over C#-komponenten.

Disse er skrevet manuelt, og brukes bare for å holde oversikt.



Figur 6.18 C#-komponenten for lastkombinasjoner i Grasshopper-systemet.

Figur 6.19 viser hvordan deler av scriptet ser ut på innsiden. Første del av scriptet, som automatisk blir markert med grå farge, er ikke mulig å justere. Informasjonen som listes opp i dette feltet, forteller blant annet noe om biblioteket som scriptet benytter. I tillegg foreligger informasjon som omhandler skjulte funksjoner som kjøres i scriptet ved endring av inputs- og outputs. Når man legger inn alle inputs og outputs i script-komponenten, vil disse automatisk listes opp i scriptet i parentes bak «RunScript». Det er viktig at alle inputs er av typen «double», slik at de sees på som et tall. Dette kan endres ved å høyre-klikke direkte på hver av inputene på C#-komponenten og velge «double» i listen av «type hints». Dermed benevnes inputene på innsiden av scriptet som «double + navn på input», mens outputene benevnes som «Ref Object + navn på output». I linjen under «private void RunScript», finner man en klammeparentes. Denne markerer scriptets start, og det er mellom denne og sluttparentesen en skal skrive selve funksjonene som man ønsker å utføre.

Aller først defineres alle variabler som skal brukes videre. Som vist i figur, gjøres dette i scriptets linje nr. 57 til 68. For eksempel skrives første linje som følger: «var Faktor_eg_takstol = new List<double> ();». I dette tilfellet blir det definert at variabelen «Faktor_eg_takstol» skal danne en liste med tall. Semikolonet til slutt må alltid benyttes for å indikere at funksjonen skal utføres når programmet kjører. Deretter (linje nr. 70 til 164) defineres ett og ett av tall-variablene som skal inn i de tidligere definerte listene. Dette scriptes ved å legge til «.Add (funksjonen som skal utføres);» bak listens variabel-navn. Det er i tillegg presentert hvilken lastkombinasjon tallet tilhører, ved å skrive dette bakerst på linjen. Dersom to skråstreker «//» benyttes, kan det skrives kommentarer som dette uten at det vil påvirke koden.

Når alle listene med tilhørende tall-variabler er definert, spesifiseres det hvilke av listene som skal tilhøre hvilken output (linje nr. 166 til 177).

```

1  Script component: C# Script
2
3  12
4  13
5  14
6  15
7  16
8  17
9  18
10 19
11 20
12 21
13 22
14 23
15 24
16 25
17 26
18 27
19 28
20 29
21 30
22 31
23 32
24 33
25 34
26 35
27 36
28 37
29 38
30 39
31 40
32 41
33 42
34 43
35 44
36 45
37 46
38 47
39 48
40 49
41 50
42 51
43 52
44 53
45 54
46 55
47 56
48 57
49 58
50 59
51 60
52 61
53 62
54 63
55 64
56 65
57 66
58 67
59 68
60 69
61 70
62 71
63 72
64 73
65 74
66 75
67 76
68 77
69 78
70 79
71 80
72 81
73 82
74 83
75 84
76 85
77 86
78 87
79 88
80 89
81 90
82 91
83 92
84 93
85 94
86 95
87 96
88 97
89 98
90 99
91 100
92 101
93 102
94 103
95 104
96 105
97 106
98 107
99 108
100 109
101 110
102 111
103 112
104 113
105 114
106 115
107 116
108 117
109 118
110 119
111 120
112 121
113 122
114 123
115 124
116 125
117 126
118 127
119 128
120 129
121 130
122 131
123 132
124 133
125 134
126 135
127 136
128 137
129 138
130 139
131 140
132 141
133 142
134 143
135 144
136 145
137 146
138 147
139 148
140 149
141 150
142 151
143 152
144 153
145 154
146 155
147 156
148 157
149 158
150 159
151 160
152 161
153 162
154 163
155 164
156 165
157 166
158 167
159 168
160 169
161 170
162 171
163 172
164 173
165 174
166 175
167 176
168 177
169 178
170 179
171 180
172 181
173 182
174 183
175 184
176 185
177 186
178 187
179 188
180 189
181 190
182 191
183 192
184 193
185 194
186 195
187 196
188 197
189 198
190 199
191 200
192 201
193 202
194 203
195 204
196 205
197 206
198 207
199 208
200 209
201 210
202 211
203 212
204 213
205 214
206 215
207 216
208 217
209 218
210 219
211 220
212 221
213 222
214 223
215 224
216 225
217 226
218 227
219 228
220 229
221 230
222 231
223 232
224 233
225 234
226 235
227 236
228 237
229 238
230 239
231 240
232 241
233 242
234 243
235 244
236 245
237 246
238 247
239 248
240 249
241 250
242 251
243 252
244 253
245 254
246 255
247 256
248 257
249 258
250 259
251 260
252 261
253 262
254 263
255 264
256 265
257 266
258 267
259 268
260 269
261 270
262 271
263 272
264 273
265 274
266 275
267 276
268 277
269 278
270 279
271 280
272 281
273 282
274 283
275 284
276 285
277 286
278 287
279 288
280 289
281 290
282 291
283 292
284 293
285 294
286 295
287 296
288 297
289 298
290 299
291 300
300 301
301 302
302 303
303 304
304 305
305 306
306 307
307 308
308 309
309 310
310 311
311 312
312 313
313 314
314 315
315 316
316 317
317 318
318 319
319 320
320 321
321 322
322 323
323 324
324 325
325 326
326 327
327 328
328 329
329 330
330 331
331 332
332 333
333 334
334 335
335 336
336 337
337 338
338 339
339 340
340 341
341 342
342 343
343 344
344 345
345 346
346 347
347 348
348 349
349 350
350 351
351 352
352 353
353 354
354 355
355 356
356 357
357 358
358 359
359 360
360 361
361 362
362 363
363 364
364 365
365 366
366 367
367 368
368 369
369 370
370 371
371 372
372 373
373 374
374 375
375 376
376 377
377 378
378 379
379 380
380 381
381 382
382 383
383 384
384 385
385 386
386 387
387 388
388 389
389 390
390 391
391 392
392 393
393 394
394 395
395 396
396 397
397 398
398 399
399 400
400 401
401 402
402 403
403 404
404 405
405 406
406 407
407 408
408 409
409 410
410 411
411 412
412 413
413 414
414 415
415 416
416 417
417 418
418 419
419 420
420 421
421 422
422 423
423 424
424 425
425 426
426 427
427 428
428 429
429 430
430 431
431 432
432 433
433 434
434 435
435 436
436 437
437 438
438 439
439 440
440 441
441 442
442 443
443 444
444 445
445 446
446 447
447 448
448 449
449 450
450 451
451 452
452 453
453 454
454 455
455 456
456 457
457 458
458 459
459 460
460 461
461 462
462 463
463 464
464 465
465 466
466 467
467 468
468 469
469 470
470 471
471 472
472 473
473 474
474 475
475 476
476 477
477 478
478 479
479 480
480 481
481 482
482 483
483 484
484 485
485 486
486 487
487 488
488 489
489 490
490 491
491 492
492 493
493 494
494 495
495 496
496 497
497 498
498 499
499 500
500 501
501 502
502 503
503 504
504 505
505 506
506 507
507 508
508 509
509 510
510 511
511 512
512 513
513 514
514 515
515 516
516 517
517 518
518 519
519 520
520 521
521 522
522 523
523 524
524 525
525 526
526 527
527 528
528 529
529 530
530 531
531 532
532 533
533 534
534 535
535 536
536 537
537 538
538 539
539 540
540 541
541 542
542 543
543 544
544 545
545 546
546 547
547 548
548 549
549 550
550 551
551 552
552 553
553 554
554 555
555 556
556 557
557 558
558 559
559 560
560 561
561 562
562 563
563 564
564 565
565 566
566 567
567 568
568 569
569 570
570 571
571 572
572 573
573 574
574 575
575 576
576 577
577 578
578 579
579 580
580 581
581 582
582 583
583 584
584 585
585 586
586 587
587 588
588 589
589 590
590 591
591 592
592 593
593 594
594 595
595 596
596 597
597 598
598 599
599 600
600 601
601 602
602 603
603 604
604 605
605 606
606 607
607 608
608 609
609 610
610 611
611 612
612 613
613 614
614 615
615 616
616 617
617 618
618 619
619 620
620 621
621 622
622 623
623 624
624 625
625 626
626 627
627 628
628 629
629 630
630 631
631 632
632 633
633 634
634 635
635 636
636 637
637 638
638 639
639 640
640 641
641 642
642 643
643 644
644 645
645 646
646 647
647 648
648 649
649 650
650 651
651 652
652 653
653 654
654 655
655 656
656 657
657 658
658 659
659 660
660 661
661 662
662 663
663 664
664 665
665 666
666 667
667 668
668 669
669 670
670 671
671 672
672 673
673 674
674 675
675 676
676 677
677 678
678 679
679 680
680 681
681 682
682 683
683 684
684 685
685 686
686 687
687 688
688 689
689 690
690 691
691 692
692 693
693 694
694 695
695 696
696 697
697 698
698 699
699 700
700 701
701 702
702 703
703 704
704 705
705 706
706 707
707 708
708 709
709 710
710 711
711 712
712 713
713 714
714 715
715 716
716 717
717 718
718 719
719 720
720 721
721 722
722 723
723 724
724 725
725 726
726 727
727 728
728 729
729 730
730 731
731 732
732 733
733 734
734 735
735 736
736 737
737 738
738 739
739 740
740 741
741 742
742 743
743 744
744 745
745 746
746 747
747 748
748 749
749 750
750 751
751 752
752 753
753 754
754 755
755 756
756 757
757 758
758 759
759 760
760 761
761 762
762 763
763 764
764 765
765 766
766 767
767 768
768 769
769 770
770 771
771 772
772 773
773 774
774 775
775 776
776 777
777 778
778 779
779 780
780 781
781 782
782 783
783 784
784 785
785 786
786 787
787 788
788 789
789 790
790 791
791 792
792 793
793 794
794 795
795 796
796 797
797 798
798 799
799 800
800 801
801 802
802 803
803 804
804 805
805 806
806 807
807 808
808 809
809 810
810 811
811 812
812 813
813 814
814 815
815 816
816 817
817 818
818 819
819 820
820 821
821 822
822 823
823 824
824 825
825 826
826 827
827 828
828 829
829 830
830 831
831 832
832 833
833 834
834 835
835 836
836 837
837 838
838 839
839 840
840 841
841 842
842 843
843 844
844 845
845 846
846 847
847 848
848 849
849 850
850 851
851 852
852 853
853 854
854 855
855 856
856 857
857 858
858 859
859 860
860 861
861 862
862 863
863 864
864 865
865 866
866 867
867 868
868 869
869 870
870 871
871 872
872 873
873 874
874 875
875 876
876 877
877 878
878 879
879 880
880 881
881 882
882 883
883 884
884 885
885 886
886 887
887 888
888 889
889 890
890 891
891 892
892 893
893 894
894 895
895 896
896 897
897 898
898 899
899 900
900 901
901 902
902 903
903 904
904 905
905 906
906 907
907 908
908 909
909 910
910 911
911 912
912 913
913 914
914 915
915 916
916 917
917 918
918 919
919 920
920 921
921 922
922 923
923 924
924 925
925 926
926 927
927 928
928 929
929 930
930 931
931 932
932 933
933 934
934 935
935 936
936 937
937 938
938 939
939 940
940 941
941 942
942 943
943 944
944 945
945 946
946 947
947 948
948 949
949 950
950 951
951 952
952 953
953 954
954 955
955 956
956 957
957 958
958 959
959 960
960 961
961 962
962 963
963 964
964 965
965 966
966 967
967 968
968 969
969 970
970 971
971 972
972 973
973 974
974 975
975 976
976 977
977 978
978 979
979 980
980 981
981 982
982 983
983 984
984 985
985 986
986 987
987 988
988 989
989 990
990 991
991 992
992 993
993 994
994 995
995 996
996 997
997 998
998 999
999 1000

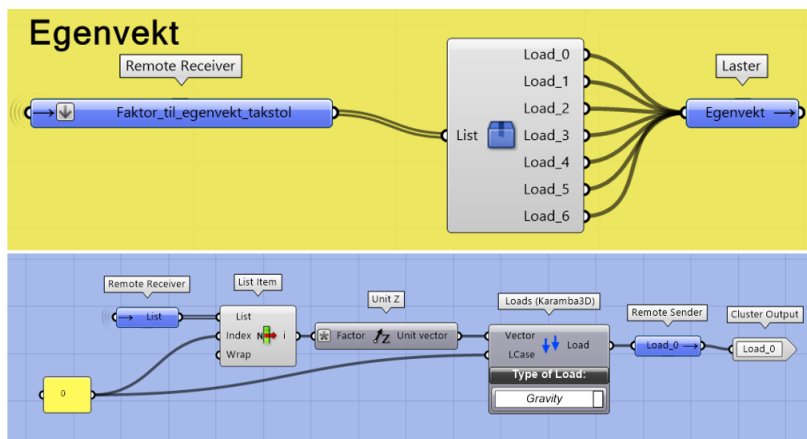
```

Figur 6.19 C#-komponenten for lastkombinasjoner fra innsiden

Clustere for laster- og lastkombinasjoner

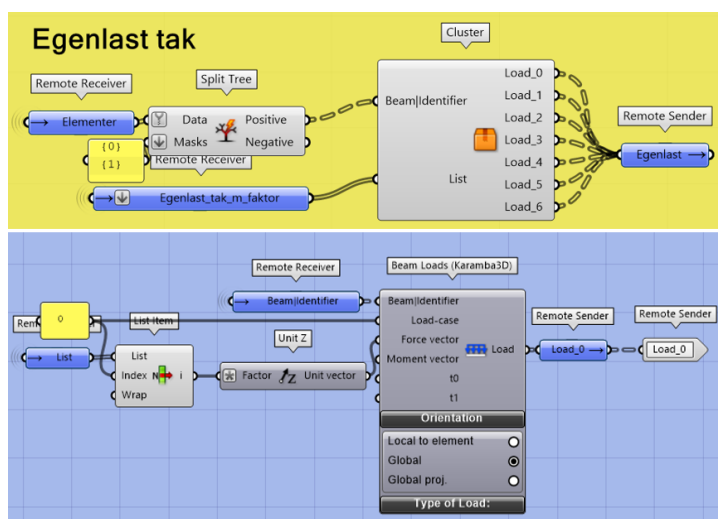
Hver av clusterene inneholder Karambas «load»-komponent repetert 7 ganger. Første komponent tar inn kreftene fra lastkombinasjon nr. 0, deretter tar neste komponent lastene lastkombinasjon nr. 1 osv. Alle de 7 komponentene stilles inn til å ha de samme egenskapene, og det er dermed bare kreftene som varierer. Figur 6.20 til Figur 6.23 viser derfor bare første komponent (for lastkombinasjon 0) for den spesifikke last-typen.

Den karakteristiske egenvekten regnes automatisk ut ved valg av «Gravity» i Karambas «Load»-komponent. Vektoren som tilkobles inneholder derfor bare lastfaktorene til kraften. Øverste del av Figur 6.20 viser hvordan clusteren presenteres i selve Grasshopper-systemet, mens nederste del viser innsiden av clusteren.



Figur 6.20 Cluster for Egenvekt (blå)

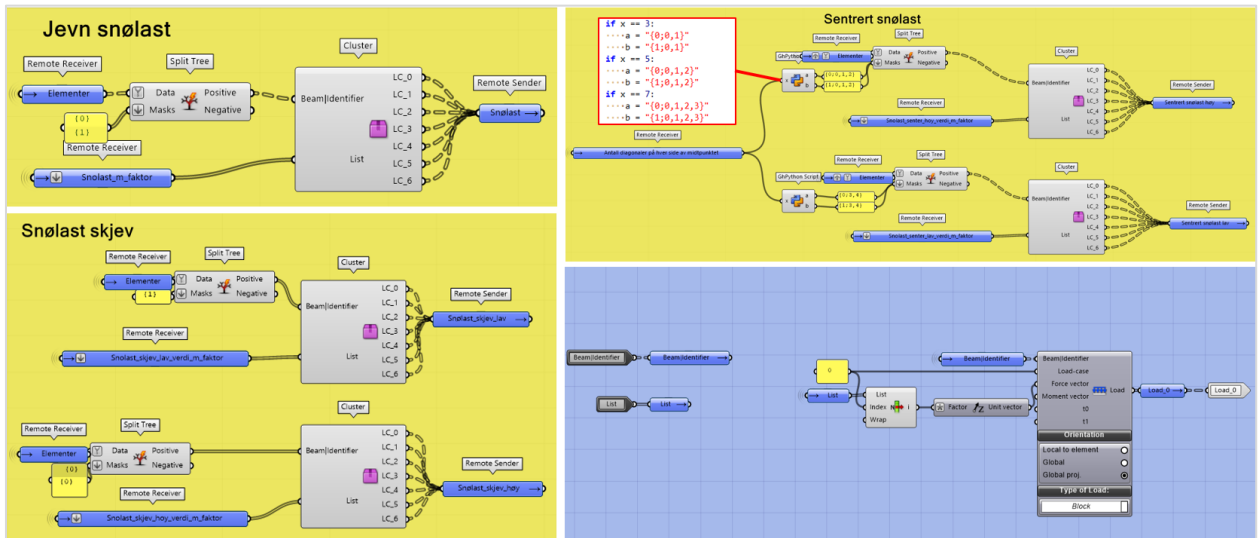
Egenlast fra tak skal være orientert langs det globale koordinatsystemet til systemet, og derfor velges «Global» i Karambas «load»-komponent. Figur 6.21 viser hvordan clusteren presenteres i selve Grasshopper-systemet i den gule ruten, mens den blå delen viser innsiden av clusteren.



Figur 6.21 Cluster for Egenlast tak (oransje)

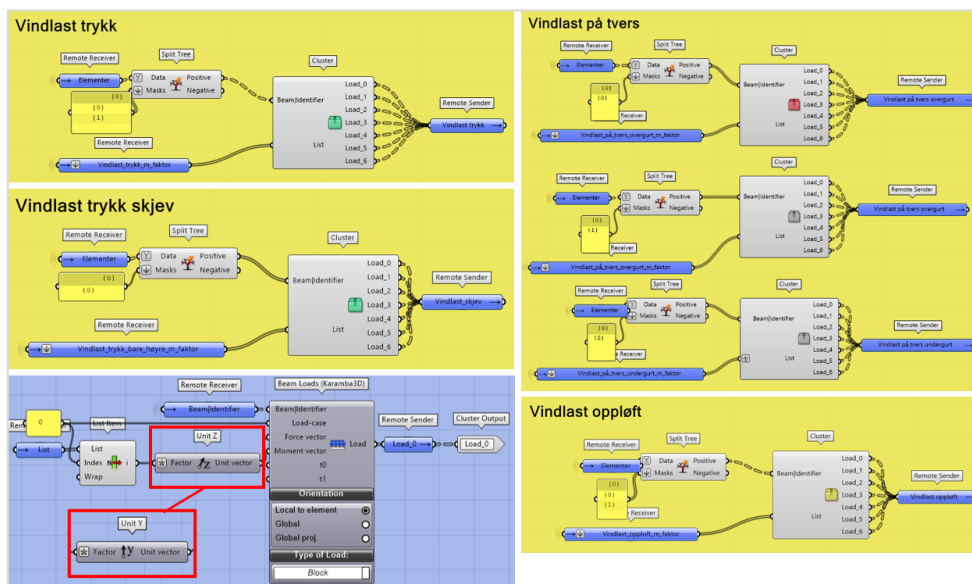
For snølastene er retningen lik for både jevn-, skjev- og sentrert snølast, som vist i Figur 6.22. Disse skal være orientert langs det globale koordinatsystemet til systemet («global projected coordinates»), i henhold til NS-EN 1991-1-3 5.2(4) [165]. Alle snølaster skal også virke i negativ z-retning. I tillegg har alle last-typen «block». Oppbyggingen i clusteren er dermed lik for alle snølaster, og disse clusterene er markert med lilla farge på ikonet. Plasseringen på bjelke-elementene, er derimot forskjellig for de ulike snølastene, men dette defineres på utsiden av clusteren. «Split tree» benyttes for å hente ut de ønskede elementene, hvor høyre- og venstre overgurt alltid er plassert i henholdsvis nulte- og første gren. Ved påføring av jevn snølast påføres like store krefter på hele høyre- og venstre overgurt. Ved skjev snølast påføres det større krefter på høyre overgurt enn på venstre. Ved sentrert snølast påføres det større krefter på overgurt-elementene som ligger over diagonalmønsteret, og en mindre kraft på de resterende elementene. Dersom det er valgt å ha fem diagonale staver, tilsvarer stavene over diagonalmønsteret elementnummer. 0, 1 og 2 i hver av

overgurtene. Dersom det imidlertid velges å ha 7 diagonale staver, må også neste element i overgurten legges til. For å få til dette har vi benyttet python-script, som markert med rødt i figuren.



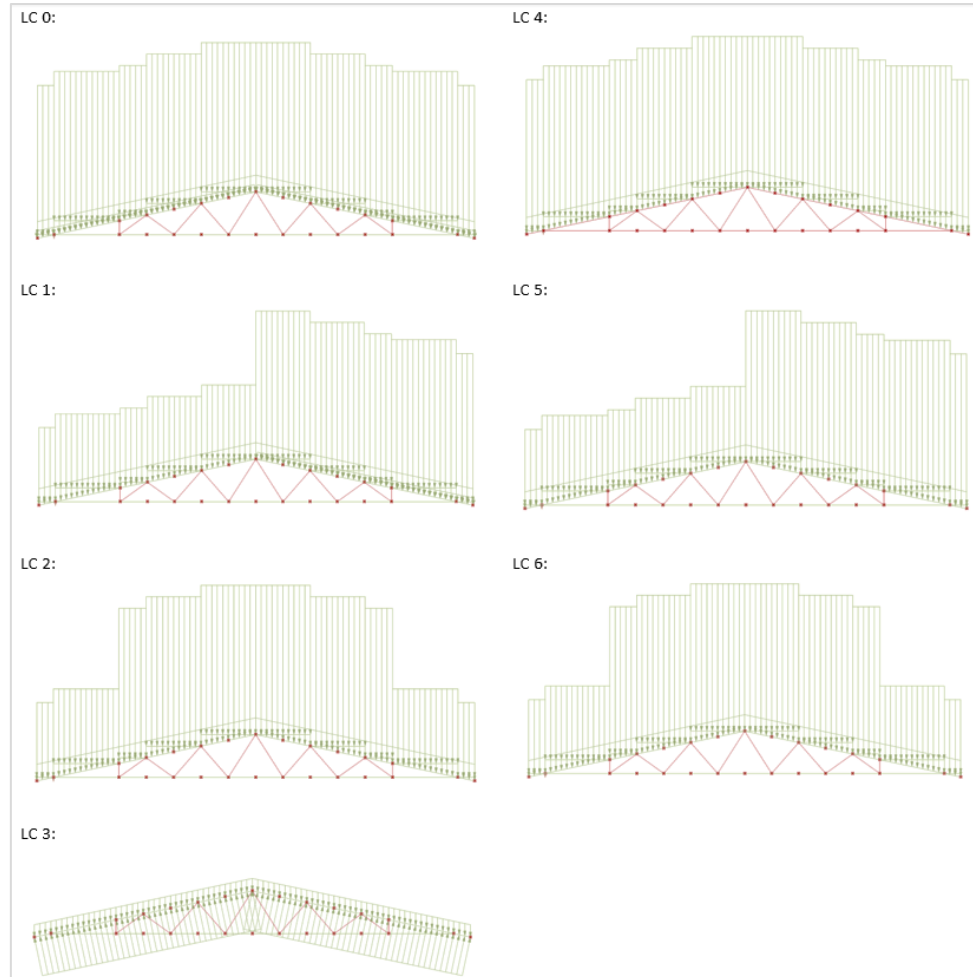
Figur 6.22 Cluster for snølaster (lilla)

Vindlastene skal alle være orientert langs det lokale koordinatsystemet til bjelke-elementet, og har last-typen «block». Det er derfor bare retningen på kraften som skiller de forskjellige clusterene. I blått vindu i Figur 6.23, er endringene som gjøres i de forskjellige clusterene markert med rødt. Vindlaster i trykk virker i negativ z-retning (clusterer markert med grønn), vindlast som gir oppløft virker i positiv z-retning (cluster markert med gul) og vindlast på tvers virker både i positiv- og negativ y-retning (clusterer markert med rød og grå farge). Ettersom rekkefølgen for både høyre- og venstre overgurt starter i toppunktet og vender utover, har disse motsatt retning i forhold til hverandre. Denne oppbyggingen er nødvendig for at systemet skal kunne justeres med antall staver, men skaper dermed forstyrrelser for lastene. Alle lastene for vindlast på tvers skal gå i samme retning, men vil ha omvendt retning i forhold til hverandre dersom begge legges i positiv y-retning. Derfor får høyre overgurt en cluster for selv, med kraftens retning definert som negativ y-retning.



Figur 6.23 Cluster for vindlaster

Visning av lastene kan presenteres visuelt ved hjelp av komponentene som presenteres i neste delkapittel (Karamba-analyse). Figur 6.24 er en oversikt over alle lastkombinasjoner. Lastkombinasjon 0, 1, 2 og 3 også har vindlast rettet innover i planet (vindlast på tvers) og vises dermed ikke fra dette perspektivet.



Figur 6.24 Presentasjon av alle laster- og lastkombinasjoner

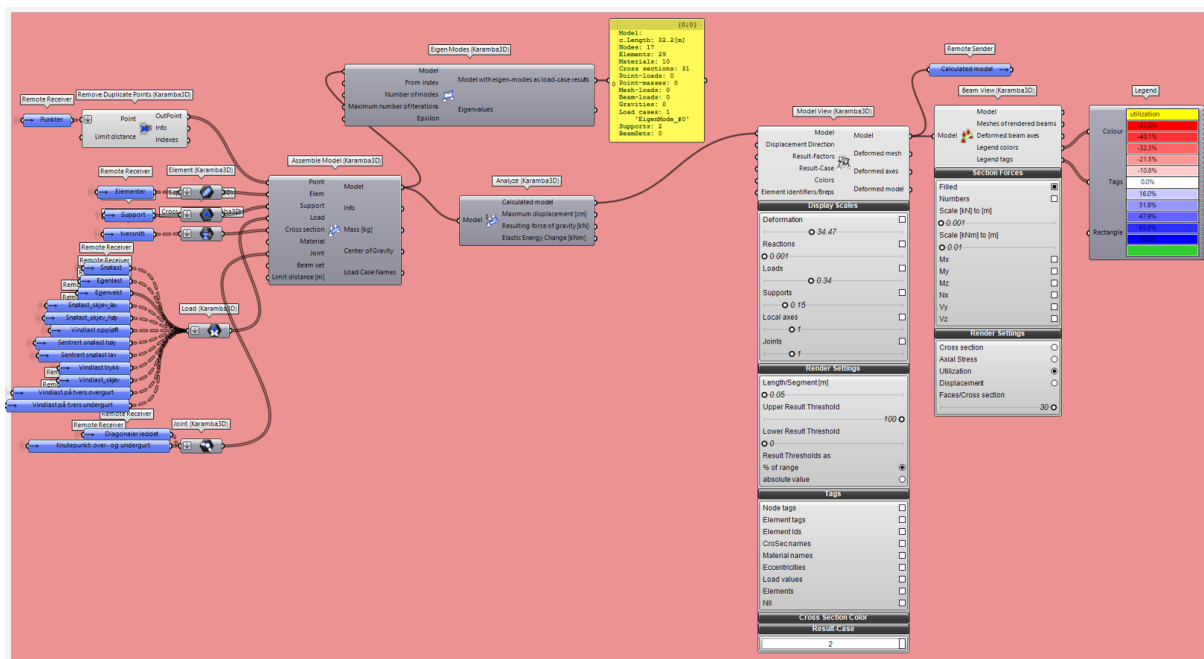
6.4.7 Karamba3D-analyse

Systemet for Karamba3D-analyse og komponenter for visualisering av resultatene fra analysen er markert med rød/mørk rosa farge i modellen, som vist i Figur 6.25. Her samles informasjon fra tidligere omtalte grupper: «Fra linje- til bjelkeelementer», «Material- og tverrsnittsegenskaper», «Opplegg og forbindelser» og «Laster». I tillegg fjernes unødvendige punkter med komponenten «Remove Duplicate Points». All informasjon sammenføres i komponenten «Assemble model», hvor verdiene blir klargjort for analyse. Komponentene «Eigen modes» er benyttet for å kontrollere at informasjonen stemmer.

Dermed føres modellen videre til «Analyse» komponenten, hvor selve FEM-analysen blir utført. Dette er én av flere algoritmer for FEM-analyse i Karamba3D. Komponentene analyserer mekanisk respons, m.fl. i form av maksimal node-forskyvning, indre deformasjonsenergi og tyngdekraft.

Komponenten forutsetter første ordens normalkrefter, samt at lengdeendringene som oppstår fra laterale deformasjoner kan neglisjeres. Altså forutsettes en «lineær elastisk analyse» [49].

Videre benyttes komponenten «Model View» for å kunne visuelt framstille størrelsen på opplagerreaksjoner og lastkombinasjoner. I tillegg kan komponenten benyttes for å se hvilket navn og materiale som tilhører hver av bjelke-elementene. For fremstilling av hver enkelt lastkombinasjon, velges ønsket lasttilfelle ved komponentens «Result-case». I tillegg benyttes komponenten «Beam View» for å kunne framstille moment-, skjær- og aksialkraftdiagrammer. Under «render settings» kan man også visualisere elementenes fulle tverrsnitt. Samtidig har «Beam View» komponenten også en mulighet for å vise utnyttelsen i farger. Denne vil imidlertid ikke være riktig i henhold til materialeegenskapene som er definert i systemet (se diskusjon i kapittel 8.4.1) [1].



Figur 6.25 Oversiktsbilde av gruppen fra Grasshopper-modellen som utfører- og visualiserer resultatene av Karamba-analyse

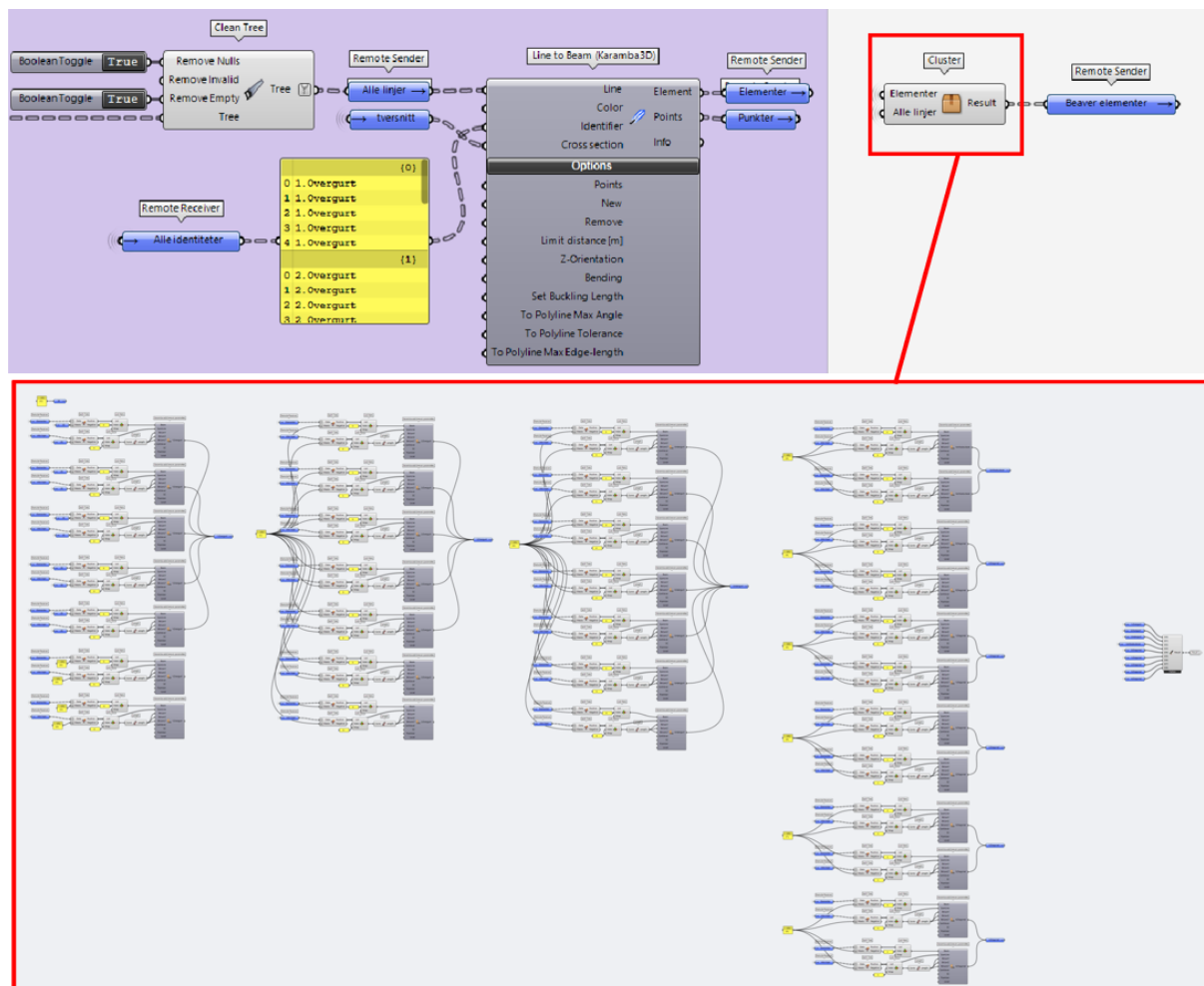
6.4.8 Utnyttelsesanalyse av elementer i Grasshopper

Vertikal og diagonale staver, samt over- og undergurt er kontrollert for bøyemoment-, skjær-, strekk- og trykkrefter i bruddgrensetilstander i henhold til kapittel 6 i EK5 (NS-EN 1995) [63]. I tillegg er også knekking både i y- og z-retning kontrollert etter kapittel 6 Ek5 (NS-EN 1995) for alle elementene [63]. Det ble visuelt programmert en komponent som utfører alle disse kontrollene. Denne komponenten ble brukt for alle elementene, dette gjør at alle kontroller utføres for alle elementene.

For å kontrollere elementene i Grasshopper ble det gjort flere mislykkede forsøk, før en metode ble valgt. I det første forsøket ble plugg-in verktøyet Beaver brukt for å kontrollere elementene, men dette skapte flere problemer. For det første eksisterte det ikke noen brukermanualer for dette verktøyet, og det eneste tilgjengelige kildene var YouTube- kanalen til «Beaver Structures» [166] og forskningsartikkelen i kapittel 3.7.3 [6]. Disse forklarte imidlertid kun bruken av en tidligere versjon av Beaver som ikke eksisterte på food4rhino nettsiden. I tillegg var det store forskjeller mellom den tilgjengelige 1.0.3 versjonen av Beaver og den som ble brukt i videoene og artikkelen. Det eneste som

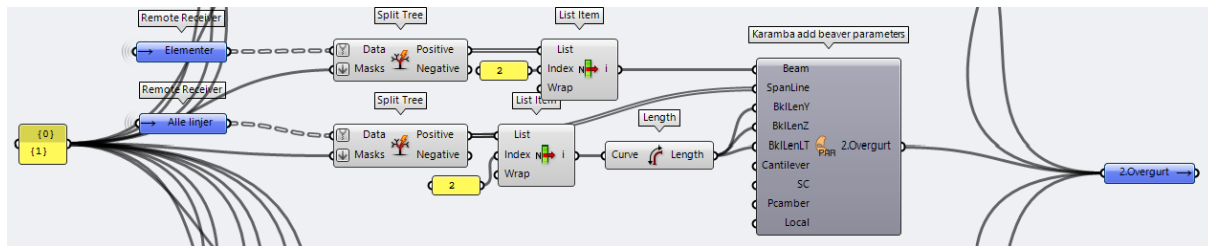
hjalp oss var eksempelfilen for bjelkeanalyse med Beaver 1.0.3 versjonen på food4rhino nettsiden [153]. Eksempelfilens oppbygging ble forsøkt å benytte direkte i vår egen Grasshopper modell.

For å få til dette måtte hvert enkelt element fra Karambas «Line to Beam»-komponent, samt dens tilhørende linje, omgjøres til Beaver-elementer. Fulle lister med alle elementer kunne ikke tilkobles Beaver-komponenten direkte, og derfor måtte det bygges en cluster som overførte ett og ett element til «Karamba add Beaver parameters»-komponenten. Denne fremgangsmåten er presentert i Figur 6.26.



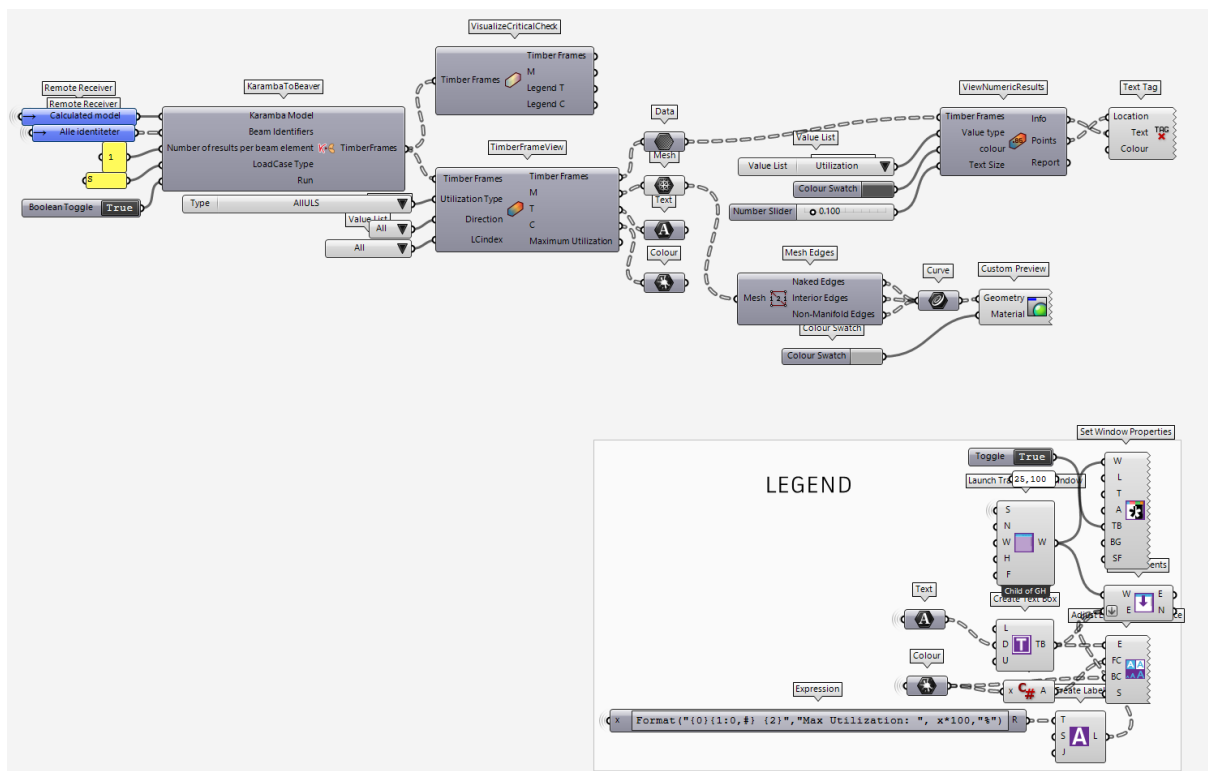
Figur 6.26 Overføring fra Karamba-elementer til Beaver-elementer

Figur 6.27 viser hvordan én av overføringene ble utført i clusteren. Alle de andre overføringene hadde tilsvarende fremgangsmåte. Komponenten «Split Tree» og «list Item», hentet ut ønsket bjelke og linje. I tillegg ble «length»-komponenten benyttet for å bestemme knekkleddene. Til slutt ble alle Beaver-elementene samlet i en liste og tilknyttet Karamba-analysen istedenfor karambas bjelke-elementer.



Figur 6.27 Overføring av «2.Overgurt» til Beaver-element

Figur 6.28 presenterer hvordan resultatet fra Karamba3D-analysen ble tilknyttet komponentene fra eksempelfilen for å utføre kontroll av elementenes utnyttelse.



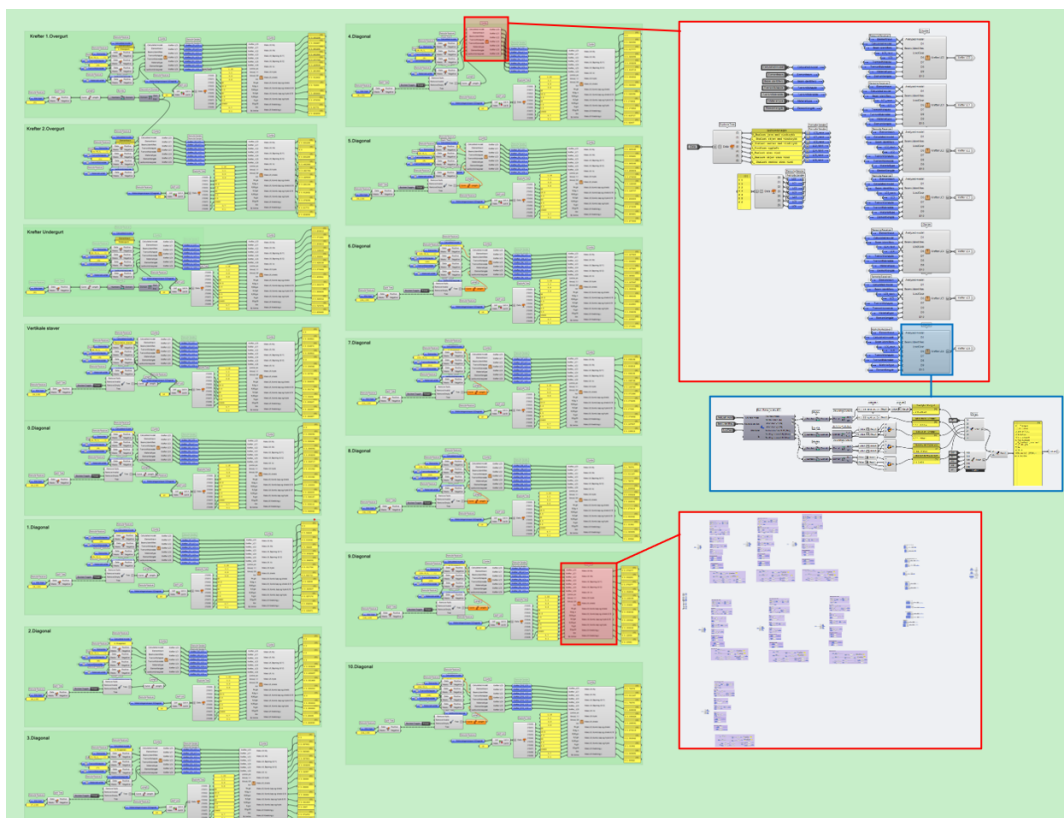
Figur 6.28 Utnyttelseskontroll med Beaver-komponenter

Deretter oppstod problemer tilknyttet resultatet fra kontrollanalysen. Beaver kunne ikke utføre en kontrollanalyse av de vertikale stavene i fagverket, noe som påvirket resultatet til hele modellen. Visningen av resultatet er presentert i Figur 6.29. Kontrollen resulterte i maks utnyttelse på 151% i overgurten og 117% i undergurten, mens de resterende verdiene var i underkant av 100%. Vi kan ikke utelukke at det er en mulighet for at modellen vår ikke er oppbygget på riktig måte med Beaver, men på grunn av tregheten Beaver skapte i modellen og den begrensede informasjon som var om Beaver, ble det besluttet å fjerne den fra modellen. Det var også flere problemer knyttet til bruken av Beaver, men disse nevnes i delkapittel 6.4.9.

Excel-dokumentet for kontrollanalyse er gitt i Vedlegg 3, og ble brukt for å kontrollere at beregningene som ble utført med visuell programmering ble utført på riktig måte. I tillegg sorterer dokumentet kreftene på en oversiktig måte, og det ble derfor også benyttet for å sammenligne kreftene mot Sweco sine resultater. Komponenter som ble laget for å overføre data til Excel, er derfor ikke forandret i modellen. Dessuten blir input-verdiene til Excel-systemet også benyttet for kontrollene som utføres med visuell programmering. Videre forklares hvordan utnyttelsesberegningen ble bygget i modellen ved hjelp av visuell programmering.

Visuell programmeringsoppbygging for kontroll av bruddgrensetilstand

Systemet som utfører kontrollene basert på visuell programmering, utgjør venstre del av den grønne gruppen (Figur 6.2, del 9). Et utklipp fra denne delen av gruppen, samt oversiktsbilder av dens tilhørende clusterer, er vist i Figur 6.31. Programmeringen for hver av elementene (1.Overgurt, 2.Overgurt, undergurt osv.) benytter de samme clusterene. Clusteren for utsortering av krefter blir vist øverst til høyre i Figur 6.31, mens clusteren som utfører alle bruddgrensekontroller blir vist nederst til høyre i Figur 6.31. Den førstnevnte clusteren (for utsortering av krefter), inneholder dessuten en annen type cluster på innsiden, som vist med blå ramme.



Figur 6.31 Systemet for utføring av kontroll i bruksgrensetilstand basert på visuell programmering

Input-verdiene som føres inn i disse clusterene, er relativt like for alle elementene. Den største forskjellen er nummereringen i «Panel»-listene, som henter ut de riktige belastningene for det bestemte elementet. Oppbyggingen av inputene for utnyttelsesberegning av under- og overgurtene er utført på en annen måte enn de diagonale og vertikale stavene. Videre presenteres hvordan

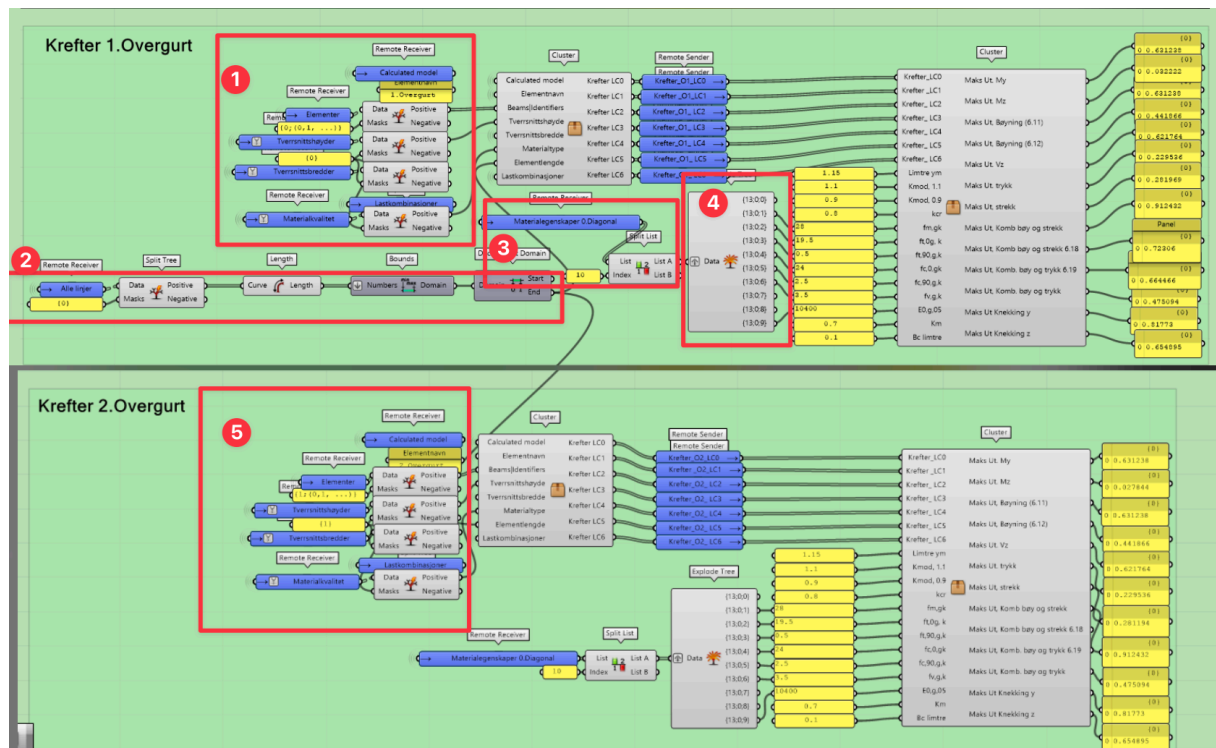
oppbyggingen av inputtene er utført for over- og undergurten, samt de vertikale og diagonale stavene. Deretter forklares programmeringen for innholdet i hver av clusterene.

Inputverdier for 1. overgurt, 2. overgurt og undergurt

I limtrefagverket er det to kontinuerlige overgurter. Figur 6.32 illustrerer den programmerte oppbyggingen for begge overgurtene, samt hvordan oppbyggingen henger sammen. Videre følger forklaring i henhold til de nummererte rammene i figuren.

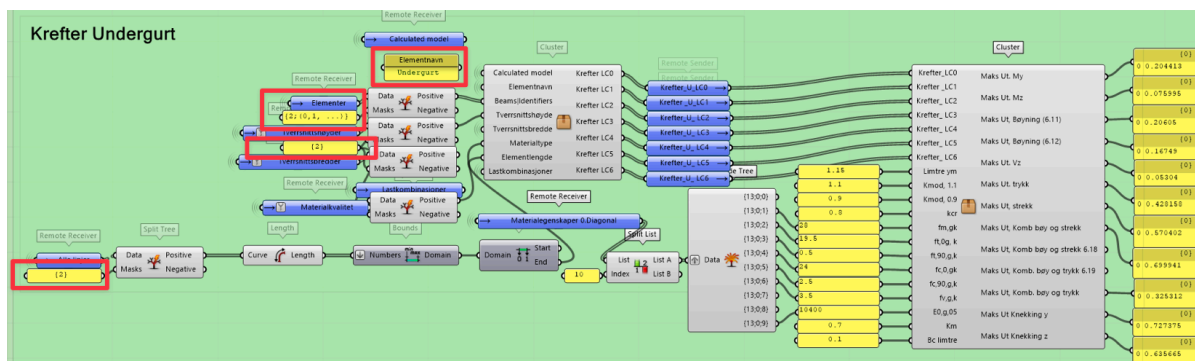
I del 1 velger «Split tree»- komponenten den nødvendige informasjon for overgurt nr.1. Inputtene til alle «Split Tree»- komponentene, er informasjon og verdier for alle elementene til fagverket. «Split Tree»- komponentene velger deretter informasjon og verdiene for det bestemte elementene, basert på listene som er tilkoblet dem. Dermed vil clusteren for utsortering av krefter få tildelt riktig bjelke, tverrsnittshøyde, tverrsnittsbredde og materialkvalitet. Videre benyttes «Calculated model», som henter alle beregninger som utføres i Kramba3D, og listen «1.Overgurt» som spesifiserer elementnavnet. I tillegg hentes listen med «Lastkombinasjoner», som overfører definerte navn for hver enkelt lastkombinasjon. Del 1 er helt lik del 5 i Figur 6.32, bortsett fra verdiene i «panel»-listene, som forandres ut ifra hvilke element som beregnes.

I del 2 velges lengden til elementet. Siden overgruten er oppdelt i flere elementlengder, velges den lengste oppdelte elementlengden, som senere skal benyttes for kontroll av knekking. Deretter velges materialkvaliteten til elementet i del 3. I dette tilfellet er det valgt GL28c, som er nr. 10 fra Excel-filen til Beaver, se vedlegg 1. Til slutt hentes de nødvendige materialegenskapene fra det valgte materialet i del 4. Materialeegenskapene benyttes videre i clusteren for utnyttelseskontroller.



Figur 6.32: Visuelle programmeringen for utnyttelsesanalyse av overgruten

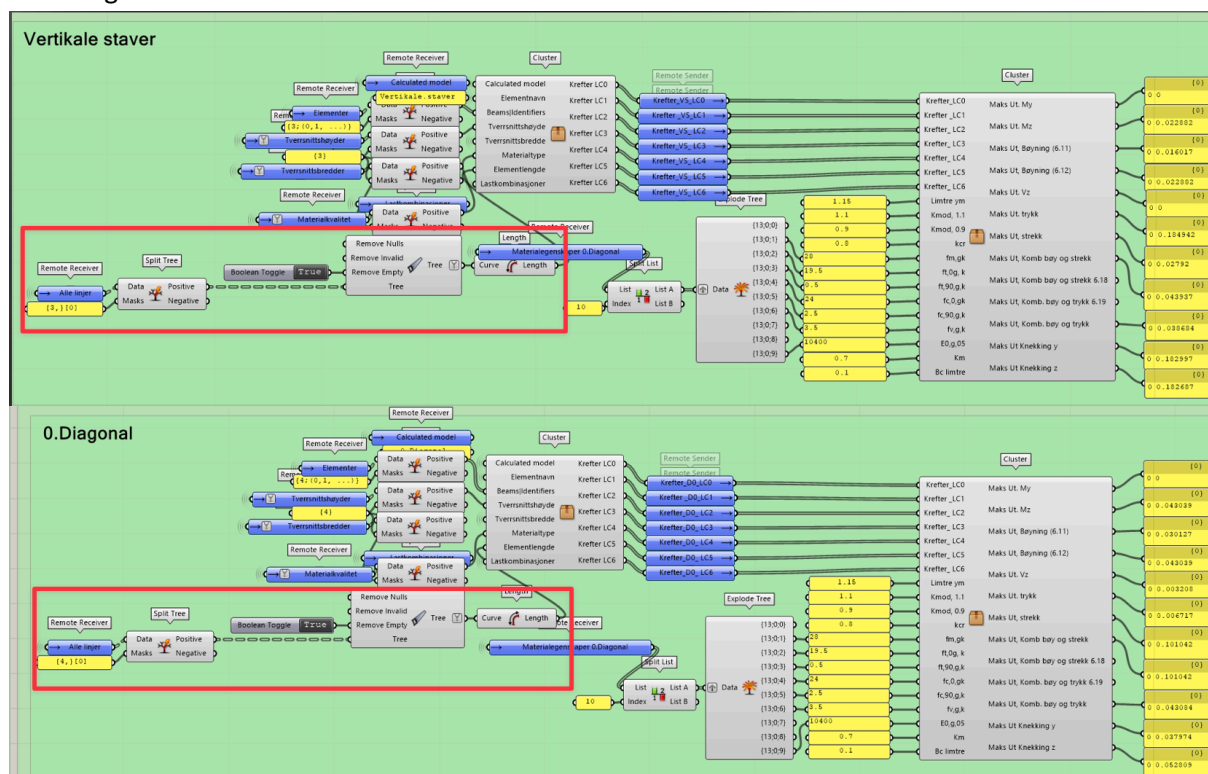
Inputverdiene for undergruten sorteres omtrent på samme måte som for hver av overgrutene. Denne eneste forskjellen er igjen verdiene fra «panel»-listene, som er markert med rødt i Figur 6.33. Egentlig skal undergruten være skjøtet på midten, men dette er ikke utført i denne oppgaven.



Figur 6.33: Kontroll undergurt

Inputverdier for vertikale og diagonale staver

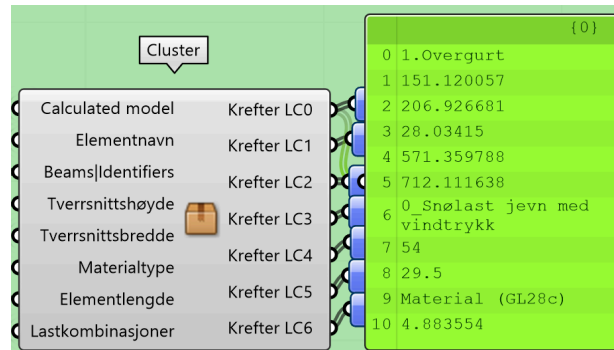
I Figur 6.34 vises programmeringen i Grasshopper for både de vertikale og diagonale stavnene i limtrefagverket. Oppbyggingen er identiske med over- og undergrut beregningene, men den markerte inputen i Figur 6.34 er oppbygget på en annen måte. Dette kommer av at elementlengden til den bestemte staven velges fra «panel»-listen. Denne oppbyggingen er utført for alle stavnene som er i limtrefagverket.



Figur 6.34: Visuelle programmeringen for utnyttelsesanalyse av stavnene

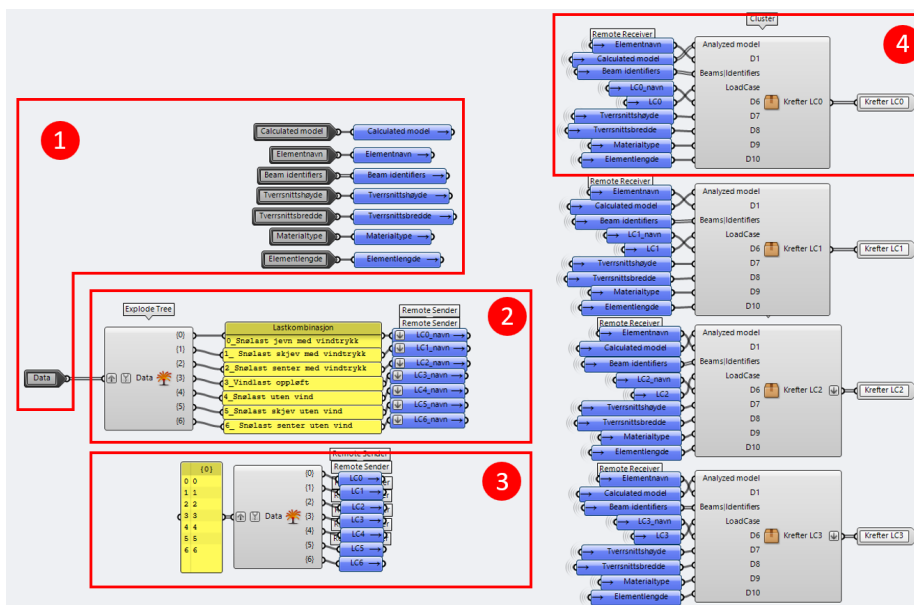
Cluster for utsortering av krefter

Clusteren som sorterer ut kreftene fra den lineær-elastiske analysen, ble først og fremst designet for overføring av data til Excel. Derfor gir den ut noe overflødig data, med tanke på hvilken informasjon som er nødvendig for beregningene som programmeres i Grasshopper. Dette gjelder blant annet identitetsnavn, navn på lastkombinasjon og materialnavn. Videre, er det benyttet sju lastkombinasjoner på limtrefagverket i henhold til oppgavens case. Dersom flere lastkombinasjoner skal brukes, må dette tilføyes av brukeren manuelt. Komponenter samler inn nødvendig informasjon som brukes til beregningsutnyttelse av elementene. Outputene vises i Figur 6.35 og er; elementnavn, maks skjær (Vz), maks moment (My og Mz), maks trykk og strekk, lastkombinasjonen, tverrsnittshøyde og bredde, materialkvalitet og elementlengde. Denne dataen overføres til utnyttelsesanalysen av elementet.



Figur 6.35: Cluster Output data

Figur 6.36 viser oppbyggingen av hvordan programmeringen oppbygges inne i Cluster-komponenten. Videre følger forklaring i henhold til de nummererte rammene i figuren. I første del hentes alle inputene som blir tilføyet clusteren fra utsiden. I del 2 fordeles navnene til hver lastskombinasjon av «Explode Tree» komponenten. Deretter benyttes også «Explode tree»-komponenten for å fordele indeksene til hver lastkombinasjon i del 3. Alle outputer fra del 1, 2 og 3 blir sortert ut til hver sin Cluster-komponent (del 4), som skal hente ut kreftene fra hver av lastkombinasjonene. Det er totalt sju av cluster-komponenter, og hver komponent representerer hver lastkombinasjon. Komponentene beregner og henter ut verdiene til normal, skjær og momentkreftene på det bestemte elementet. I



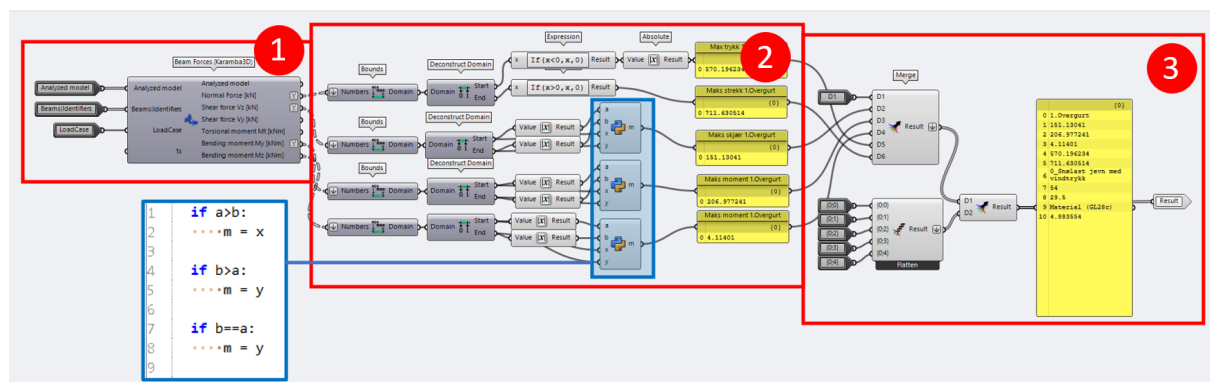
Figur 6.36: Oppbyggingen av Komponent for Belastningskrefter

Figur 6.37 viser hvordan Cluster-komponenten, som henter ut kreftene fra hver av lastkombinasjonene, er oppbygget fra innsiden. Videre følger forklaring i henhold til de nummererte rammene i denne figuren.

I første del sorterer «Beam Forces (Karamba3D)»-komponenten ut de ulike belastningskreftene til det bestemte elementet etter den valgte lastkombinasjonen. «Analyzed model» inputen kommer med modell informasjon som er generert av Karamba3D fra «Model Wiew»-komponenten i analysedelen til Karamba3D. I under- og overgrutene er det flere elementer som har forskjellige verdier av belastningskreftene. For disse velges den høyeste belastningsverdien for alle elementene som er definert som over- eller undergurt. Mens for de diagonale og vertikale stavnene velges den høyeste verdien for den bestemte staven. Alle skjær, moment og aksialkrefter blir definert i hver sin liste i «Beam Forces (Karamba3D)»-komponenten. I del 2, hentes derfor de verdiene som skal benyttes for bruddgrensek kontrollene. For beregningen av skjær (V_z) og bøyemomentene (M_y og M_z) hentes høyeste og laveste verdi fra listen, ved bruk av komponentene «Bounds» og «Deconstruct Domine». Men listen kan inneholde både negative og positive verdier, og det er ikke nødvendigvis den høyeste positive verdien man ønsker å benytte i bruddgrensek kontrollene. I tillegg vil man ikke bare hente den høyeste absoluttverdien, da man også ønsker å vire kraftens fortegn. Derfor ble det laget et script, som vist med blå ramme i Figur 6.37, som henter ut den høyeste absoluttverdien, med dens tilhørende fortegn.

I beregningen av normalkraft velges den høyeste positive verdien som er strekkraften og den laveste negative verdien som er trykkraften av «Deconstruct Domain»-komponenten. For at feil verdier ikke skal velges for trykk og strekk kreftene, er det bruk «Expression»-komponenter som kun velger negative verdier for trykk og positive verdier for strekk. Siden over- og undergurten er sammensatt av flere elementer, vil enkelte elementer være utsatt for trykkrefter, mens andre vil være utsatt for strekkrefter. Begge disse belastningene blir dermed uthentet og kontrollert for over- og undergurten. Når det kommer til stavnene som enten kan ha strekk eller trykk, velges kun den normalbelastningen som påvirker det bestemte elementet.

I del 3 sammenkobles all output informasjon av «Merge»-komponentene til en liste, som videre brukes for utnyttelsesanalysen.



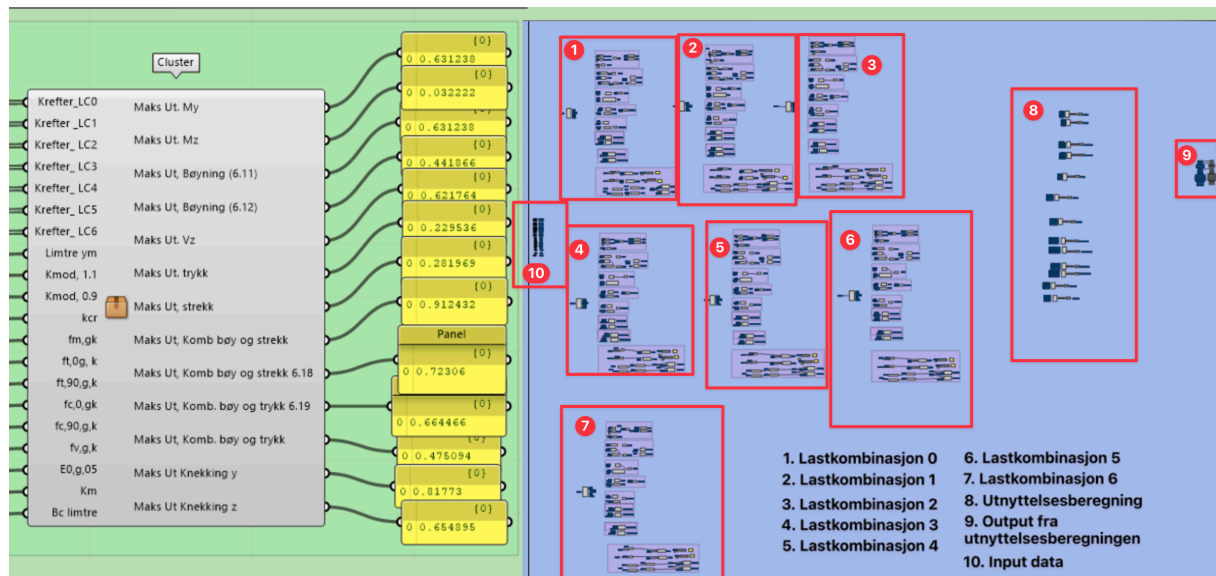
Figur 6.37: Beregning av belastningskrefter i Cluster-komponent

Komponent for kontrollanalyse av elementer

Til venstre i Figur 6.38 vises clusteren for utnyttelsesanalyse av elementene. Til høyre illustreres hvordan komponenten er programmert fra innsiden. Hvert element får tildelt hver sin cluster, og clusteren inneholder kontroller i henhold til kapittel 6 i EK5, for hver enkelt lastkombinasjon. Hver av de nummererte rammene i Figur 6.38 illustrerer følgende:

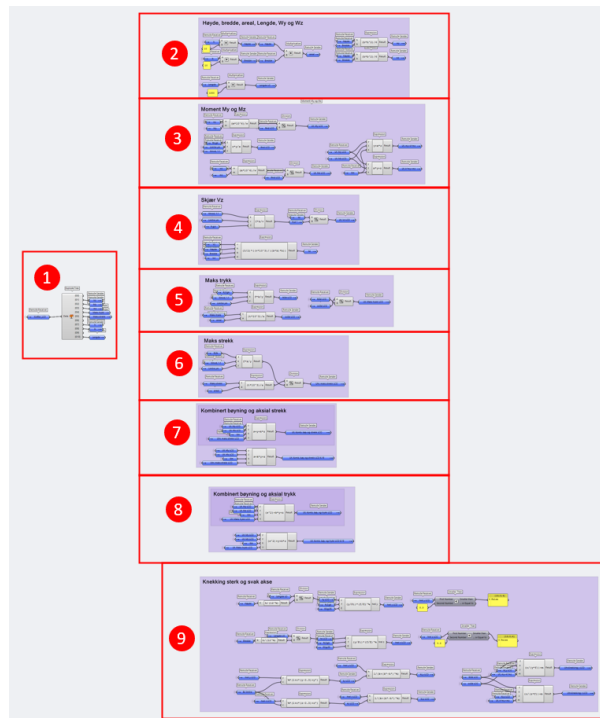
1. Kontroll av verdiene fra «Krefter_LC0»
2. Kontroll av verdiene fra «Krefter_LC1»
3. Kontroll av verdiene fra «Krefter_LC2»
4. Kontroll av verdiene fra «Krefter_LC3»
5. Kontroll av verdiene fra «Krefter_LC4»
6. Kontroll av verdiene fra «Krefter_LC5»
7. Kontroll av verdiene fra «Krefter_LC6»
8. Utsortering av høyeste utnyttelsesverdi fra kontrollene av alle lastkombinasjoner. Hvert resultat sendes videre til del 9.
9. Sortering av output-verdier (som vist i «Panel»-listene til venstre i figur).
10. Sortering av input-verdier for kontrollanalysen.

Del 1-6 i Figur 6.38 har identisk oppbygging, og skilles bare ved hvilken lastkombinasjon som benyttes.



Figur 6.38: Oppbygging av kontrollanalyse komponent

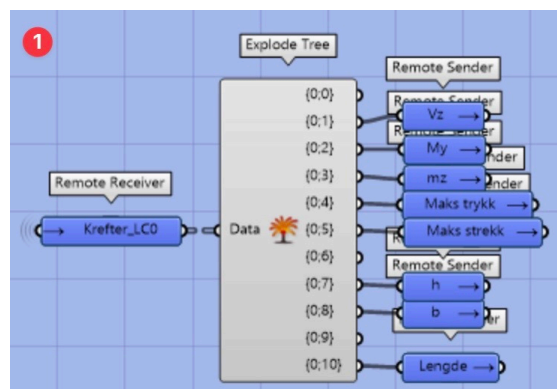
Figur 6.39 er viser hvordan verdiene kontrolleres for én av lastkombinasjonene. Hver kontroll er nummerert fra 1-9.



Figur 6.39: Oversikt over beregningskontrollen etter EK5 for elementer

Videre følger forklaring i henhold til de nummererte rammene i Figur 6.39.

Del 1 henter de nødvendige dataene fra clusteren som sorterer ut kreftene, og bruker disse videre i kontrollberegningene. Oppbygningen for listen med krefter som kommer inn, ble laget for å sende nødvendig informasjon til Excel-dokumentet. All input-data er dermed ikke nødvendig for kontrollberegningen. Derfor benyttes «Explode Tree»-komponenten, og figur viser de uthentede dataene som er skjær, moment (My og Mz), trykk, skrekk, høyde, bredde og lengden til det bestemte elementent. Programmeringen er vist i Figur 6.40.



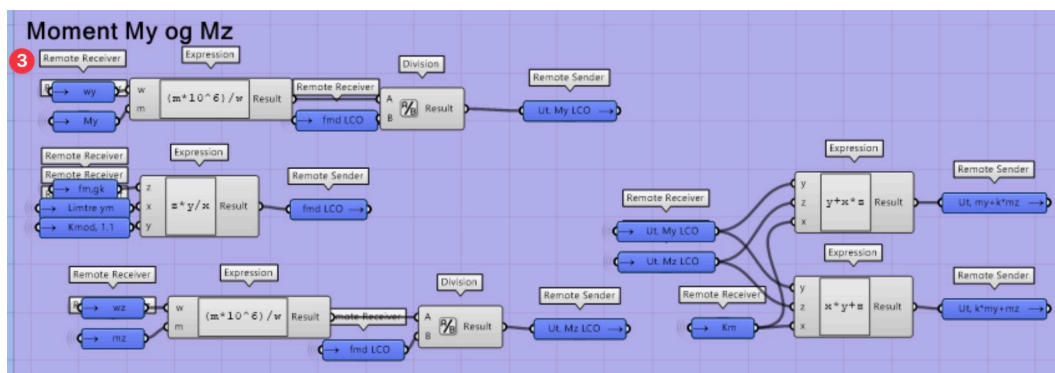
Figur 6.40: Input data for kontrollanalysen

I del 2, blir høyden og bredden omgjort fra centimeter til millimeter og lengden fra meter til millimeter. Videre blir motstandsmomentet beregnet både i svak og sterk akse. Måten dette er visuelt programmert på vises i Figur 6.41.



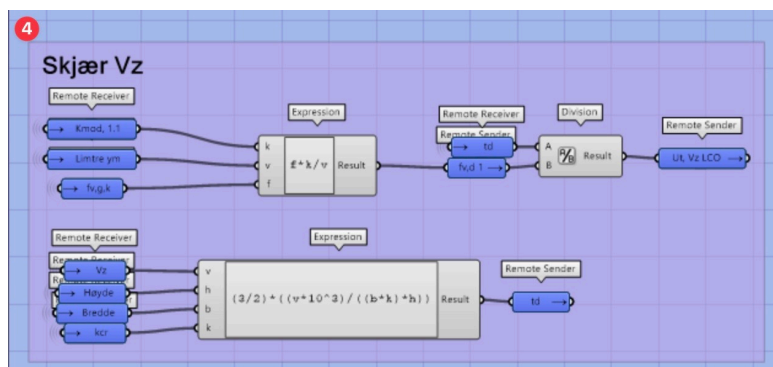
Figur 6.41: Beregning motstandsmoment

I del 3 blir utnyttelseskontrollen for bøyemomentet for sterk og svak akse og kombinasjonen av dem beregnet. Dette er gjort ved å bruke «Expression»-komponenten til å lage beregningsformler etter EK5. Figur 6.42 viser hvordan beregningen er utført i Grasshopper.



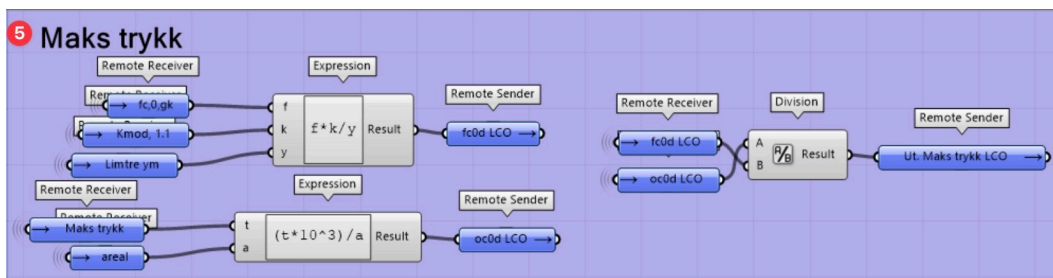
Figur 6.42: Utnyttelse av bøyemoment sterk og svak akse

I del 4 beregnes utnyttelsene for skjærkraft i sterk akse, tilsvarende ligning 3.4, 3.15 og 3.16 fra kapittel 3.4.7. Figur 6.43 viser hvordan dette er oppbygget i Grasshopper.



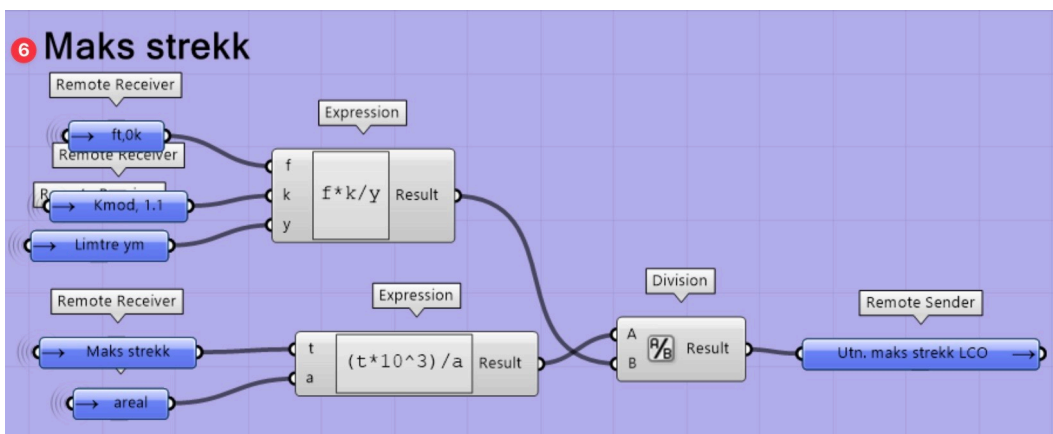
Figur 6.43: Utnyttelseskontroll Skjær (Vz)

I del 5 blir utnyttelsen av maksimalt aksialt trykk kontrollert, tilsvarende ligning 3.5, 3.17 og 3.18 fra kapittel 3.4.7. Figur 6.44 viser hvordan dette er visuelt programmert i Grasshopper.



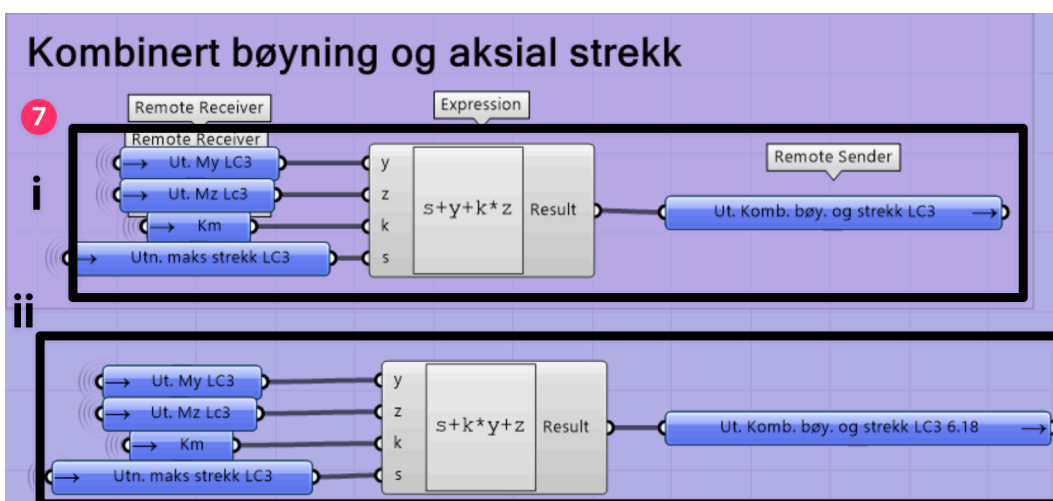
Figur 6.44: Utnyttelseskontroll av aksial trykk

I del 6 kontrolleres utnyttelsen for maks aksiale strekkrefter i forbindelsen, tilsvarende ligning 3.6, 3.20 og 3.21 fra kapittel 3.4.7. I disse beregningene benyttes derimot brutto tverrsnitt, ettersom Sweco ikke benytter nettotverrsnitt i sine beregninger. Figur 6.45 viser hvordan dette er visuelt programmert i Grasshopper.



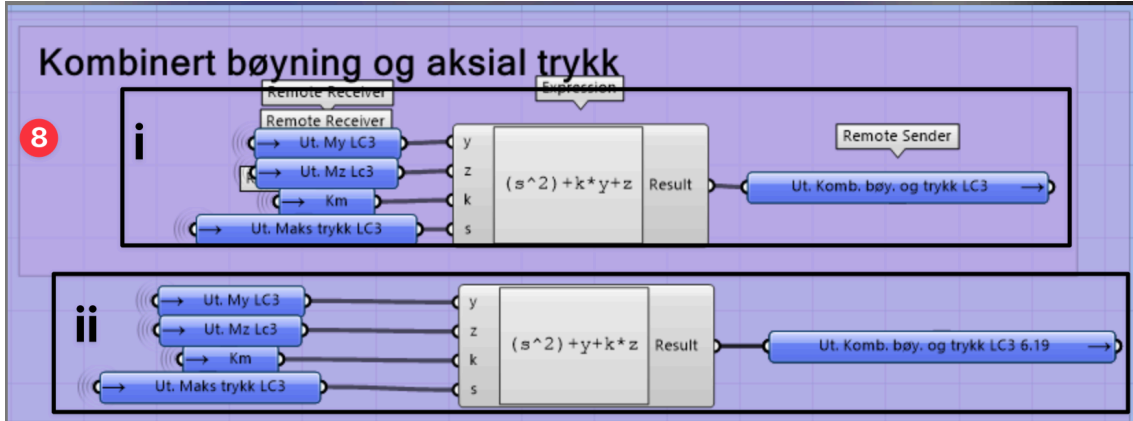
Figur 6.45: Utnyttelseskontroll av aksial strekk

I del 7 beregnes utnyttelsen for kombinasjon av bøyingsmoment og aksial strekk i henhold til 6.2.3 i EK5. Figur 6.46 viser hvordan dette er utført i Grasshopper.



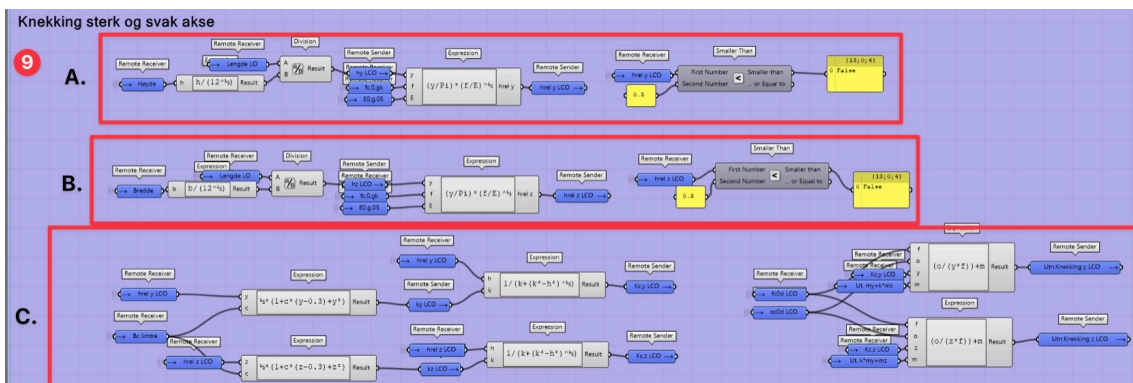
Figur 6.46: Kontroll kombinert bøyning og aksial strekk

I del 8 kontrolleres utnyttelsen for kombinasjonen av momentbøyning og aksial trykk i henhold til 6.2.4 i EK5. Måten dette er utført på i Grasshopper vises i Figur 6.47.



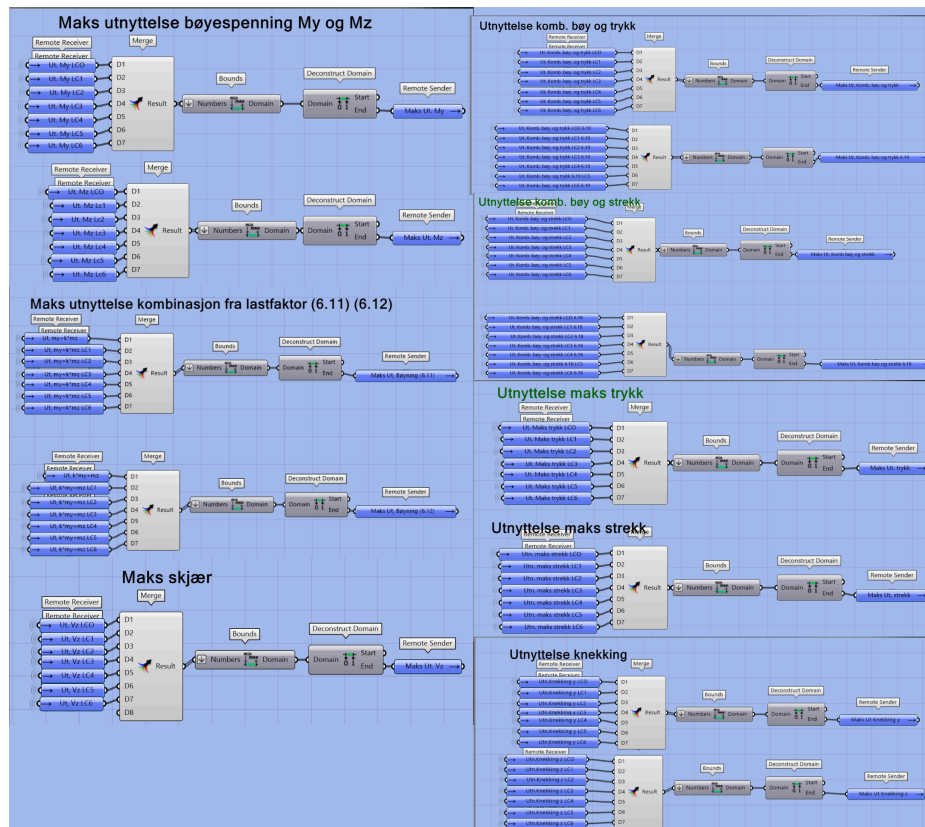
Figur 6.47: Kontroll kombinert bøyning og aksial trykk

I del 9 utføres det en kontroll for knekking både i sterk og svak akse. Oppbyggingen for denne kontrollen er illustrert i Figur 6.48. Knekk lengden som brukes i disse kontrollene er ikke redusert. Det er valgt å utføre analysen med full lengde av elementet, ettersom dette er til sikker side. I Figur 6.48-A og B utføres beregninger etter kapittel 6 punkt 6.3.2 (1) tilsvarende ligning 3.10 fra kapittel 3.4.7. I tillegg indikerer det gule panelet at punkt 6.3.2 (2) tilfredstilles. Resultatet av denne tilfredstillingen har ikke noen påvirkning på beregningen, og eksisterer kun for å vise resultatet. Figur- C utføres utnyttelseskontrollen etter punkt 6.3.2 (3) i EK. Dette tilsvarer ligning 3.2 og 3.3, samt ligning 3.11 - 3.14 fra kapittel 3.4.7.



Figur 6.48: Knekkingskontroll både sterk og svak akse

Del 8 fra Figur 6.38 vises tydeligere i Figur 6.49. I denne figuren vises hvordan utnyttelseskontrollene for alle de utførte kontrollene samles, og hvordan den høyeste utnyttelses verdien av alle lastkombinasjonene velges som output. Dette er gjort ved å bruke «Merge» som lager en liste med alle verdiene og av komponentene «Bounds» og «Deconstruct Domain» som velger den høyeste verdien fra listen. Dette sendes deretter videre til del 9 i Figur 6.38 og er outputene for utnyttelsesberegningsskomponenten som er vist med panelene i samme figur.

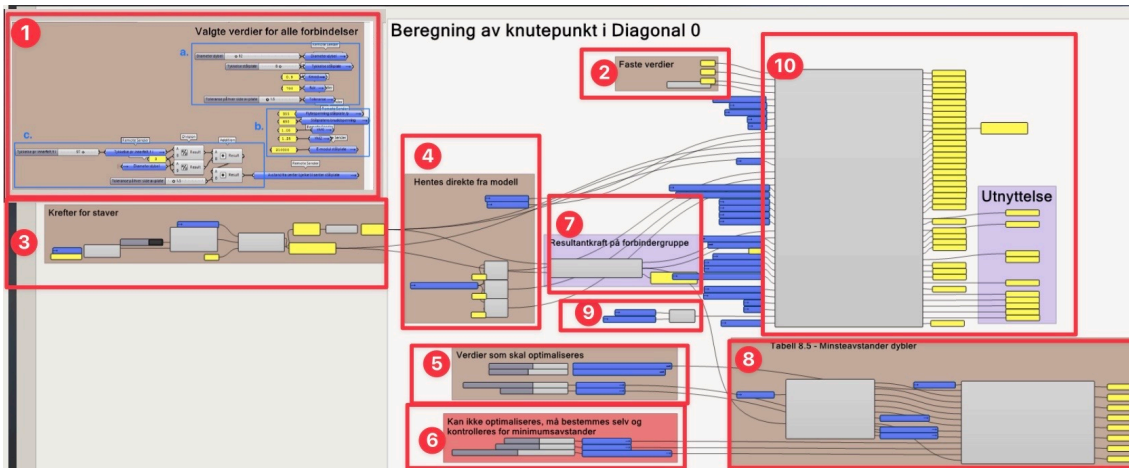


Figur 6.49: Output verdier til utnyttelseskomponenten

6.4.9 Beregning av forbindelser i Grasshopper

Forbindelsene i limtre-fagverket er beregnet etter kapittel 8 i EK5 (NS-EN 1995) [63]. Kapittel 5 i beregningsrapporten til Sweco [116] ble bruk som veiledning for beregningen av dybel-forbindelsene, siden denne tar hensyn til krav fra EK5 og beregninger etter Kolbein Bell kompendium. Enkelte beregningsendringer ble gjort basert på anbefalinger fra rapporten til Vegvesenet [92], Limtreboka beregningseksempel [64], og «Dimensjonering av trekonstruksjoner» av Bell [22]. Stålplatene i dybel-forbindelsen er beregnet etter kapittel 3 i EK3 (NS-EN 1993 1-8) [115] og kapittel 6 i EK3 (NS-EN 1993 1-1) [167]. Limtreboka kapittel 14.2.5 [58] ble bruk som mal for utnyttelsesberegningene av stålplatene.

I kapittel 3.7.3 ble det påpekt at Beaver kunne brukes til utnyttelsesberegning av stål-mot-tre forbindelsen med dybler i henhold til EK5. Det ble derfor forsøkt å benytte Beaver for å utføre en kontroll av stål-mot-tre forbindelser med 1.0.2 versjonen. Forsøket er vist i Figur 6.50, hvor visningen av dyblene er presentert i svart ramme nede til høyre. Fremgangsmåten som ble benyttet er hentet direkte fra eksempelfilen fra food4rhino-nettsiden [153], og tilpasset ved å velge plassering på to av punktene i fagverket. Det viste seg at komponentene kun kan etablere tre-mot-tre forbindelser med to skjærplan. YouTube-kanalen «Beaver Structure» [166] hadde heller ingen videoer om forbindelser. På grunn av dette ble det ikke bygget videre på dette systemet.



Figur 6.51: Visuell programmering av kontrollanalyse av forbindelse

Del 1 - Definerer av verdier som benyttes for alle forbindelser

Et utklipp fra del 1 i Figur 6.51 er presentert i Figur 6.52. I del 1 defineres verdier som brukes i alle forbindelsene. I Figur 6.52 vises de definerte verdiene og hvordan denne delen er oppbygget. Videre følger forklaringer for ramme a, b og c i Figur 6.52.

Første «Number-slider»-parameteren i del «a.», bestemmer diameteren til dybelen. Verdien til denne parameteren er fra 7 mm til 29 mm, og følger dermed kravet som dybeldiameter etter punkt 8.6 (2) fra EK5 [63]. I den andre «Number-slider»-parameteren bestemmes tykkelsen til stålplaten. Det er ikke definert noen krav for tykkelsen til stålplaten i denne parameteren, siden det kun utføres en kontroll for tykk stålplate i denne kontrollen. Brukeren må selv undersøke om denne tykkelsen er innenfor kravet i punkt 8.2.3 (1) fra EK5. I den tredje «Number-slider»-parameteren bestemmer toleransen på her siden av stålplatene i forbindelsen. I den første panelen bestemmes lastvarighetsklassen K_{mod} som brukes i forbindelses kontrollen. Denne vil være lik for alle forbindelser. I den andre panelen defineres bruddspenningen f_u, k til dybelen.

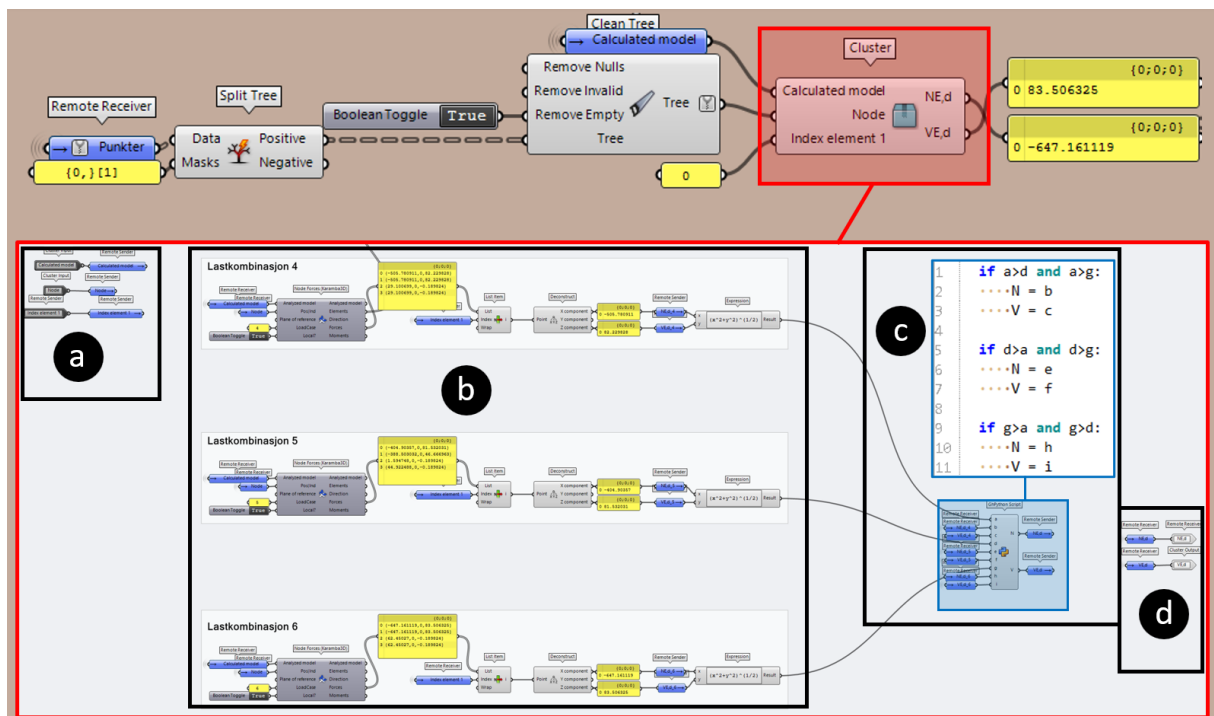
I del «b.» er materialegenskaper, partialkoeffisient og stålets E-modul forbindelsen definert. Den første er f_y , er flytespenningen til stålplaten, den andre er bruddspenningen til stålplaten. De to neste panelene er stålets partialkoeffisient som er gitt i Limtreboka [58], og den siste er E-modulen til stålplaten.

«Number-slider»-parameteren i del «c.», bestemmer tykkelsen på den innerste tredelen mellom stålplatene i forbindelsen. I Ek5 er denne definert som t_2 i punkt 8.2.3 (3). Videre beregnes avstanden fra senteret til elementet til senteret av stålplatene, og brukes i Figur 6.2-10.2 for å visualisere dybelene og stålplaten i Rhino.

Inputene sendes dermed videre til Clusteren som henter ut kreftene som gir den største totalbelastningen for dybelgruppen. Det er valgt å bruke lastkombinasjonene «4», «5» og «6», siden disse har K_{mod} faktor lik 0,9 og vil dermed gi størst belastning. Oppbyggingen av clusteren, er inndelt i rammer med alfabetisk rekkefølge fra «A» til «D». Videre følger forklaring for hver av rammene.

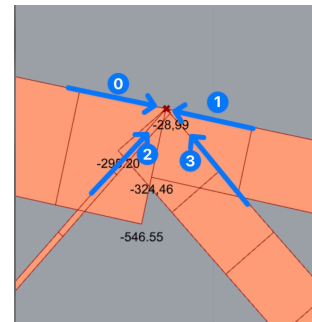
I del «A» blir inputene til clusteren listet opp. Disse overføres direkte videre til del «B» for bruk i komponenten «Node Forces (Karamba3D)». Her blir det spesifisert at første gruppe skal gi kreftene fra fjerde lastkombinasjon, mens de resterende gruppene skal gi kreftene fra henholdsvis femte og sjette lastkombinasjon. Komponentene har også en «Boolean Toggle» satt til «True», for at den skal hente ut krefter i henhold til det lokale koordinatsystemet for elementene. Med dette vil «Node Forces»-komponenten gi belastningskreftene på alle elementene som knyttes til den utvalgte noden. Deretter vil «list item»-komponenten, velge elementet som ble spesifisert ved «Indeks element»-inputen på utsiden av clusteren. Men ettersom kreftene oppgis som koordinater, på følgende vis: (aksialkraft, moment, skjærkraft), benyttes «Deconstruct Point»-komponenten for å skille kreftene fra hverandre. Aksialkraften hentes ut ved «x», mens skjærkraften hentes ut ved «z». Til slutt beregnes resultantkraften som kvadratroten av aksialkraften i annen addert med skjærkraften i annen.

I del «C» samles resultant-, aksial- og skjærkraften for hver lastkombinasjonsberegning i et «Python»-skript. Innsiden av scriptet er vist med blå ramme i Figur 6.54. Programmeringen i scriptet legger til rette for at kreftene som benyttes i dybelgruppen («Ned» og «Ved»), kommer fra lastkombinasjonen som gir størst resultantkraft. Til slutt blir output-verdiene som skal benyttes videre for kontrollen av forbindelsen, listet opp i del «D».



Figur 6.54 Generering av krefter for dybelgrupper i staver og toppunkt (blå cluster)

For dybelgrupper i kontinuerlig over- og undergurt, er de opptredende kreftene programmert i en cluster med brunt ikon, som vist i Figur 6.57. I disse tilfellene vil det være fire elementer som møtes i forbindelsen, som vist i Figur 6.55. Her skal aksialkraften og skjærkraften («NEd» og «VEd»), regnes som differansen mellom kreftene fra venstre- og høyre side av gurten. Dette er i henhold til eksempelet i kapittel 27.8 fra «Limtreboka beregningseksempler» [64]. I tillegg velges en egen skjærkraft for kontroll av splitting («VEd_splitting»). Denne settes som den høyeste skjærkraftverdien, enten fra venstre- eller høyre side av gurten. I motsetning til den blå clusteren, må det i dette tilfellet hentes krefter for 2 elementer, nemlig venstre- og høyre side av gurten. Derfor har clusteren en ekstra inputverdi, hvor indeksen til det andre elementet skal defineres. Dermed vil «Index element 1» bli element 0 og «Index element 2» vil bli element 1, som vist i Figur 6.55.



Figur 6.55 Antall elementer i en node

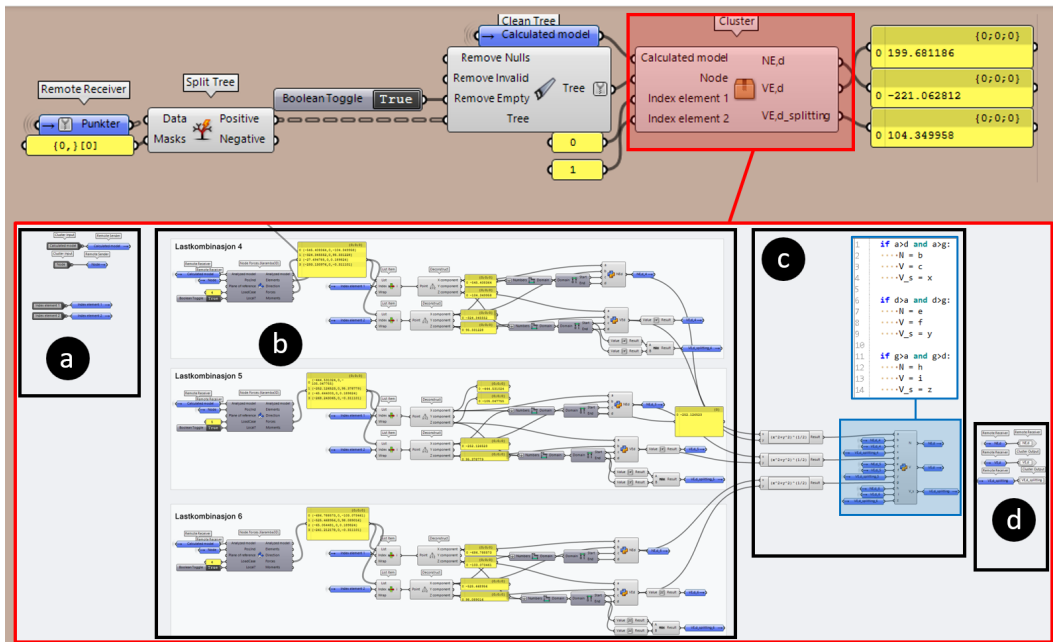
Ettersom det er differansen mellom skjær- og aksialkreftene som skal benyttes, kan ikke kreftene hentes direkte fra «Deconstruct Point»-komponenten (som i den blå clusteren). Derfor er det programmert et script, for både normalkraft og skjærkraft, som vist i Figur 6.56. Innholdet i begge scriptene er likt, det er kun betegnelsene «NEd» og «VEd» som skiller dem.

Scriptene blir forklart med utgangspunkt i skjærkraft-scriptet, i den visshet om at normalkraft-scriptet er utført på tilsvarende måte. For verdiene «a» og «b», hentes skjærkraften fra henholdsvis venstre- og høyre del av gurten. For verdien «c» hentes den laveste skjærverdien, mens for verdien «d» hentes den høyeste. For å finne «c» og «d», benyttes derfor «Bounds» og «Deconstruct domain» komponentene for å hente høyeste og laveste verdi. Scriptet sier dermed at dersom både skjærkraften fra venstre element og høyre element er positive, eller dersom det ene elementet har positiv verdi og det andre negativ verdi, så skal den laveste skjærverdien trekkes fra den høyeste. Men dersom skjærkraften fra både venstre- og høyre element er negative, så skal den høyeste verdien trekkes fra den laveste.

Normalkraft	Skjærkraft
1 if a>0 and b>0:	1 if a>0 and b>0:
2 ...-NEd = d-c	2 ...-VEd = d-c
3	3
4 if a<0 and b<0:	4 if a<0 and b<0:
5 ...-NEd = c-d	5 ...-VEd = c-d
6	6
7 if a>0 and b<0:	7 if a>0 and b<0:
8 ...-NEd = d-c	8 ...-VEd = d-c
9	9
10 if a<0 and b>0:	10 if a<0 and b>0:
11 ...-NEd = d-c	11 ...-VEd = d-c
12	12
13	13
14	14

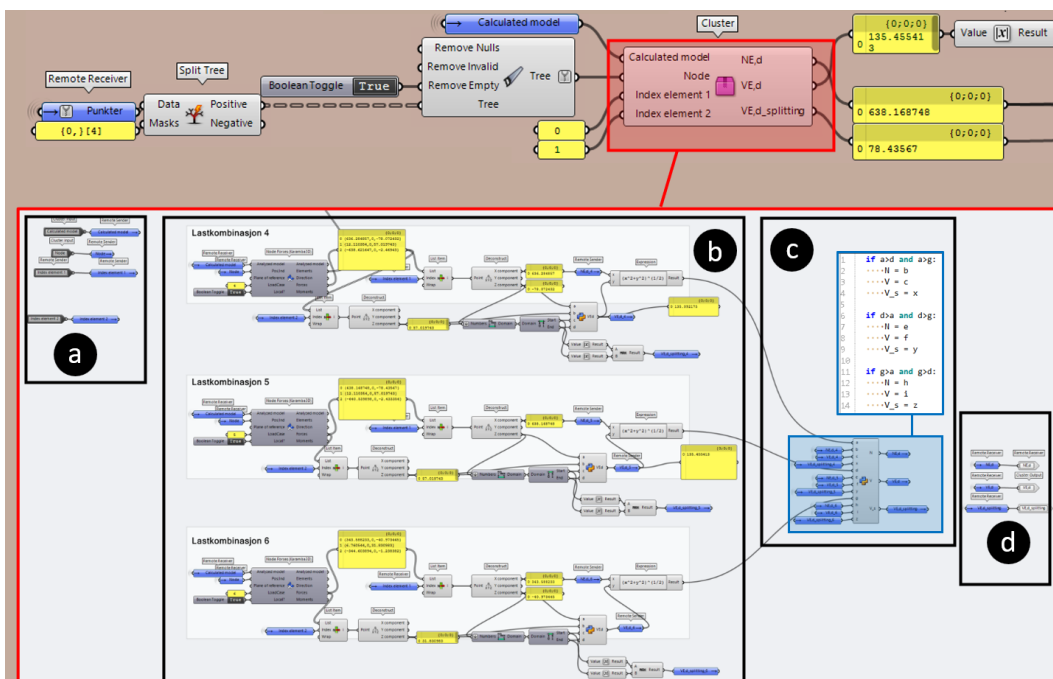
Figur 6.56 GhPython skript for å addere eller subtrahere skjær- og aksialkrefter

Skjærkraften for kontroll av splitting hentes ut som den høyeste absoluttverdien fra begge elementene. Det som skiller del «D» i den brune clusteren fra del «D» i den blå clusteren, er at også skjærkapasiteten som skal benyttes i splittingsberegningen er inkludert i scriptet.



Figur 6.57 Generering av krefter for dybelgrupper i kontinuerlig over- og undergurt (brun cluster)

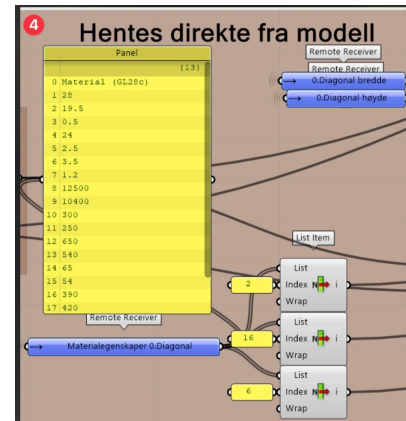
Programmering for den lille clusteren, er presentert i Figur 6.58. Denne benyttes for generering av krefter for dybelgruppe i overgurt, for noden i krysningspunktet mellom over- og undergurt. Basert på verdiene som er benyttet hos oppdragsgiver, har vi antatt at de opptredende kreftene i denne dybelgruppen er hentet på følgende måte: Normalkraften hentes direkte fra øvre del av overgurten, og differansen mellom overgurtsdelene benyttes dermed ikke i dette tilfellet. For normalkraften benyttes derfor samme fremgangsmåte som i den blå clusteren. For skjærkreftene benyttes derimot differansen mellom de to delene av gurten, og dermed er utførelsen på samme måte som i den brune clusteren. Det blir også benyttet en egen skjærkraft for splitting, på samme måte som i den brune clusteren.



Figur 6.58 Generering av krefter for noden i krysningspunktet mellom over- og undergurt (lilla cluster)

Del 4 – Innhenting av material- og tverrsnittsegenskaper

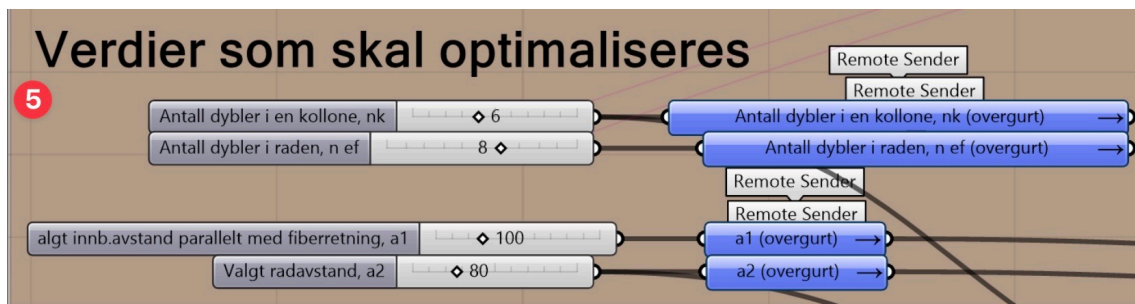
Et utklipp fra del 4 i Figur 6.51, er presentert i Figur 6.59. Her hentes de nødvendige materialfaktorene som skal brukes i dybelkontrollen. Dette er gjort med «list item» komponenten, og verdien i den tilkoblede panelen som henter materialegenskapsverdiene $f_{t,0,g,k}$, $f_{v,g,k}$ og $\rho_{g,k}$ til det valgte materialet. Dette er vist i Figur 6.58. Øverst i samme figur er inputene for dimensjonen til elementet som dybelforbindelsen beregnes etter.



Figur 6.59: Valg av materialfaktorer

Del 5 – Valgmuligheter for antall dybler og innbyrdes avstander

Et utklipp fra del 5 i Figur 6.51, er presentert i Figur 6.60. Her velges antallet av dybler og avstandene mellom dem i forbindelsen av «number slider»- parametere. «nk» angir antall dybler som er i hver kolonne, mens «n ef» angir antall dybler i hver rad. Begge disse parameterne gir verdier fra 2 til 12, men kan forandres etter behov. «a1» er innbyrdes avstand mellom dyblene og «a2» er avstandene mellom dyblene i raden. De minste verdiene i disse parameteren er minimumsavstandene etter tabell 8.5 fra EK5 [63], og må settes inn i parameterne før verdiene kan optimaliseres av optimaliseringsverktøy i Grasshopper.



Figur 6.60: Valg av antall og avstander mellom dyblene

Del 6 – Valgmuligheter for kant- og endeavstander, samt tykkelse på stålplate

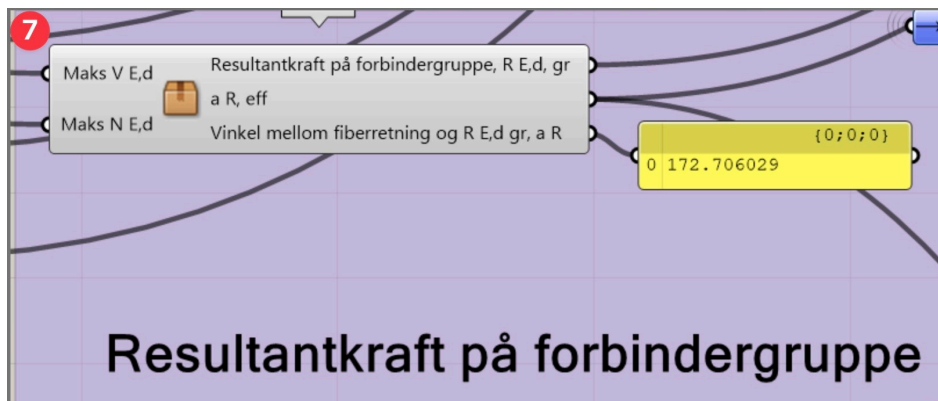
Et utklipp fra del 6 i Figur 6.51, er presentert i Figur 6.61. I denne delen velges kant og endeavstanden fra tredelen til de ytterste dybelene . Disse avstandene brukes også for ubelastet kant og ende. «e1 og e2» er avstanden fra ende av stålplaten til dybelen. Disse verdiene bestemmes og kontrolleres av brukeren selv, og kan ikke optimaliseres av optimaliseringsverktøy i Grasshopper.



Figur 6.61: Valg av ende- og kant avstand fra dybel til tredel og stålplate

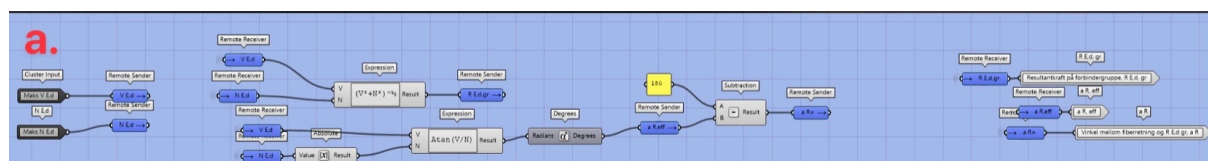
Del 7 – Beregning av resultatkraft og vinkel

Et utklipp fra del 7 i Figur 6.51, er presentert i Figur 6.62. I denne Clusteren beregnes resultatkraften til dybelgruppen og vinkelen mellom kraften og fiberretningen. Inputen kommer fra Figur 6.51-3 (opptredende krefter i dybelgruppen).



Figur 6.62: Cluster for beregning av resultatkraften til dybelgruppen og vinkel mellom fiberretningen

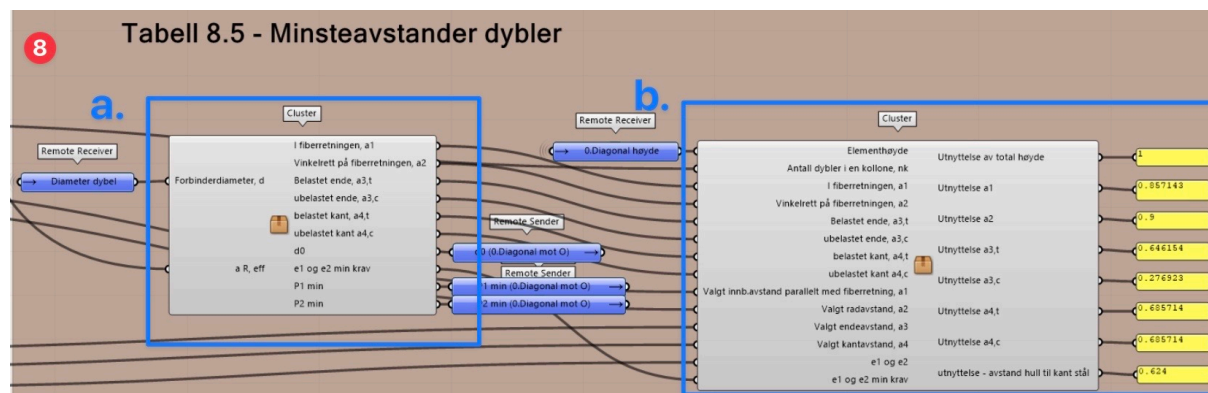
I Figur 6.63 vises den visuelt programmerte beregningene som utføres på insiden av clusteren som omtales ovenfor.



Figur 6.63: Visuelt programmerte beregningen av vinkel og resultatkraft

Del 8 – Beregning og kontroll av minimumsavstander

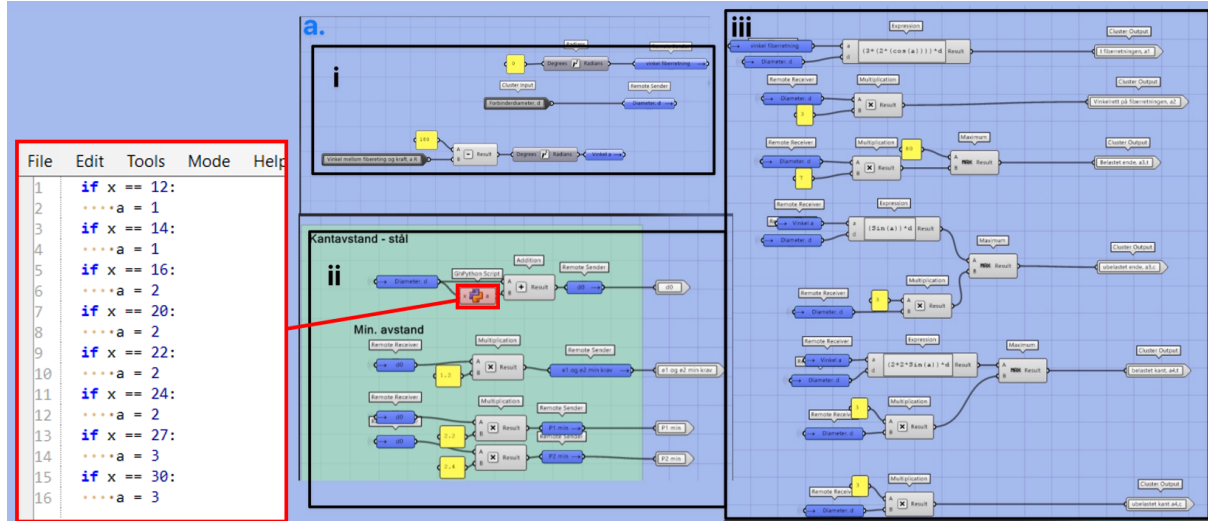
Et utklipp fra del 8 i Figur 6.51, er presentert i Figur 6.64. I denne delen beregnes minimumsavstandene av Figur 6.64-a, mens det i Figur 6.64-b undersøkes om utnyttelsen mellom valgt avstander og kravet til minimumsavstandene. Del b brukes til optimaliseringen og for å undersøke hvor mye stålplassen utnytter høyden til elementet.



Figur 6.64: Beregning av minimumsavstander og utnyttelse

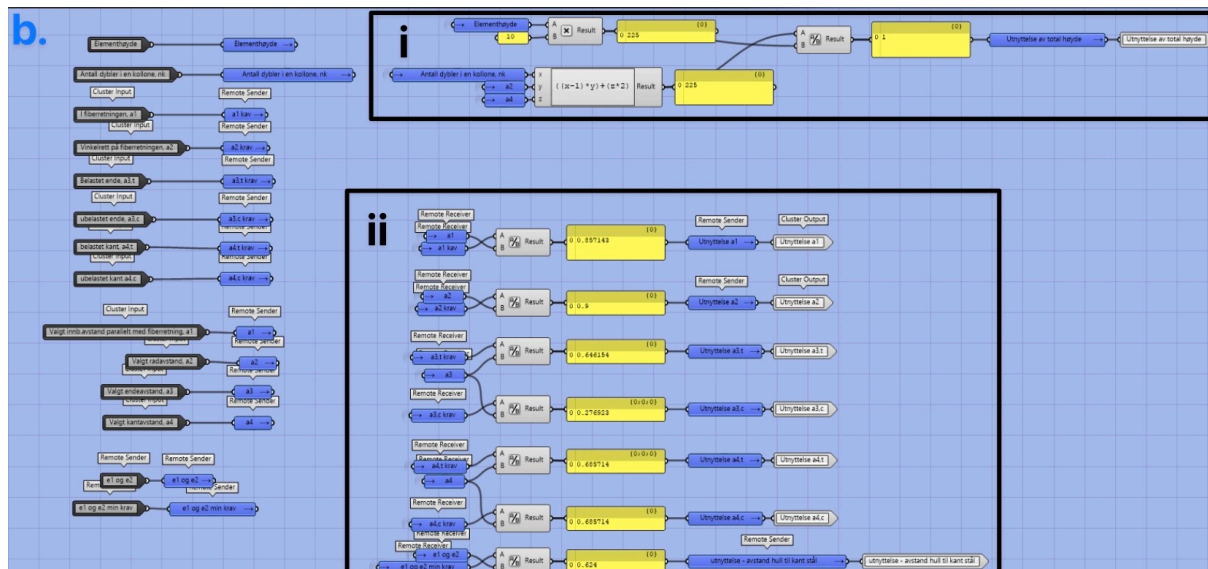
I Figur 6.65 vises den visuelt programmerte beregningen av minimumsavstandene. I Figur 6.65-i ligger inputene som brukes for del «ii» og del «iii». Deretter beregnes minimumsavstander for stålet i

forbindelsen i Figur 6.65-ii. Dette gjøres i henhold til NS-EN 1993 1-8 [115]. I beregningen av d_0 er det brukt et GhPython skript som velger størrelsen på hullene i stålplaten basert på diameteren til dybelen. Dette er basert på verdiene gitt i NS-EN 1993 1-8 [115]. Python skriptet er vist med rød ramme i figuren. Til slutt beregnes minimumsavstandene etter tabell 8.5 i EK5 i Figur 6.65-iii [63].



Figur 6.65: Visuell programmering for beregning av minimumsavstander

I Figur 6.66 vises hvordan Clusteren i Figur 6.64-b er visuelt programmert. I Figur 6.66-i beregnes utnyttelsen mellom høyden til dybelgruppen og høyden til elementet. Beregningen av høyden til dybelgruppen vist i «Expression»-komponenten i del i. I Figur 6.66-ii beregnes utnyttelsene mellom valgte avstander og minimumsavstander.



Figur 6.66: Utnyttelse mellom valgte lengder og minimumsavstander

Del 9 - Valg av stålplatens knekk lengde

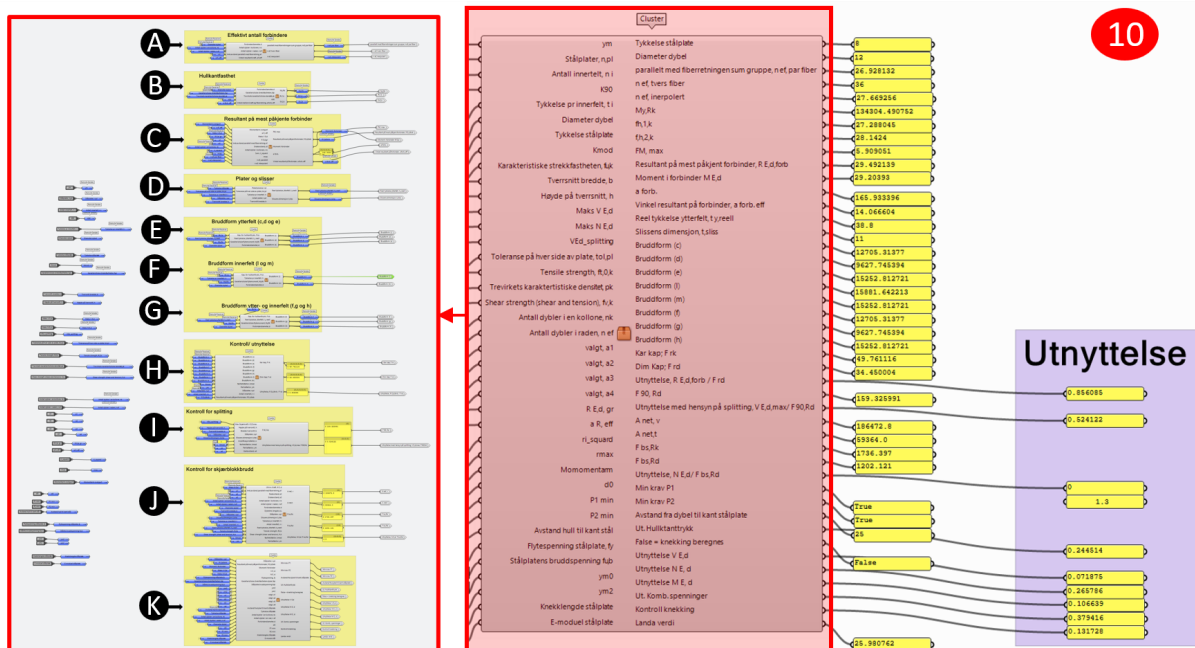
Et utklipp fra del 9 i Figur 6.51, er presentert i Figur 6.67. Her velges lengden mellom stålplatene i forbindelsene, som skal brukes i kontroll for knekking av stålplaten. Denne delen brukes kun for knutepunktene i stavene, men for knutepunktene i over og undergurten brukes den valgte avstanden mellom dybelene parallelt med fiberretningen a_1 .



Figur 6.67: Valg av knekk lengde for stålplate i stavene

Del 10 – Samling av alle knutepunkts-kontroller

Et utklipp fra del 10 i Figur 6.51, er presentert i Figur 6.68. Dette er komponenten som beregner utnyttelse av dybelforbindelsen i limtrefagverket. Den utfører kontroll av stål-mot-tre forbindelser i henhold til EK5 [63], samt kontroll av stål materialet i henhold til limtreboka og EK3 [58, 115, 167]. I denne Clusteren er det totalt 11 andre Clusterer som utfører nødvendige beregninger for kontrollen. Hver av disse er markert alfabetisk, som vist til venstre i Figur 6.68, og videre følger forklaringer for hver av disse delene.

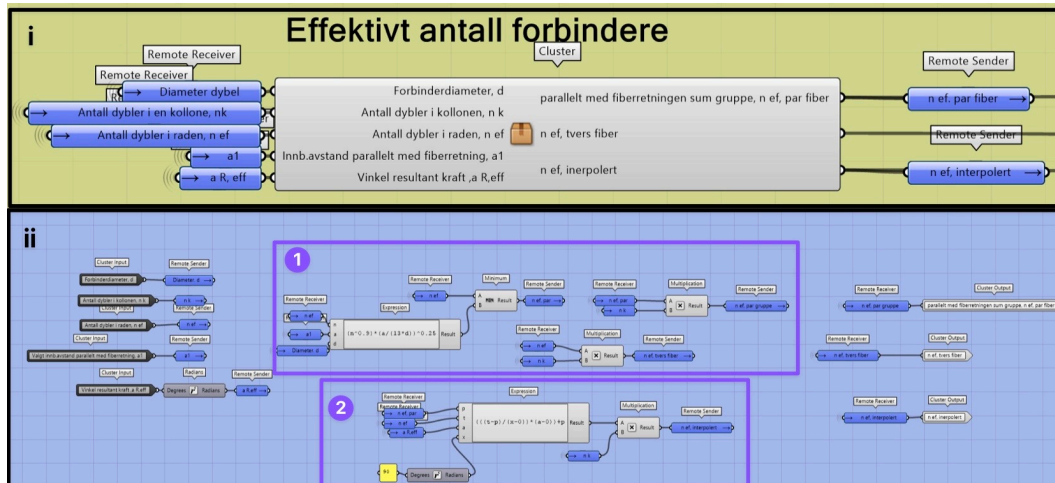


Figur 6.68: Komponent for alle kontroller av dybelforbindelse

Gruppe A fra Figur 6.68, beregner effektivt antall av dybler i forbindelsen. I Figur 6.69-i vises selve clusteren som utfører denne beregningen, og i Figur 6.69-ii vises den visuelt programmerte beregningen. I Figur 6.69-ii viser del 1 beregningen for n_{ef} fra punkt (8.34) i EK5 [63]. I «Expression»-komponenten i Figur 6.69-ii-2 er n_{ef} lineært interpolert etter punkt (8.35) i EK5 [63]. Dette er utført på samme måte som i kapittel 5 i beregningsrapporten fra Sweco [116], og beregnes som i formel 6.3:

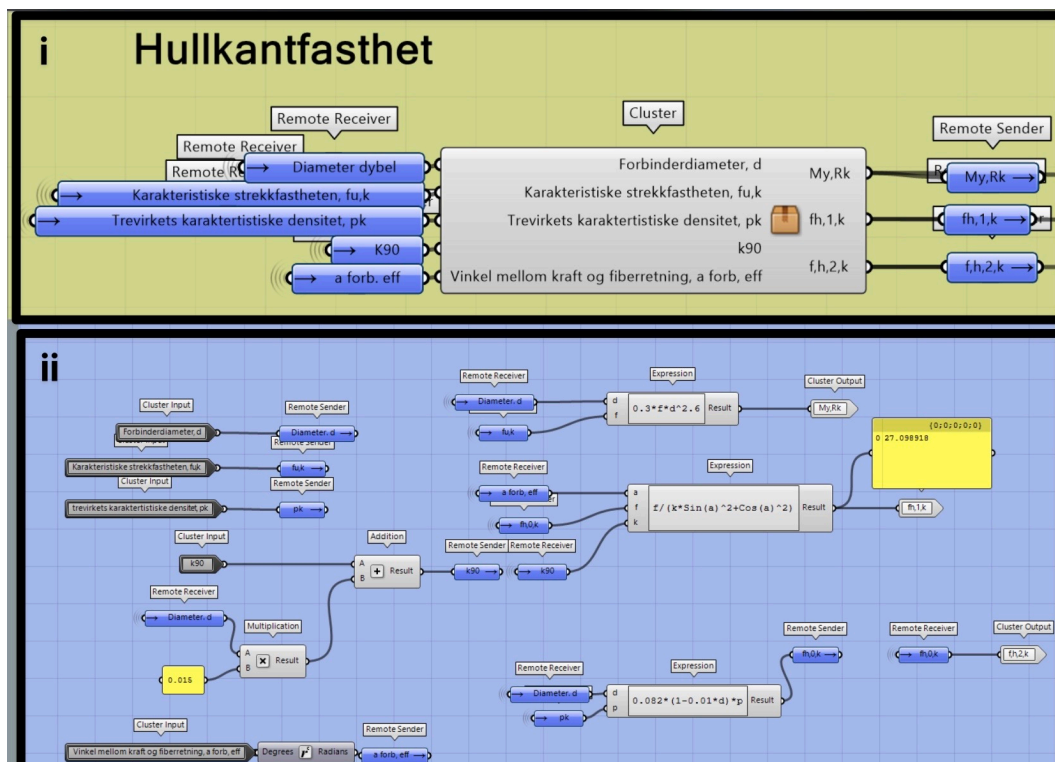
$$\left(\left((n_{ef} - n) / (90 - 0) \right) * (\alpha - 0) \right) + n_{ef} \tag{6.3}$$

Vinkel α er vinkel mellom resultantkraften til dybelgruppen og fiberretningen.



Figur 6.69: Cluster og beregning av effektivt antall forbindere

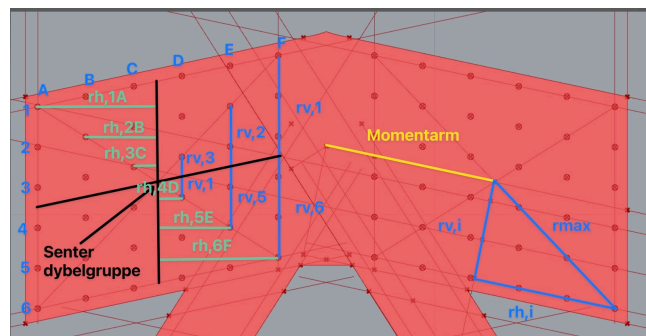
Gruppe B fra Figur 6.68, er vist i Figur 6.70, hvor Figur 6.70-i viser Clusteren som beregner den karakteristiske hullkantfastheten og den karakteristiske verdien for flytemomentet. Den visuelt programmerte beregningen er illustrert i Figur 6.70-ii, der $M_{y,Rk}$ beregnes etter punkt 8.30 i EK5 [63]. Hullkantfasthetene $f_{h,\alpha,k}$ og $f_{h,0,k}$ fra punkt 8.31 og 8.32 i EK5 [63] er $f_{h,1,k}$ og $f_{h,2,k}$ i Figur 6.70-ii.



Figur 6.70: Beregning av hullkantfasthet og flytemomentet

I gruppe C fra Figur 6.68, beregnes resultantkraften på den dybelen som har størst belastning, se Figur 6.72. Del «i» viser utsiden av clusteren, mens del «ii» viser innsiden hvor beregningene utføres. I del 1 i Figur 6.72-ii beregnes vinkelen mellom resultantkraften til den mest påkjente forbinderen og fiberretningen til tredelen. Deretter velges den største vinkelen mellom vinkelen til resultantkraften til dybelgruppen og vinkelen til resultantkraften til den mest påkjente dybelen. Grunnen til dette er for å alltid skal velge den høyeste vinkelen som er på fiberretningen. I de fleste forbindelsene er det ikke alltid moment, og med denne oppbyggingen velges det vinkelen som gir lavest hullkantfasthet som ble beregnet i Figur 6.70.

I Figur 6.72-ii-2 beregnes momentet som oppstår når stivhetssenteret til forbindelsen ikke er i senteret til dybelgruppen. Momentarmen som oppstår på grunn av dette er illustrert i Figur 6.71, og denne avstanden måles i clusteren for visualisering av stålplatene og dyblene som vil forklares i Kapittel 6.4.10. Momentet beregnes ved å multiplisere momentarmen med skjærkraften i forbindelsen. I Figur 6.72-ii-3 utføres det en elastisk beregning for å finne resultantkraften til den mest påkjente dybelen, i henhold til momentet i forbindelsen. Denne beregningen er basert på formlene i kapittel 11.9 i boka «Dimensjonering av trekonstruksjoner» [22] og kapittel E14.2 fra «Timber Engineering» boka [57].



Figur 6.71: Illustrasjon for beregning av $F_{M,max}$

Formelen for å finne belastningen på forbindelsen i en momentoverførende forbindelse er presentert i formel 6.4:

$$F_{M,max} = \left(\frac{M_{E,d}}{\sum r_i^2} \right) * r_{max} \quad (6.4)$$

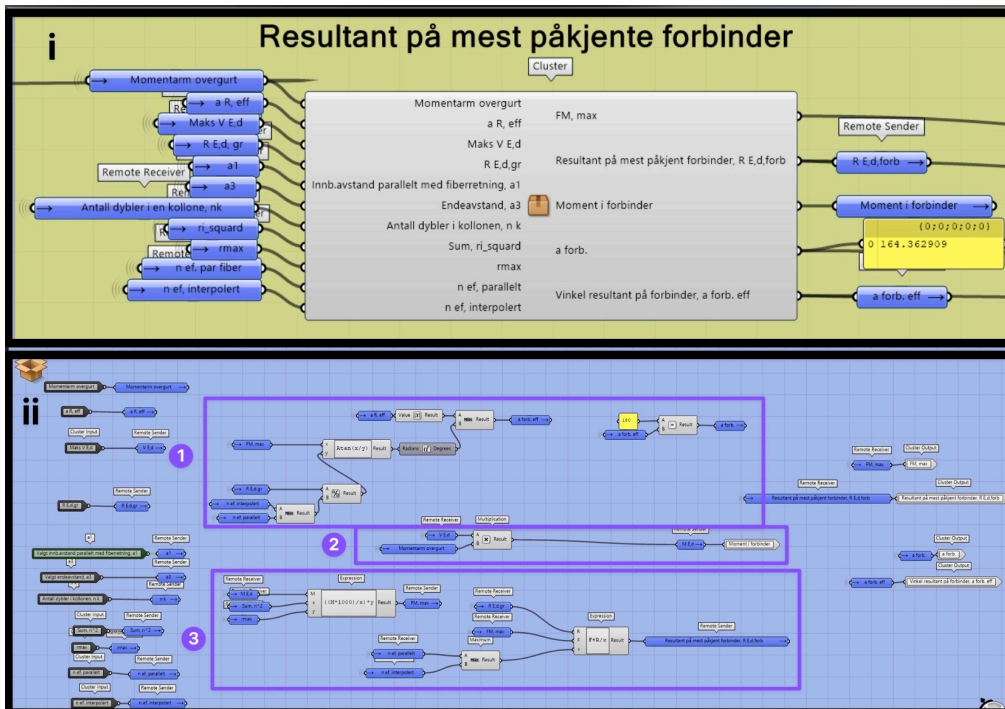
r_{max} er den diagonale avstanden fra senteret til dybelgruppen til den ytterste dybelen, og beregnes med hypotaguras av den lengste vertikale og horisontale avstanden fra senteret av dybelgruppen. Dette er illustrert i Figur 6.71 og formelen er gitt som presentert i formel 6.5.

$$r_{max} = \sqrt{r_{h,i}^2 + r_{v,j}^2} \quad (6.5)$$

Beregningen av $\sum r_i^2$ er summen av alle vertikale og horisontale avstander fra hver dybel til senteret av dybelgruppen. Basert på antall dybler i Figur 6.71, blir denne formel som presentert i formel 6.6.

$$\sum r_i^2 = \sum (r_h^2 + r_v^2) = \sum_i \left(\sum_{j=A}^F (r_{h,ij}^2) + r_{v,i}^2 \right) \quad (6.6)$$

Både r_{max} og $\sum r_i^2$ måles og beregnes basert på disse formlene i clusteren for visualiseringen av stålplaten og dyblene. I kapittel 6.4.10 blir dette forklart videre. I Figur 6.72-ii-3 summeres $F_{M,max}$ og resultantkraften til dybelgruppen som er delt på antall effektive dybler i fobrindelsen. Ettersom punkt 8.5.1.1. (4) i EK5 [63] påpeker at den interpolerte n_{ef} skal brukes når det er vinkel mellom last og fiberretningen. Resultantkraften til dybelgruppen blir delt på den høyeste verdien av $n_{ef,parallel}$ og $n_{ef,interpolert}$. Dette er ikke nødvendig, ettersom $n_{ef,interpolert}$ alltid vil være høyest når det er en vinkel mellom lasten og fiberretningen. Når det er ikke er en vinkel blir disse verdiene alltid like. Dette ble likevel gjort, ettersom dette utføres i beregningsrapporten til Sweco [116].



Figur 6.72: Komponent for beregning av resultantkraft på mest påkjente dybel

I gruppe D fra Figur 6.68, beregnes dimensjonen til slissen og den effektive tykkelsen til ytterfeltet. Clusteren som utfører dette er vist i Figur 6.73-i, mens den visuelt programmerte beregningen er vist i Figur 6.73-ii. I Figur 6.73-ii-1 beregnes den effektive tykkelsen basert på anbefalinger fra rapporten til Vegvesenet [92], masteroppgaven om dybelforbindelse i fagverksbruer [79] og forelesningsnotater fra BYG514 (2021) [94]:

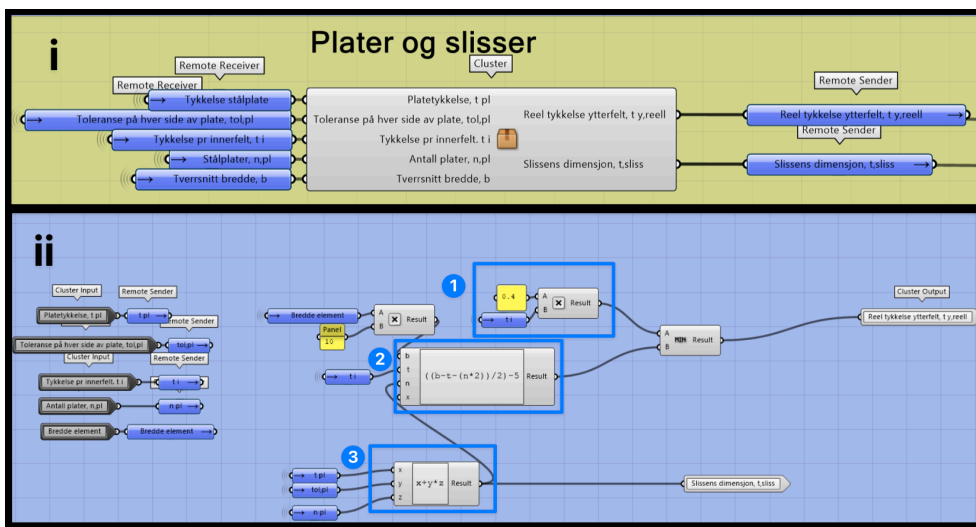
$$t_y = 0,4 * t_i (\text{innerfelt}) \tag{6.7}$$

I Figur 6.73-ii-2 beregnes den faktiske tykkelsen på ytterfeltet og har en korrigeret for spisset ende på 5 mm. I denne oppgaven er tykkelsen på innerfeltet alltid 97 mm, stålplaten 8 mm og toleransen på hver side av platen 1,5 mm. Dette gjør at i enkelte tilfeller vil den effektive tykkelsen være større enn

den reelle tykkelsen. Ved å bruke «minimum»-komponenten, vil ikke dette problemet oppstå. Tykkelsen på ytterfelter bestemmes som presentert i formel 6.8. Denne er lik formel 3.22.

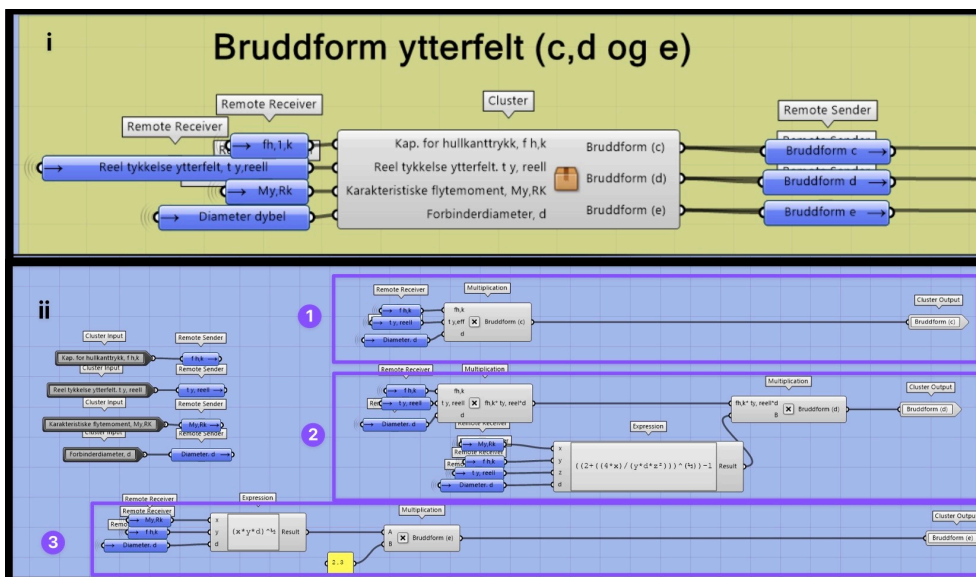
$$t_y = \min \left\{ \begin{array}{l} 0,4 * t_i \text{ (innerfelt)} \\ ((b - t_i \text{ (innerfelt)}) - (s * n)) / 2 - 5 \end{array} \right. \quad (6.8)$$

«n» representerer antall ståplater og «s» representerer dimensjonen på slissene. Tallet to er antallet på ytterfeltet og tallet fem er korrigeringen for spisset ende. I Figur 6.73-ii-3 beregnes slissens dimensjon.



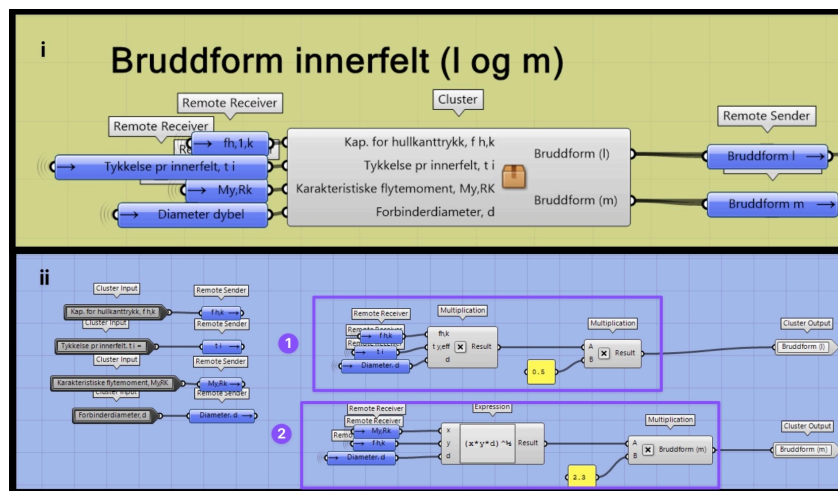
Figur 6.73: Beregning av slissens dimensjon og effektiv tykkelse ytterfelt

I gruppe E fra Figur 6.68, beregnes bruddformene for ytterfelt med tykkstålplate i dybelforbindelsen. Dette er bruddform c, d, og e fra 8.2.3 (3) i EK5 [63]. Clusteren for denne beregningen er vist i Figur 6.74-i og den visuelt programmerte beregningen er vist i Figur 6.74-ii. Bruddform c er vist i Figur 6.74-ii-1, bruddform d er vist i Figur 6.74-ii-2 og bruddform e Figur 6.74-ii-3.



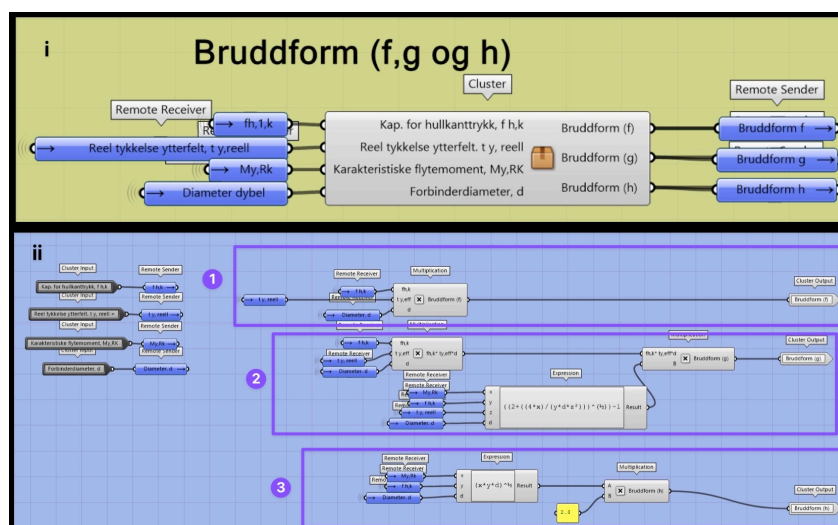
Figur 6.74: Visuelt programmert beregning av bruddform c, d og e

I gruppe F fra Figur 6.68, beregnes bruddformen for innerfeltet til dybelforbindelsen med tykkstålplate. Dette er bruddform l og m fra 8.2.3 (3) i EK5 [63] som vises i Figur 6.75. Figur 6.75-i viser Clusteren som beregner dette og i Figur 6.75-ii vises den visuelt programmerte beregningen til bruddformene. I Figur 6.75-ii-1 vises beregningen for bruddform l, mens Figur 6.75-ii-2 viser beregningen for bruddform m.



Figur 6.75: Visuelt programmert beregning av bruddform l og m

I gruppe G fra Figur 6.68, blir bruddformene f, g og h beregnet i henhold til punkt 8.2.3 (3) i EK5 [63]. Selv om det egentlig ikke er nødvendig å beregne disse bruddformene, ettersom de beregnes på samme måte som bruddformene c, d og e, ble de likevel tatt med i modellen. Dette kommer av at bruddform h er med i beregningsrapporten til Sweco (vedlegg), og på grunn av en metode for utnyttelsesberegning som ble lært i faget BYG514 (2022), inkluderte disse bruddformene. Denne metoden gikk ut på å multiplisere den laveste bruddformen av f, g, h, l og m med antall skjærplan i dybelforbindelsen. Denne metoden brukes ikke i oppgaven, men det er valg å beholde bruddform f, g og h i modellen selv om de ikke er relevante for den valgte utnyttelsesmetoden. Clusteren som utfører beregningen av bruddformen, er vist i Figur 6.76-i og Figur 6.76-ii viser den visuelt programmerte beregningen. I del «ii», vises bruddform f i del «1», bruddform g i del «2» og bruddform del «3».

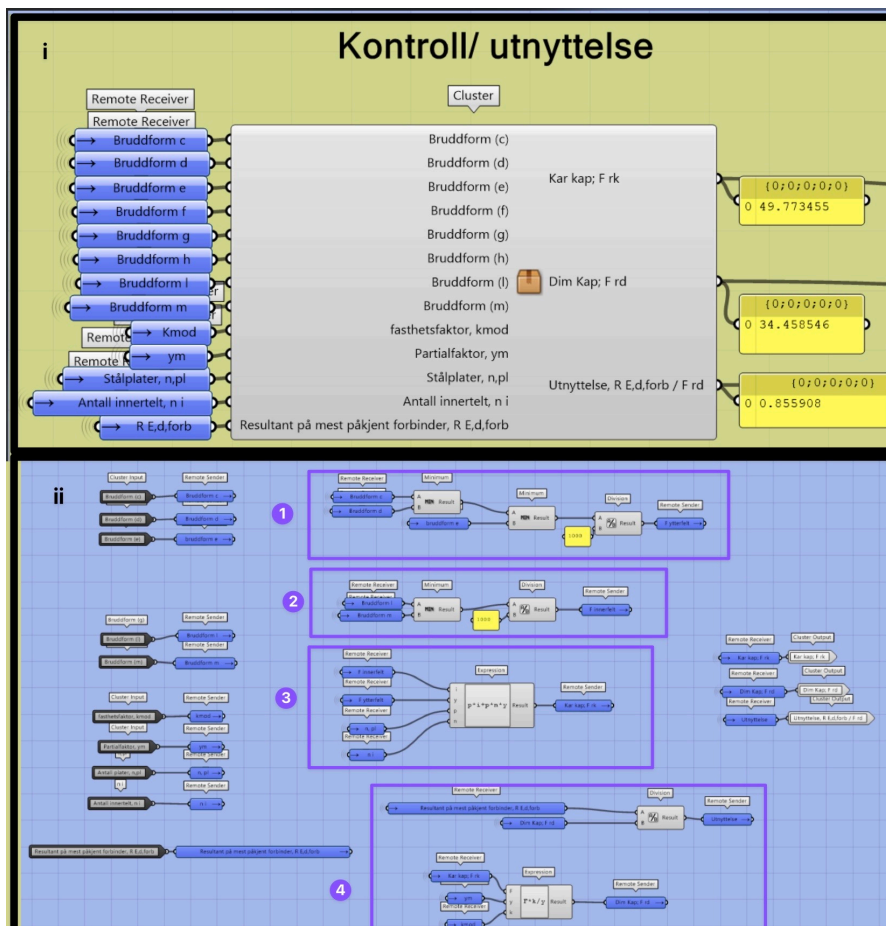


Figur 6.76: Visuelt programmert beregning av bruddform f, g og h

I gruppe H fra Figur 6.68, blir utnyttelsen, karakteristiske, og dimensjonerende kapasiteten til en dybel beregnet. Beregningsmetoden for å finne den karakteristiske kapasiteten til dybelen er basert på beregningen fra beregningsrapporten fra Sweco (vedlegg). Denne metoden summerer dimensjonerende bruddformen for ytterfeltet og innerfeltet, der hver av dem er multiplisert med antall skjærplan. Formelen er presentert i (6.9).

$$F_{dybel} = 2 * F_{ytterfelt} + 2 * n_i * F_{innerfelt} \tag{6.9}$$

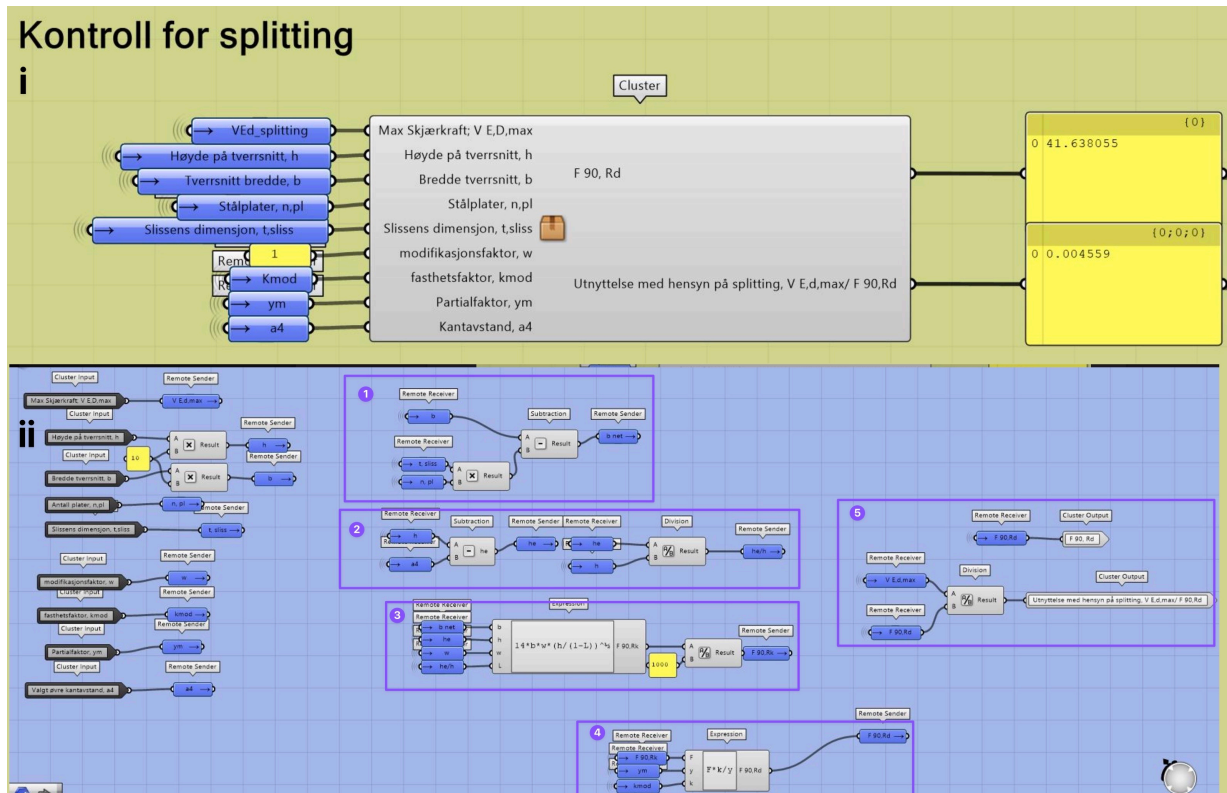
Tallet 2 er antall skjærplan for hver av feltene, og n_i er antall innerfelt. I Figur 6.77 vises komponenten som utfører utnyttelses kontrollen av dybelen. I del «i» vises Clusteren som er laget for å utføre denne beregningen. I del «ii» vises de visuelt programmerte beregningene i Clusteren. I Figur 6.77-ii-1 velges den dimensjonerende bruddformen for ytterfeltet, mens i Figur 6.77-ii-2 velges den dimensjonerende bruddformen for innerfeltet. I Figur 6.77-ii-3 beregnes den karakteristiske kapasiteten som presentert i formel 6.8, der tallet 2 erstattes med antall ståplater. I den nederste delen av Figur 6.77-ii-4 beregnes den dimensjonerende kapasiteten til dybelen, og i øverst i samme figur beregnes utnyttelsen av dybelforbindelsen.



Figur 6.77: Utnyttelse/kontroll av dybelforbindelsen

I gruppe «I» fra Figur 6.68, utføres det en splittingskontroll av forbindelsen. Denne kontrollen er beregnet i henhold til beregningskontrollen fra beregningsrapporten til Sweco (vedlegg), som gjelder kun for gran og furu. I Figur 6.78-i vises Clusteren som utfører denne beregningen, og Figur 6.78-ii

viser den visuelt programmerte beregningen. I Figur 6.78-ii-1 beregnes netto bredden til tverrsnittet, ved å subtrahere antall stålplater med størrelsen på slissene. I Figur 6.78-ii-2 beregnes forholdet mellom $\frac{h_e}{h}$, og i Figur 6.78-ii-3 beregnes den karakteristiske splittingskapasiteten. I Figur 6.78-ii-4 beregnes den dimensjonerende splittingskapasiteten, og i Figur 6.78-ii-5 beregnes utnyttelsen.



Figur 6.78: Kontroll splitting

I gruppe «J» fra Figur 6.68, utføres det en kontroll for skjær-blokkbrudd, og er beregnet i henhold til beregningsrapporten til Sweco (vedlegg). I Figur 6.79-i vises komponenten som utfører denne kontrollen. I Figur 6.79-ii- vises de visuelt programmerte delen som beregner kontrollen. I Figur 6.79-ii-1 beregner netto tykkelse for tredelen i forbindelsen ($t_{net,1}$), der innerfeltet og ytterfeltene summeres. Her brukes den effektive tykkelsen til ytterdelen som ble beregnet i Figur 6.79, ettersom denne tykkelsen også brukes i kapasitetsberegningen av forbindelsen. I Figur 6.79-ii-2 beregnes netto tykkelsen til dybelen ($t_{net,2}$), som beregnes ved å subtrahere antall med størrelsen og antall slisser. I Figur 6.79-ii-3 beregnes gjennomsnittlig belastet endeavstand ($l_{v,1}$) og i Figur 6.79-ii-4 beregnes gjennomsnittlig innbyrdes avstand ($l_{v,i}$). I Figur 6.79-ii-5 beregnes gjennomsnittlig antall dybler i hver rad (n), og er utføres ved å dele det totale antallet av dybler på antall dybler i hver rad. I Figur 6.79-ii-6 beregnes den totale nettolengden på skjærbruddarealet, og denne er gitt i beregningsrapporten til Sweco som:

$$L_{net,v} = \sum l_{v,i} = 2 * l_{v,1} + 2 * (n_r - 1) * l_{v,i} - 2 * (n - 0,5) * d \quad (6.10)$$

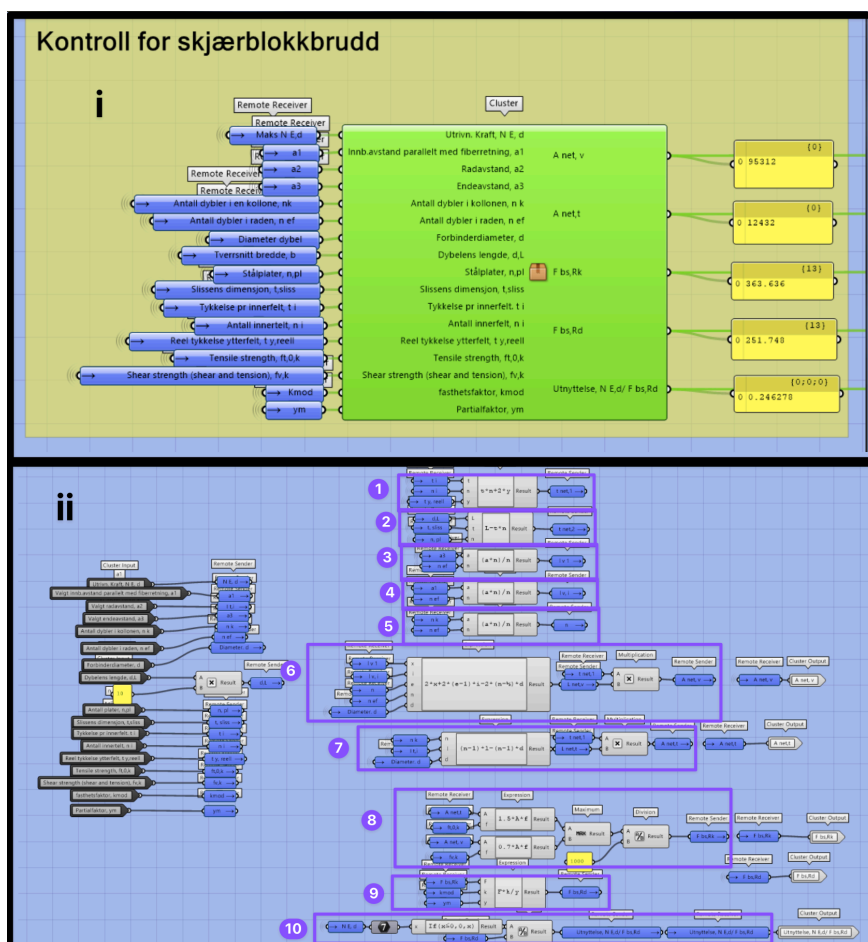
Her er (n_r) antall rader med dybler, og d er dybeldiameter. Videre i samme figur beregnes netto skjærareal. I Figur 6.79-ii-7 beregnes netto bredde av tverrsnittet som er vinkelrett på fiberretningen ($L_{net,t}$). Formelen på denne er gitt:

$$L_{net,t} = \sum l_{t,i} = (n_r - 1) * l_{t,i} - (n_r - 1) * d \tag{6.11}$$

Videre i samme figur beregnes netto tverrsnittsareal som er vinkelrett på fiberretningen. I Figur 6.79-ii-8 beregnes den karakteristiske blokkskjærkapasiteten. Denne beregningen er gitt slik:

$$F_{bs,Rk} = \max \left\{ \begin{matrix} 1,5 * A_{net,t} * f_{t,0,k} \\ 0,7 * A_{net,v} * f_{v,k} \end{matrix} \right. \tag{6.12}$$

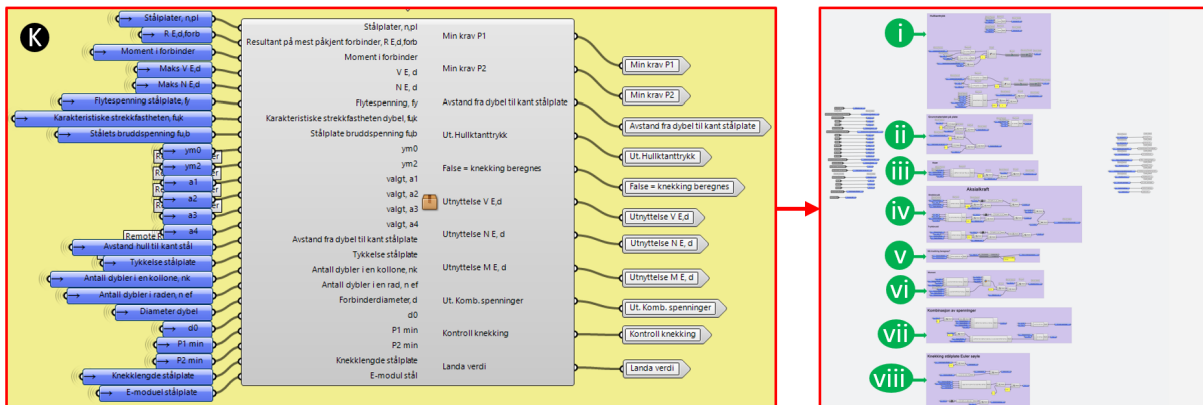
I Figur 6.79-ii-9 beregnes den dimensjonerende kapasiteten ($F_{bs,Rd}$), som brukes til utnyttelseskontrollen i Figur 6.79-ii-10. I del «10» ble «lf»-funksjon brukt som gjør at aksialkraften blir null dersom den er i trykk, og gjør at utnyttelsen blir null. Når den er i strekk er verdien positiv, slik at kontrollen beregnes.



Figur 6.79: Kontroll skjærblokkbrudd

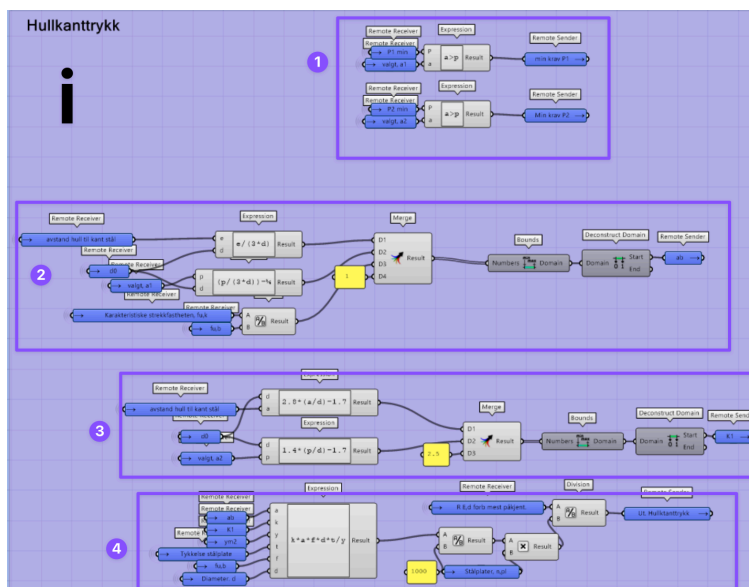
I gruppe «K» fra Figur 6.68, kontrolleres stålet i forbindelsen i henhold til kapittel 14.2.5 i Limtreboka og EK3. Cluster-komponenten utfører kontroll av brudd i stålplaten, som er hullkanttrykk, strekkbrudd, trykkbrydd, bøyebrydd, skjærbrudd, kombinasjon av spenninger og knekkingskontroll

av stålplaten. Hver av disse er markert med romertall, som vist til høyre i Figur 6.80, og videre følger forklaringer for hver av disse delene.



Figur 6.80: Cluster for kontroll av stål

Gruppen med romertall «i» er vist i Figur 6.81, hvor beregningskontrollen for hullkantrykk i stålplaten blir utført. I Figur 6.81-1 blir det gjort en undersøkelse på om de valgte avstanden mellom dyblene oppfyller minstekravet i henhold til Ek3 [115]. I Figur 6.81-2 og Figur 6.81-3 velges verdien for α_b og k_1 i henhold til Ek3 [115]. I Figur 6.81-4 utføres kontrollen for hullkantrykket med resultantkraften på mest påkjente forbinder. Kapasiteten til hullkantrykket er multiplisert med antall stålplater.



Figur 6.81: Beregning av hullkantrykk stålplate

Gruppen med romertall «ii» er vist i Figur 6.82, og her illustreres beregningen for å finne høyden og bredden til stålplaten. Beregning for å finne høyden ser slik ut:

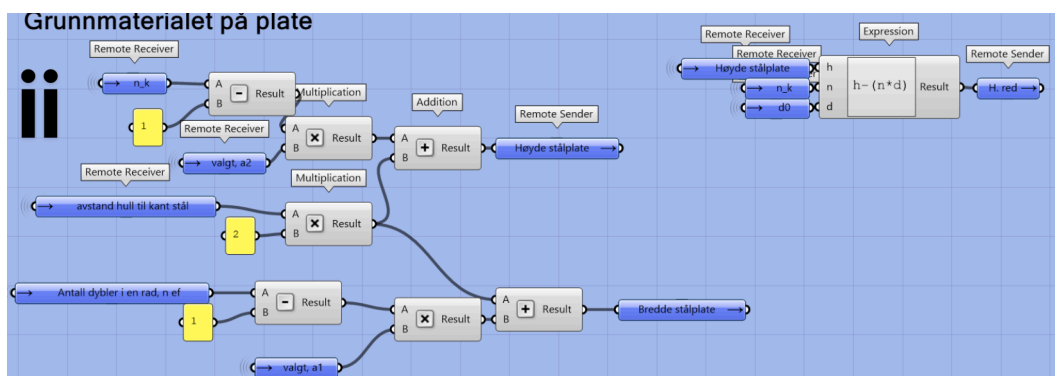
$$(n_k - 1) * a_2 + (e_1 * 2) \tag{6.13}$$

I denne formelen er n_k antall dybler i en kolonne, mens verdien a_2 er den valgte innbyrdes normalt på fiberretningen, og e_1 avstand fra ytterste dybel til enden av stålplaten. Til høyre i figur beregnes

den reduserte høyden på grunn av dybelhullene i stålplaten (h_{red}). Nederst i figuren beregnes lengden til stålplaten, som er navngitt bredde i «telepathy»-senderen. Formelen for lengden er gitt:

$$(n_{ef} - 1) * a_1 + (e_1 * 2) \tag{6.14}$$

I denne formelen er n_{ef} antall dybler i en rad. a_1 er valgt innbyrdesavstand mellom dyblene parallelt med fiberretningen.

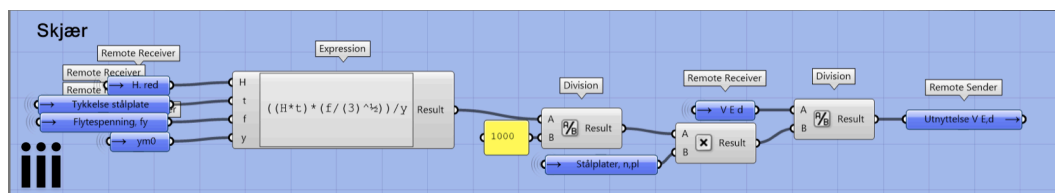


Figur 6.82: Redusert stålplatehøyde

Gruppen med romertall «iii» er vist i Figur 6.83, og her vises den visuelt programmerte kontrollen for skjærbrudd i stålplaten. Formelen er basert på at det er flytning i hele tverrsnittet til platen, og blir beskrevet som vist i formel 6.15:

$$V_{c,Rd} = V_{pl} = \frac{A_v * (f_y / \sqrt{3})}{\gamma_{M0}} \tag{6.15}$$

A_v er skjærarealet og er beregnet av den reduserte høyden og tykkelsen til stålplaten. f_y er stålplatenes flytespenning. Dette er formel (14-18) i Limtreboka [58]. Videre multipliseres skjærkapasiteten med antall stålplater, og brukes i utnyttelsesberegningen for skjær.



Figur 6.83: Skjærbrudd stålplate

Gruppen med romertall «iv» er vist i Figur 6.84, og her utføres kontrollen for strekk- og trykkbrudd i stålplaten i henhold limtreboka [58]. I Figur 6.84-iv-1 utføres kontrollen for lastbæringskapasiteten til det fulle tverrsnittet til stålplaten. Dette er formel 14-9 i limtreboka [58] og er gitt som:

$$N_{pl,Rd} = \frac{f_y * (t * h)}{\gamma_{M0}} \tag{6.16}$$

t er tykkelsen til stålplaten og h er stålplatens fulle høyde. Videre blir kapasiteten multiplisert med antall stålplater. For at det ikke skal beregnes for tykk i strekkkontrollene i Figur 6.84-iv-1 og 2, er det brukt en «if» kommando i «Expression»-komponenten som gjør at N_{Ed} blir null dersom den er negativ. I kontrollen i Figur 6.84-iv-4 brukes også «if» kommandoen, bare her blir N_{Ed} null dersom verdien er positiv og en «absolute»-komponent brukes for å gjøre utnyttelsesverdien positiv. Dette gjør at dersom det er trykk i stålplaten, vil strekkkontrollen bli null, og dersom det er strekk, blir trykkkontrollen null. I Figur 6.84-iv-5 blir den relevante normalkraftverdien valgt av «Maximum»-komponenten.

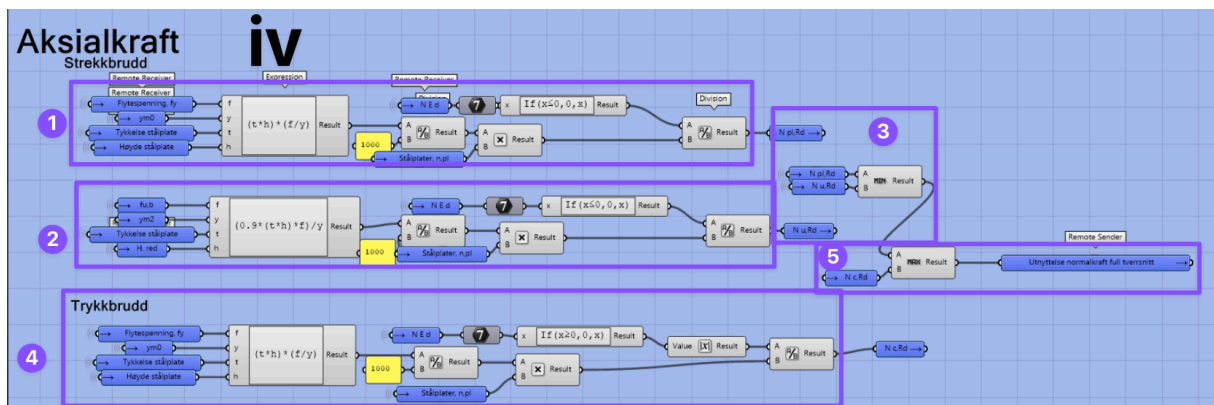
Ettersom det er to bruddmuligheter i strekk, utføres kontrollen for kapasiteten til nettotverrsnittet i Figur 6.84-2. Dette er formel 14-10 i limtreboka [58], og er definert slik:

$$N_{u,Rd} = \frac{0,9 * f_u * (h_{red} * t)}{\gamma_{M2}} \tag{6.17}$$

Her er $(h_{red} * t)$ nettotverrsnittet til stålplaten, og videre multipliseres kapasiteten med antall stålplater. I Figur 6.84-iv-3 velges den minste kapasiteten av $N_{pl,Rd}$ og $N_{u,Rd}$. I Figur 6.84-iv-4 utføres kontrollen for trykkbrudd i stålplaten i henhold til formel 14-11 i limtreboka [58], og er definert som følge:

$$N_{c,Rd} = \frac{A * f_y}{\gamma_{M0}} \tag{6.18}$$

Her er brukt stålplatens fulle tverrsnitt, ettersom dybelene fyller hullene.



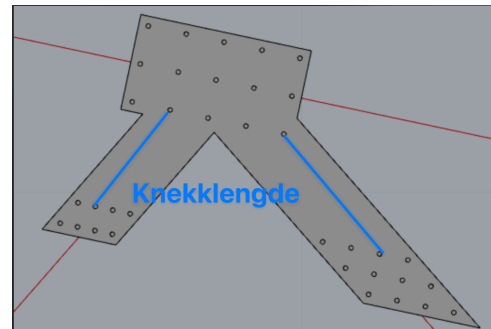
Figur 6.84: Trykk- og strekkbrudd i stålplaten

Gruppen med romertall «v» er vist i Figur 6.87, og her utføres det en kontroll for å undersøke om stålplaten må kontrolleres for knekking. I Limtreboka er det presisert i formel 14-12 at dersom a_1 (avstanden mellom dybelene parallelt med fiberretningen) er mindre enn følgende verdi:

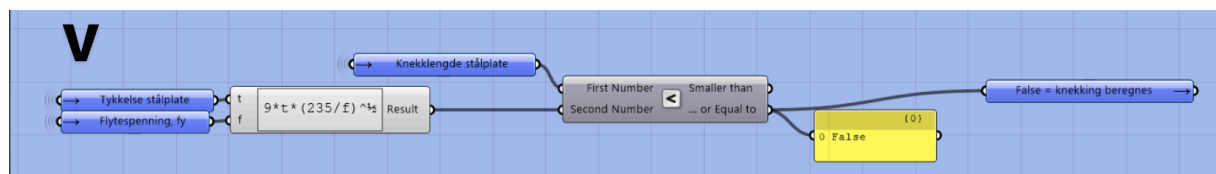
$$9t * \varepsilon = 9t * \sqrt{\frac{235}{f_y}} \tag{6.19}$$

- så skal platen kontrolleres for knekking.

Dette utføres for alle knutepunktene i over- og undergurtne, men for knutepunktene i stavene er det valgt å heller bruke den lengste avstanden på stålplatene i forbindelsen. Dette er vist i Figur 6.85. Grunnen til dette er at denne lengden vil knekke før a_1 avstanden. Dersom knekkingsavstanden mellom stålplatene er mindre enn verdien i formel 6.19, må stålplaten antas å være en Euler søyle, og knekking må kontrolleres. I Figur 6.86 vises knekkingskontrollen.



Figur 6.85: Eksempel på knekkengde i stålplate



Figur 6.86 Undersøkelse om knekking i platen må kontrolleres

I Gruppen med romertall «vi», som vist i Figur 6.88, kontrolleres stålplaten for bøyebrydd. Formelen som er definert i limtreboka [58] er:

$$M_{pl} = \frac{f_y * b * h^2}{4} \quad (6.20)$$

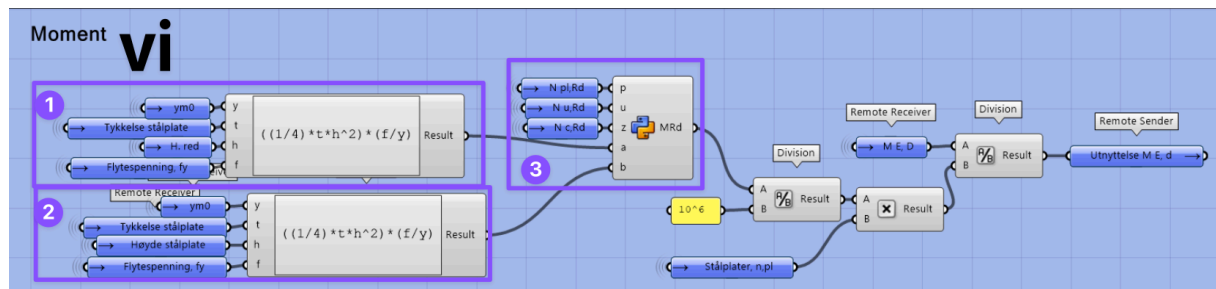
Denne formelen er benyttet i begge beregningene som er vist i vist i Figur 6.88-vi-1 og 2. I Figur 6.88-vi-1 beregnes det for et fullt rektangulært stålplate tverrsnitt, mens Figur 6.88-vi-2 beregner for et redusert rektangulært stålplate tverrsnitt. I Figur 6.88-vi-3 velges den relevante bøyebryddkapasiteten til stålplaten, basert på verdiene til $N_{pl,Rd}$ og $N_{u,Rd}$. Dersom $N_{pl,Rd} \geq N_{u,Rd}$ er det ikke nødvendig å ta hensyn til hullene og verdien fra M_{pl} i Figur 6.88-vi-1 velges av GhPython skriptet. I motsatt tilfelle når $N_{pl,Rd} \leq N_{u,Rd}$ velges M_{pl} fra beregningen i Figur 6.88-vi-2. Dette er basert på formel 14-17 i limtreboka [58]. For at riktig M_{pl} verdi skal velges når det et trykk i stålplaten, velger GhPython skriptet bøyekapasiteten for det fulle stålplate tverrsnittet, som er beregningen i Figur 6.88-vi-2. GhPython skriptet vises i Figur 6.87. Videre i Figur 6.88 omgjøres kapasiteten fra newton til kilonewton og multipliseres med antall stålplater.

```

Grasshopper Python Script Editor
File Edit Tools Mode Help
1  if p<u:
2      ...MRd = b
3  if p>u:
4      ...MRd = a
5  if z>p and z>u:
6      ...MRd = b

```

Figur 6.87: Python skript for valg av bøyekapasitet i stålplaten



Figur 6.88 Bøyekapasitet i stålplaten

I Gruppen med romertall «vii», som vist i Figur 6.89, utføres det en kontroll for kombinerte spenningstilstander. Denne kontrollen inneholder både normal- og skjærspenninger, og formelen defineres slik i Limtreboka [58]:

$$\left(\frac{\sigma_{x,Ed}}{f_y/\gamma_{M0}}\right)^2 + \left(\frac{\sigma_{z,Ed}}{f_y/\gamma_{M0}}\right)^2 - \left(\frac{\sigma_{x,Ed}}{f_y/\gamma_{M0}}\right)\left(\frac{\sigma_{z,Ed}}{f_y/\gamma_{M0}}\right) + 3\left(\frac{\tau_{Ed}}{f_y/\gamma_{M0}}\right) \leq 1.0 \quad (6.21)$$

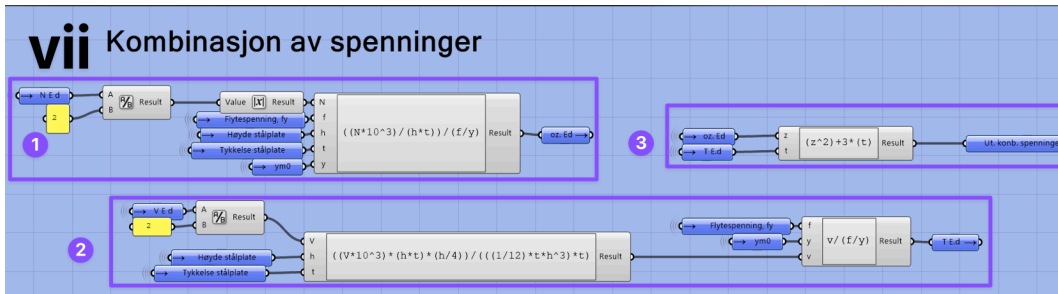
Denne formelen er lik formel 6.1 i EK3 1-1 [167]. Beregningen av dimensjonerende normalspenning i lengderetningen av stålplaten ($\sigma_{x,Ed}$) gjøres i Figur 6.89-vii-1. Ettersom normalkraftene ikke vil gå tverretningen på stålplaten, beregnes ikke dimensjonerende normalspenning ($\sigma_{z,Ed}$). Formelen for beregning av $\sigma_{x,Ed}$ er utført slikt:

$$\sigma_{x,Ed} = \left(\frac{N_{Ed}}{H_{stålplate} * t_{stålplate}}\right) \quad (6.22)$$

I Figur 6.89-vii-2 beregnes den dimensjonerende skjærspenningen (τ_{Ed}), denne er beregnet etter formel 6.19 fra EK3 1-1 [167], og er gitt slik:

$$\tau_{Ed} = \left(\frac{V_{Ed} * S}{I * t}\right) \quad (6.23)$$

Videre i Figur 6.89-vii-2 deles skjærspenningen på stålets flytespenning og partialkoeffisient til stålet. I Figur 6.89-vii-3 utføres kontrollen av kombinasjonen av spenningene fra formel. I alle kontrollene utført i Figur 6.89-vii er normalkraften og skjærkraften delt på antall stålplater.

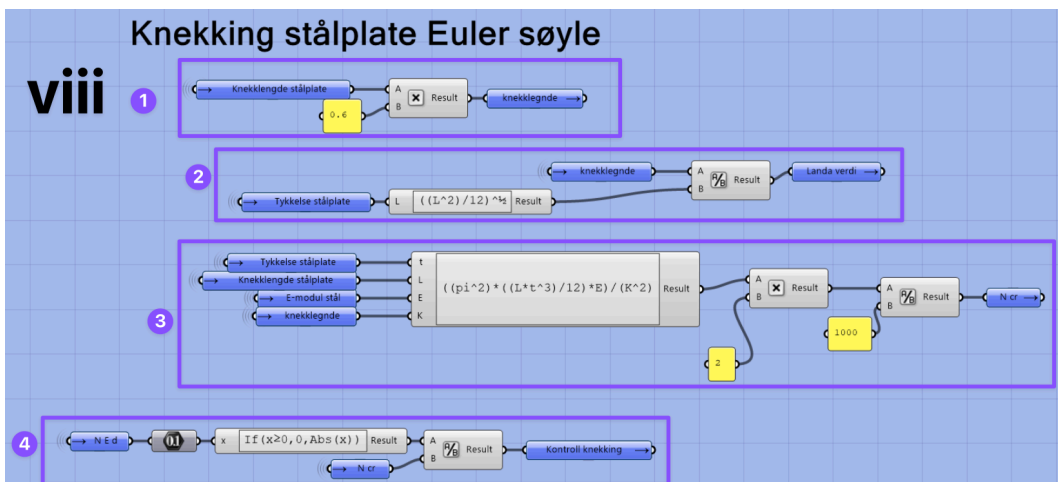


Figur 6.89 Kontroll kombinasjon av spenninger i stålplaten

I Gruppen med romertall «viii», som vist i Figur 6.90, utføres knekkingskontroll av stålplate. Denne beregningen er basert på at stålplaten er som en Euler søyle, siden dette presiseres i Limtreboka. I Figur 6.90-viii-1 multipliseres knekk lengden med 0,6, ettersom dette er nevnt i Limtreboka [58]. Knekk lengden vil være valgt a_1 avstand mellom dyblene i knutepunktene i under- og overgrutene, eller knekkavstanden som ble vist i Figur 6.85. Denne blir målt i visualiseringen av dyblene og stålplatene. I Figur 6.90-viii-2 beregnes først treghetsradiusen av stålplaten, og deretter slankhetsverdien som er navngitt λ . I Figur 6.90-viii-3 beregnes den ideelle kraften for den relevante knekkformen N_{cr} denne er gitt slik:

$$N_{cr} = \frac{\pi^2 * \left(\frac{L_{Knekk lengde} * t^3}{h} \right) * E_{modul}}{L_{redusert Knekk lengde}} \tag{6.24}$$

Denne multipliseres med antall stålplater. I Figur 6.90-viii 4 utføres kontrollen av knekklegden. Det er brukt en kommando «lf» på normalkraften for at knekkingen ikke skal beregnes, hvis det er det er strekkraft i forbindelsen.



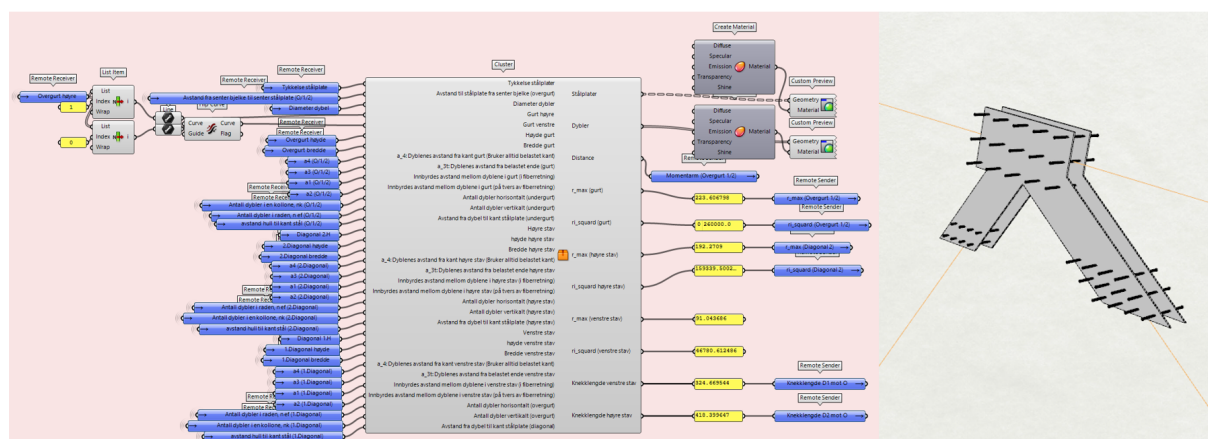
Figur 6.90 Knekkingskontroll av stålplate

6.4.10 Modellering av forbindelser med stålplater og dybler

I de lys-rosa gruppene dannes visningen av alle forbindelser med dybler og stålplater. Vi har bygget fem forskjellige Clustere for de ulike forbindelses-typene. Clusterene kan kopieres og brukes på motsatt side av fagverket, men da må «flip curve» benyttes for at start- og endepunktene på elementene er riktig i henhold til clusterens oppbygging. Videre i dette kapittelet følger forklaringer på hvordan hver av clusterene ble bygget. Den oransje clusteren for kontinuerlig overgurt med to tilknyttede staver blir beskrevet grundig steg for steg. Ettersom de resterende clusterene stort sett kan beskrives på samme måte som den oransje, vil det bare trekkes fram programmering som er særegen for modellering av disse.

Forbindelser i kontinuerlig overgurt tilknyttet 2 staver (Oransje cluster)

Forbindelsene i kontinuerlig overgurt med to tilknyttede staver, har dybelgruppens senter plassert midt i krysningspunktet hvor overgurtene og diagonalene møtes. Denne clusteren er markert med oransje farge på ikonet, som vist i Figur 6.91.



Figur 6.91 Cluster for forbindelse i kontinuerlig overgurt tilknyttet 2 staver (oransje)

Figur 6.92 viser oppbygging av den oransje clusteren. De gule gruppene tilhører dybelgruppen i overgurt, mens de grønne og blå clusterene tilhører henholdsvis høyre og venstre stav. I tillegg viser de røde gruppene programmering som tilhører både stavene og overgurt.



Figur 6.92 Forbindelser i kontinuerlig overgurt tilknyttet 2 staver (Oransje cluster)

Figur 6.93 viser rekkefølgen på operasjonene som utføres for hver av elementene i den oransje clusteren. Dette illustreres med nummererte, fargede rammer.



Figur 6.93 Nummerert fremgangsmåte for forbindelse i kontinuerlig overgurt tilknyttet 2 staver (Oransje cluster)

Tabell 6.5 beskriver innholdet i hver av de nummererte rammene fra Figur 6.93. Fargene i tabellen viser til fargene i figuren.

Tabell 6.5 Beskrivelse av rekkefølgen på operasjonene som utføres for modellering av forbindelse

RAMME	BESKRIVELSE AV GRUPPENS INNHOLD
1	Input-verdiene listes opp. Disse verdiene beskriver stålplate-tykkelse, stålplatens avstand fra senter, dybel-diameter og dyblenes avstander, samt elementenes systemlinjer, høyder og bredder.
2	Avstandene for stålplatens høyde og bredde beregnes, basert på avstandene til dyblene.
3	Denne rammen gjelder kun stavene, og definerer endepunktet til staven. Dette benyttes som utgangspunkt for stålplatelengde og plassering av dybelgruppen i stavene.
4	Her etableres linjer som definerer stålplatens ytterkanter (både parallelt- og vinkelrett med elementets fiberretning).
5	Samling av stålplatens hjørner- og vendepunkter fra alle elementer, som blir sortert i løpende rekkefølge langs stålplatens ytterkant. I tillegg er det programmert funksjoner for noen av punktene som bare skal eksistere i enkelte tilfeller.
6	Samlingen fra ramme 5 omgjøres til en sammenhengende linje, og deretter til en flate (surface). Denne kopieres og skaper utgangspunktet for to stålplater, som plasseres riktig i forhold til elementenes senterlinje. Til slutt dannes et volum som gir flaten riktig fasong (som en stålplate).
7	Etablerer et punkt i krysningspunktet mellom stavene og overgurt. Dette midtpunktet benyttes som utgangspunkt for dybelgruppen.
8	Lager et rutenett med punkter hvor dyblene skal plasseres. Deretter bygges en sylinder i hver av punktene som gir dyblene riktig utseende.
9	Måler avstander og beregner r_i _squared og r _max, som benyttes videre for beregning av resultatant på mest påkjente forbinde.
10	Måling av elementets lengste knekk lengde for å kunne kontrollere stålplaten for knekking.
11	Output-verdiene listes opp. Nødvendige outputer omfatter visning av stålplatene (volumene) og visning av dyblene (sylindrene), samt verdier som er nødvendige for eurokodekontroller.

Figur 6.94 viser et utklipp fra andre ramme, hvor avstandene for stålplatens høyde og bredde beregnes. I dette tilfellet er det hentet ut beregningene for overgurt, siden utførelsen i stavene er nokså lik. Det som er annerledes ved beregningene for stavene, er markert med rødt.

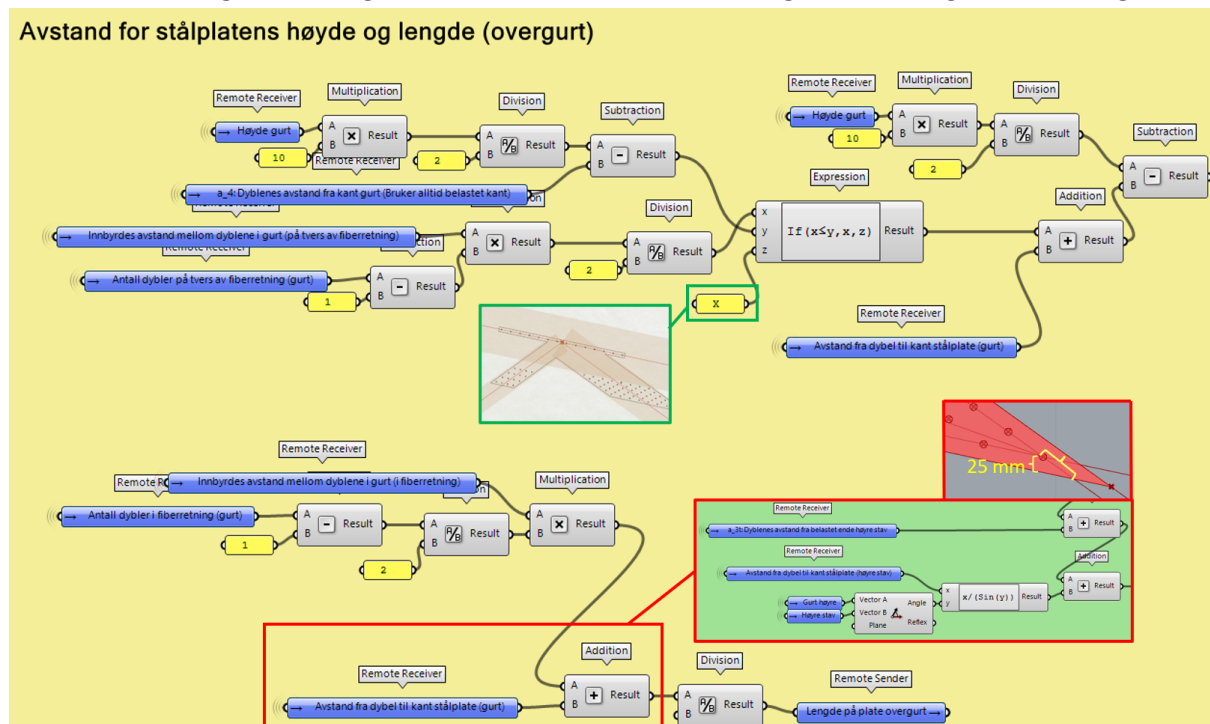
Øverst i Figur 6.94 beregnes høyden til stålplaten. I dette tilfellet defineres høyden som avstanden fra systemlinje til platens øvre- eller nedre kantlinje. For å finne denne, multipliseres valgt innbyrdes avstand mellom dyblene med antall dybler på tvers av fiberretning. Dermed deles verdien på to, fordi en kun skal bruke halve høyden til stålplaten (avstand fra systemlinje til platens ytterkant). For at systemet også skal overholde avstanden til kant bjelke, kan ikke høyden til stålplatene overstige tre-

bjelkens høyde subtrahert med kantavstand. Derfor er det etablert en kommando, hvor høyden blir definert som «X» dersom kantavstanden overskrides. Dette vil føre til at hele forbindelsen krympes sammen, som vist med grønn utheving. På denne måten vil en få en tydelig visuell tilbakemelding, dersom man har tastet inn verdier som ikke er oppnåelige.

Så langt i beregningen, vil stålplate-høyden rekke fra senterlinje øverste dybel. Her må det tillegges en overdekning, og derfor adderes høydeverdien med en bestemt avstand fra senter dybel til kant stålplate. Til slutt subtraheres stålplatehøyden fra bjelkens totale høyde, ettersom det er avstand mellom kant bjelke og stålplate som benyttes ved videre modellering.

Nederst i Figur 6.94 beregnes lengden til stålplaten. I dette tilfellet defineres lengden som avstanden fra midtpunktet av forbindelsen til platens ytre kantlinje (målt parallelt med fiberretning). For å finne denne, multipliseres valgt innbyrdes avstand mellom dyblene med antall dybler i fiberretning. Dermed deles verdien på to, fordi en kun skal bruke halve lengden til stålplaten (avstand fra forbindelsens midtpunkt og utover). Også her, må det tillegges en overdekning. Derfor adderes lengden med den bestemte avstanden fra senter dybel til kant stålplate. Til slutt divideres verdien med 1000, fordi lengden skal oppgis i meter.

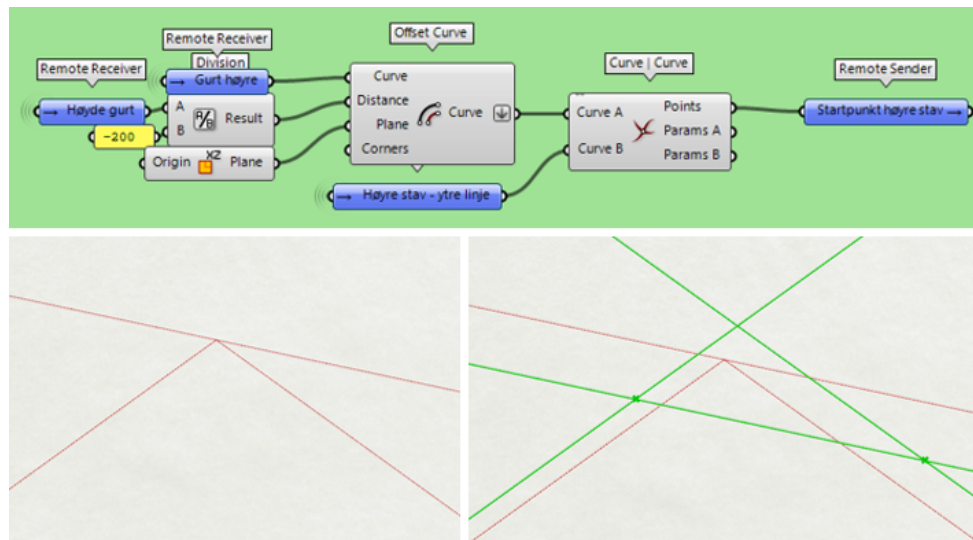
For stavene må en legge til en liten justering ved stålplatens overdekning, som uthevet med rødt i Figur 6.94 Ettersom stålplatens nedre kant skal ligge parallelt med overgurtens fiberretning, vil overdekningen ikke overholdes i global z-retning. Derfor benyttes trigonometriske sammenhenger, slik at overdekningen ikke lenger tilsvarer avstanden i fiberretning, men heller global z-retning.



Figur 6.94 Beregning av avstander for stålplatens høyde og bredde (ramme 2)

I ramme 3 bestemmes endepunktet til stavene som vist i Figur 6.95. De diagonale stavene vil kuttes ved overgurtens nedre kantlinje, og dermed må dybelgruppens endeavstand måles fra dette punktet. Overgurtens nedre kantlinje dannes med komponenten «Offset Curve». Denne tar utgangspunkt i

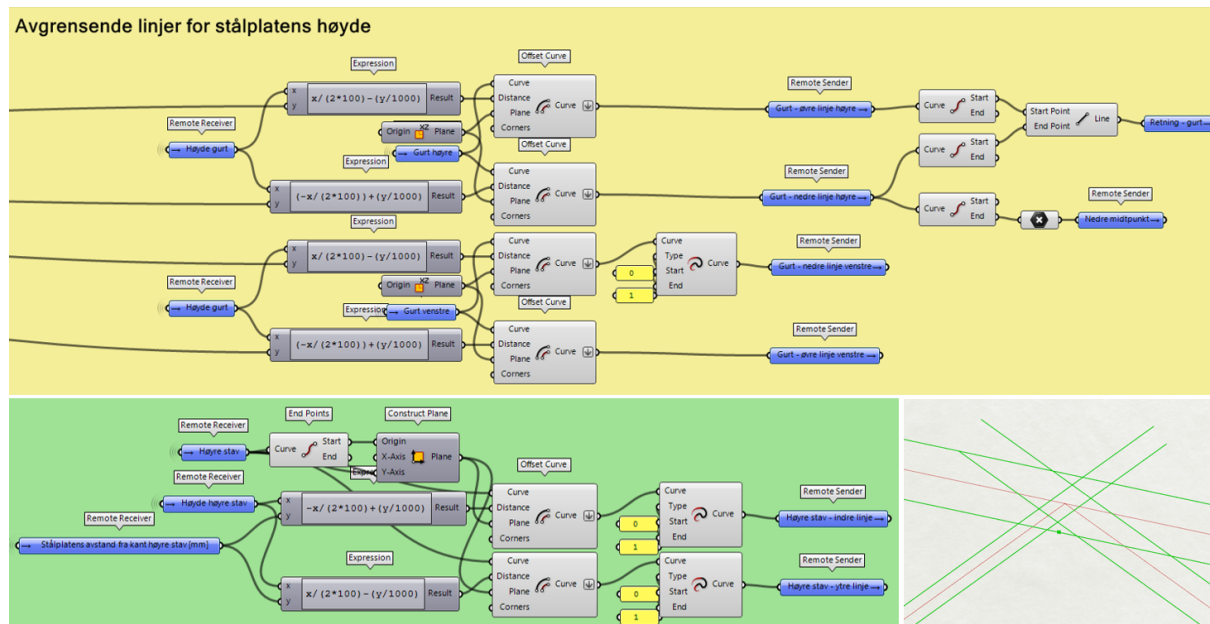
overgurtens systemlinje og etablerer en ny linje med avstanden: overgurtens høyde dividert med 2. Den nye linjen vil krysse diagonalenes ytterlinjer, og ved dette krysningspunktet settes endepunktet. (Diagonalenes ytterlinjer etableres i ramme 4)



Figur 6.95 Definerer av stavers endepunkt (ramme 3)

Videre etableres linjer som definerer stålplatens ytterkanter i ramme 4. I Figur 6.96 blir stålplatens øvre- og nedre kantlinjer etablert. Den gule ruten viser programmering for overgurt, mens den grønne viser programmering for høyre stav (venstre stav vil ha samme utførelse som høyre). Bildet nede til høyre er sluttresultatet både for overgurt og staverne. Også her benyttes «Offset Curve» for å etablere nye linjer med en gitt avstand fra de opprinnelige systemlinjene. «Expression»-komponenten benyttes for å bestemme avstanden. Denne avstanden beregnes ved å subtrahere høyden som hentes fra ramme 2 fra den totale høyden til elementet. Dette utføres for hvert element, både for over- og underkant. Om avstanden skal ha positivt eller negativt fortegn er avhengig av hva som er start- og endepunktet til systemlinjen. XZ-planet kan benyttes for overgurt, mens det for staverne må bygges et plan som følger systemlinjenes lengdeakse. For dette benyttes «Construct Plane»-komponenten som kan lage et plan basert på en linje og dens endepunkter.

I tillegg er noen av linjene forlenget med «Extend Curve»-komponenten. Disse er lagt til på et senere tidspunkt, hvor det har blitt oppdaget at linjene ikke strekker langt nok for å nå et krysningspunkt eller lignende. Til slutt er det hentet informasjon fra overgurt-linjenes endepunkter, til bruk for videre modellering. Det blir blant annet hentet ut en linje som ligger på tvers av overgurtens fiberretning, som benyttes senere for å etablere linjer i samme retning. Denne kalles «retning gurt». I tillegg settes det et målepunkt i overgurt ved stålplatens nedre midtpunkt.

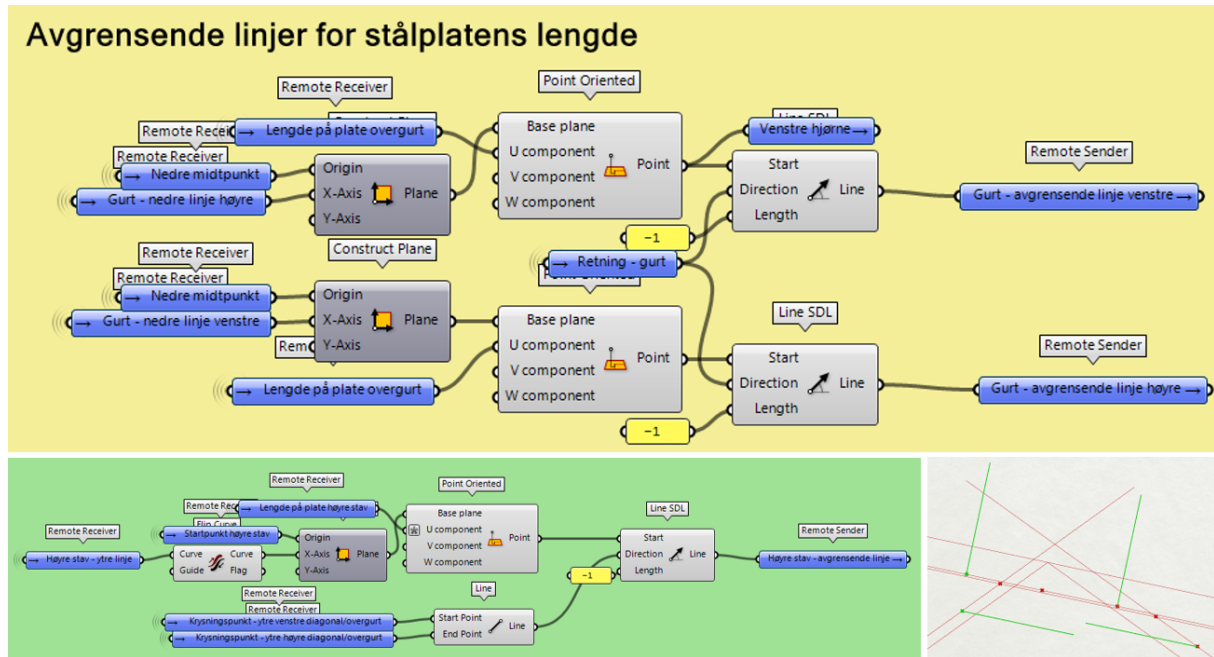


Figur 6.96 Linjer som definerer stålplattens ytterkant parallelt med elementets fiberretning (ramme 4)

I ramme 4 blir også stålplattens kantlinjer på tvers av fiberretning etablert, som vist i Figur 6.97. På samme måte som ovenfor viser den gule ruten programmering for overgurtene, mens den grønne viser programmering for høyre stav (venstre stav vil ha samme utførelse som høyre). I tillegg viser bildet nede til høyre sluttresultatet for både for overgurtene og stavene.

Ved bruk av komponenten «Point Oriented» kan det etableres et punkt med en gitt avstand fra origo i et plan. Og som tidligere nevnt kan «Construct Plane»-komponenten lage et plan basert på en linje og punkter. For overgurtene benyttes derfor målepunktet «nedre midtpunkt» som origo, og overgurtens nedre linjer som u-akse. Deretter benyttes lengdene som ble beregnet i ramme 2, for å bestemme avstanden til punktet i u-akse fra origo. Med dette utgangspunktet kan det etableres linjer med «Line SDL», som etablerer en linje fra punktet i en bestemt retning. Retningen bestemmes med «retning gurt» som hentes fra tidligere programmering.

For stavene benyttes endepunktene fra ramme 3 som origo, og stavenes ytre linjer som u-akse. I tillegg bestemmes retningen ved hjelp av punkter som hentes fra ramme 5.

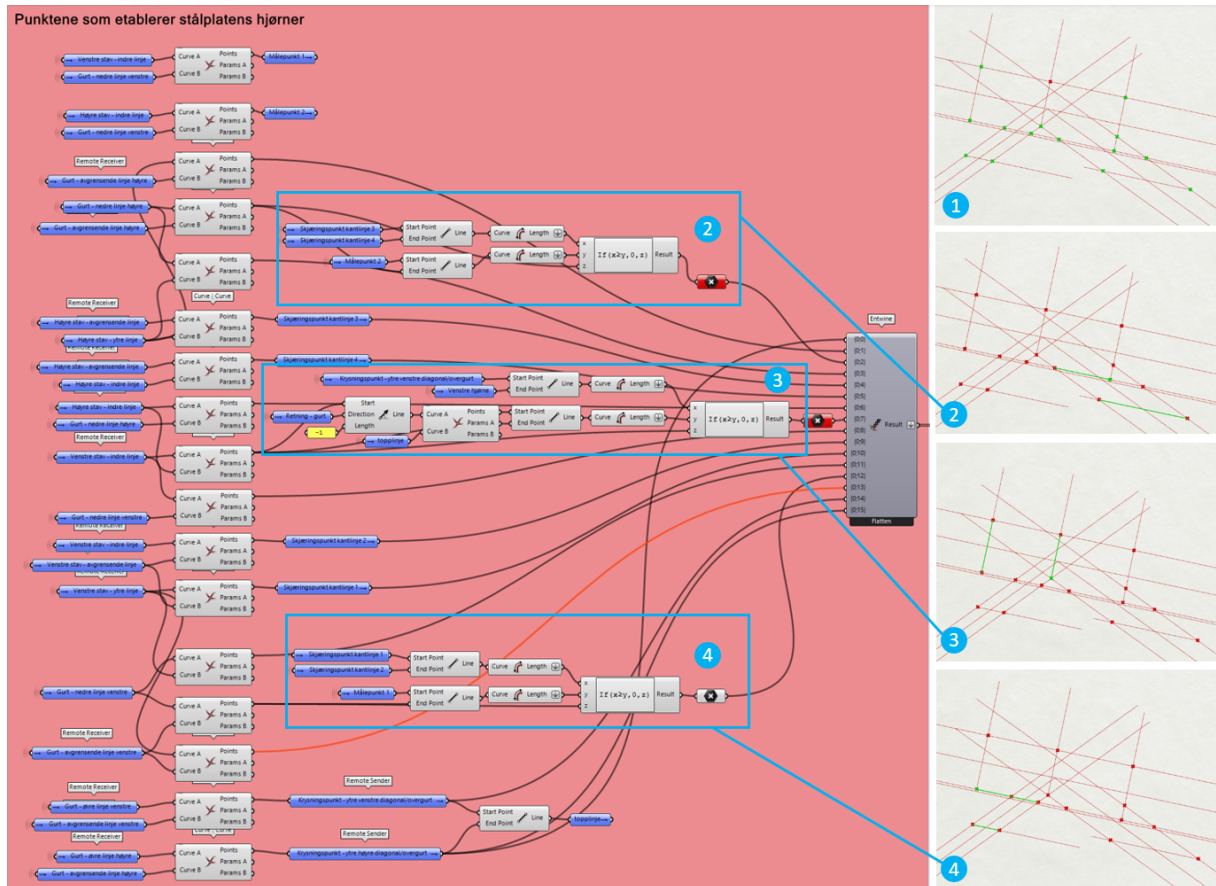


Figur 6.97 Linjer som definerer stålplaten ytterkant vinkelrett på elementets fiberretning (ramme 4)

I ramme 5 benyttes linjene fra alle elementer i ramme 4 til å danne en samling av stålplaten hjørner- og vendepunkter. Dette utføres ved bruk av «Curve | Curve»-komponenten som danner punkter i krysningspunktet mellom to linjer. Punktene er sortert i løpende rekkefølge langs stålplaten ytterkant. Alle punktene er uthevet med grønn farge i bilde nr.1 i Figur 6.98.

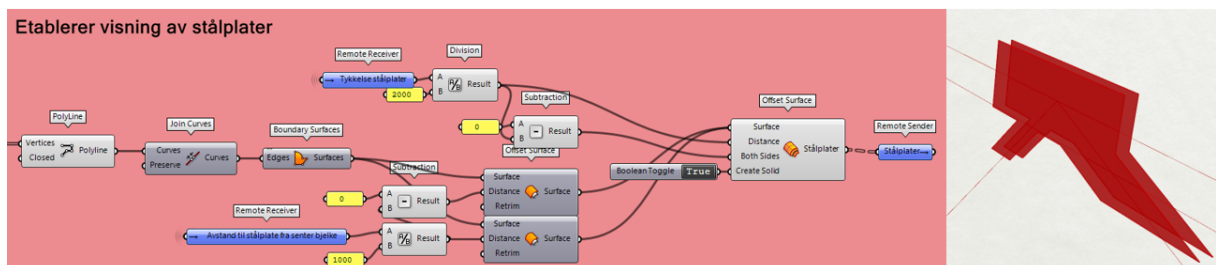
I tillegg er det programmert funksjoner for noen av punktene, fordi de bare skal eksistere i enkelte tilfeller. Øverste punkt med tilleggsprogrammering (tilknyttet bilde nr. 2), gjelder krysningspunktet mellom overgurtens høyre/nedre kantlinje og dens avgrensene linje på tvers av fiberretning. Dette punktet skal ikke være et hjørne eller vendepunkt i stålplaten dersom høyre stav treffer overgurten på høyre side (utsiden) av dette punktet. For å få til dette er det etablert en målingslinje mellom det gjeldende punktet og krysningspunktet mellom overgurt og høyre stavs indre linje. Dermed er det dannet en funksjon som sier at dersom denne linjen er kortere enn høyre stavs avgrensene linje på tvers av fiberretning, så skal ikke punktet fungere. Dersom linjen imidlertid er lengre enn (eller like lang som) høyre stavs avgrensene linje, så skal punktet fungere som et hjørne/vendepunkt i stålplaten. I tilfellet som vist i Figur 6.98, er målingslinjen kortere enn avgrensene linje, og derfor utheves punkt-komponenten med rød farge for å indikere at komponenten ikke inneholder et punkt (og vil dermed ikke fungere). Akkurat det samme er utført på motsatt side av overgurten, som vist i bilde nr. 4 i Figur 6.98. Her er målingslinjen lengre enn venstre stavs avgrensene linje, og punktet vil dermed fungere som et hjørne/vendepunkt.

Til slutt er det også utført samme type tilleggsprogrammering for krysningspunktet mellom stavenes indre kantlinjer (tilknyttet bilde nr. 3 i Figur 6.98). Dette punktet skal bare være et hjørne eller vendepunkt i stålplaten dersom avstanden mellom det gjeldende punktet og stålplaten toppunkt er lengre enn overgurtens avgrensene linje på tvers av fiberretning.



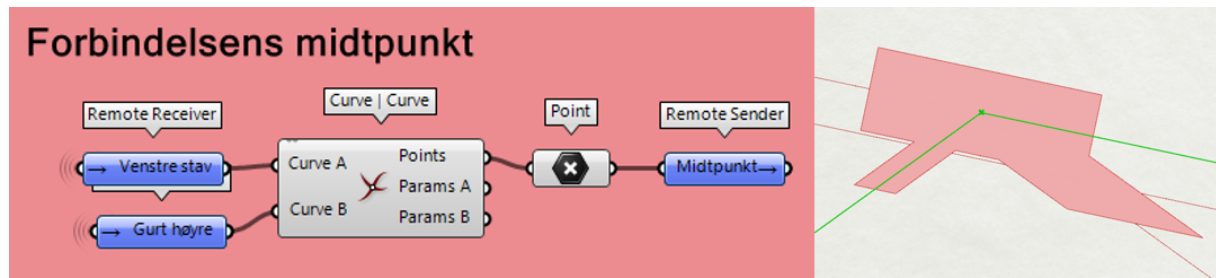
Figur 6.98 Samling av stålplatenes hjørner- og vendepunkter (ramme 5)

Ramme 6 blir vist i Figur 6.99. Her blir samlingen av stålplatenes hjørner- og vendepunkter benyttet for å etablere linjer mellom alle punktene ved bruk av komponenten «PolyLine». Deretter benyttes «Join Curves» for å etablere én sammenhengende linje, som legger grunnlag for å danne en flate med komponenten «Boundary surfaces». Flaten ligger nå i overgurtens systemlinje, og kopieres derfor ved hjelp av «Offset Surface» til hver sin side. Avstanden fra hver side av senterlinjen blir bestemt som input-verdi på utsiden av clusteren. De to nye flatene skaper dermed utgangspunktet for to stålplater. Ved hjelp av en annen «Offset Surface»-komponent har en mulighet til å gjøre flatene om til volumer. Den gir også mulighet til å definere tykkelsen til stålplaten. På denne måten vil flatene se ut som to stålplater.



Figur 6.99 Etablere visning av stålplater (ramme 6)

I ramme 7 etableres et punkt i krysningpunktet mellom stavene og overgurt. Komponentene «Curve|Curve» benyttes for dette, som vist i Figur 6.100. Dette midtpunktet benyttes som utgangspunkt for dybelgruppen.

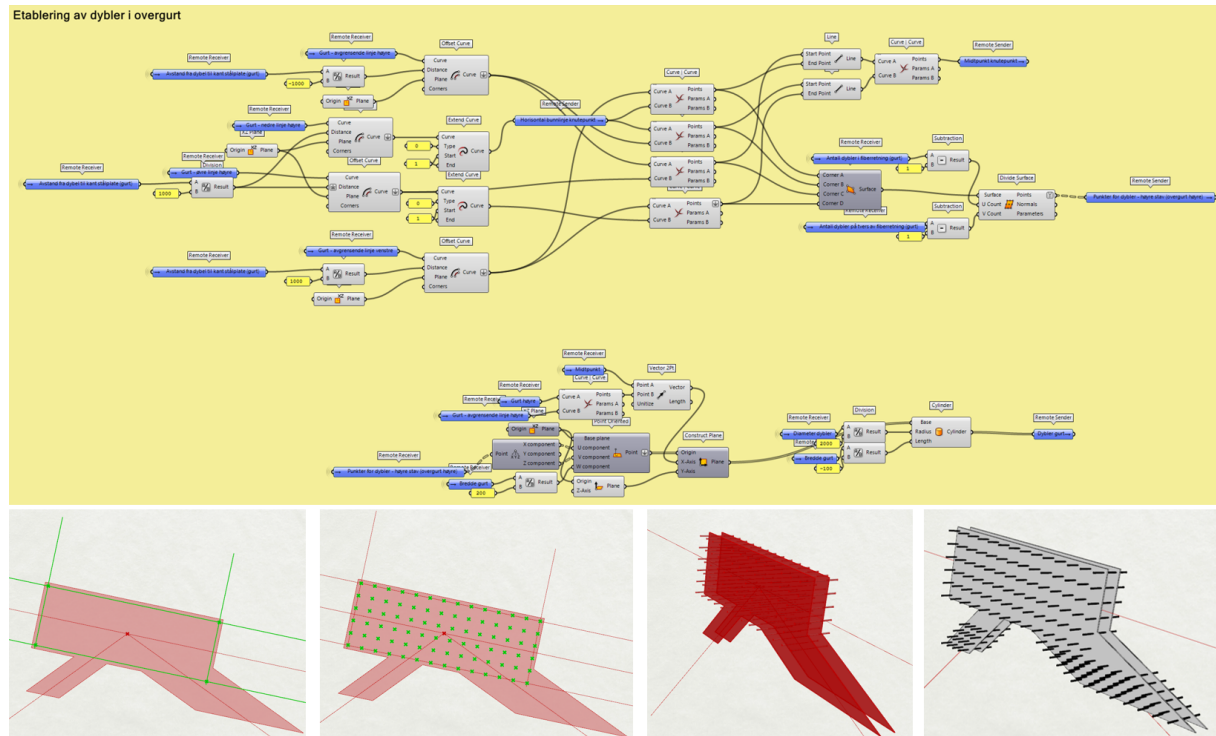


Figur 6.100 Definerings av forbindelsens midtpunkt (ramme 7)

I ramme 8 dannes dybelgruppene, hvor Figur 6.101 viser hvordan dybelgruppen i overgurt ble bygget. Med utgangspunkt i stålplatens ytre linjer, benyttes «Offset Curve» for å danne dybelgruppens avgrensende linjer (vist i første bilde fra venstre i Figur 6.101). De nye linjene vil forskyves i henhold til den definerte avstanden mellom dybel og kant stålplate. Dermed blir «Curve | Curve»-komponenten benyttet for å hente ut hjørnene til dybelgruppens avgrensende linjer. Disse hjørnene hentes ut for å danne en flate med komponenten «4Pt Surface». Med dette har en grunnlaget for å benytte komponenten «Divide Surface», som lager ett rutenett med punkter hvor dyblene skal plasseres. Her vil antall ruter i komponentens u- og v-retning bestemmes i henhold til definert antall dybler i fiberretning og på tvers av fiberretning (som settes på utsiden av clusteren). Den resulterende punktgruppen blir vist i andre bilde fra venstre i Figur 6.101.

I nederste del av den gule rammen i Figur 6.101, benyttes punktgruppen fra tidligere for å bygge én sylinder i hver av punktene. Komponentene «Cylinder» vil etablere sylindere ut fra origo i ett plan. Derfor må ett plan etableres for hvert enkelt punkt i punktgruppen. Planenes origoer må dermed ligge i punktgruppens x- og z-koordinater, samt forskyves i y-akse. Dersom de ikke forskyves i y-akse vil sylinderen starte i gurtens senterlinje. Derfor benyttes komponenten «Point Oriented» sammen med komponenten «Deconstruct», for å hente ut punktenes x- og z-koordinater. I tillegg legges det inn en y-koordinat med halle overgurt-bredde. På denne måten blir origoene for «construct plane»-komponenten definert, og i tillegg spesifiseres retningen på planets x- og y-akse.

I tillegg spesifiseres sylinderens radius og lengde. Radius beregnes basert på dyblenes diameter som bestemmes som input-verdi på utsiden av clusteren. I tillegg vil lengden være lik bjelkelementets bredde.

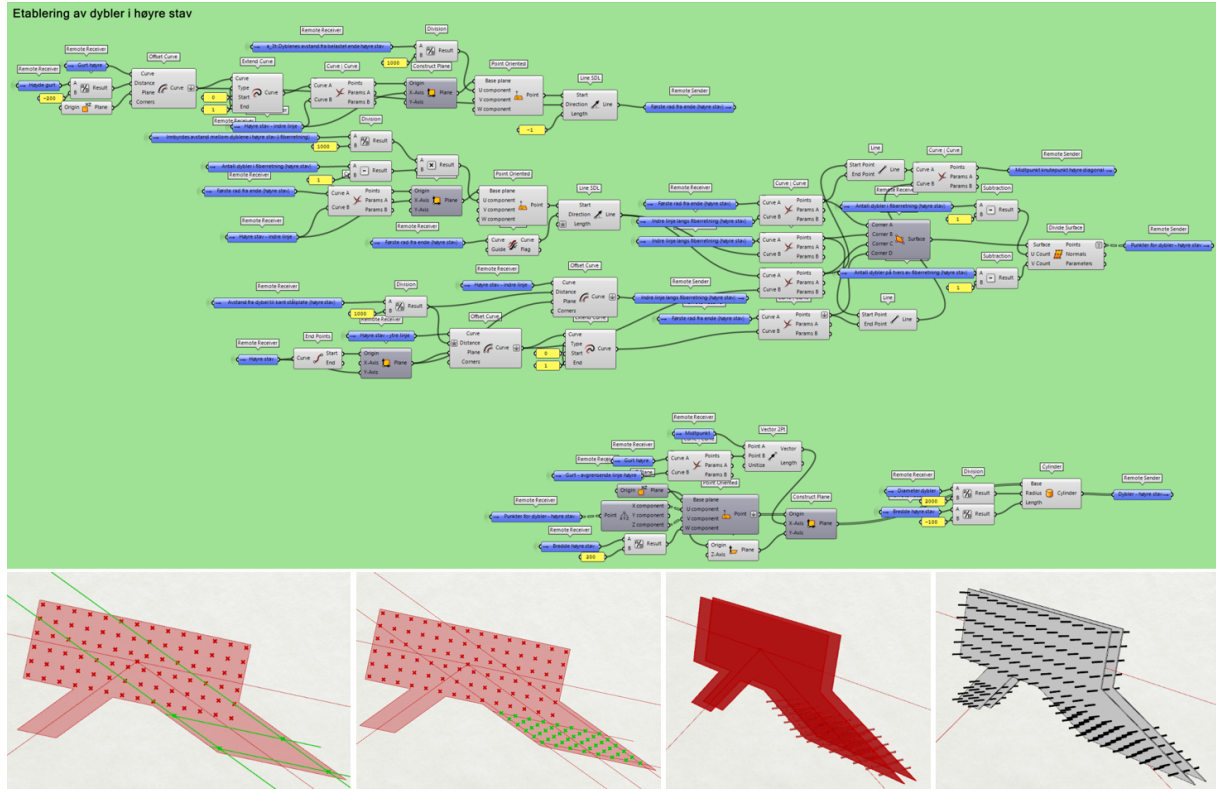


Figur 6.101 Etablering av overgurtens dybler i rutenett-struktur (ramme 8)

I Figur 6.102 vises det hvordan dybelgruppen i høyre stav er bygget (dybelgruppen i venstre stav har identisk oppbygging som høyre stav). Dybelgruppene i stavene bygges med samme fremgangsmåte som i overgurtten. Det etableres en ramme for dybelgruppens avgrensende linjer, deretter lages ett rutenett med punkter og til slutt blir punktene omgjort til sylindere. Forskjellen ligger i hvordan man bygger den avgrensende rammen for dybelgruppen.

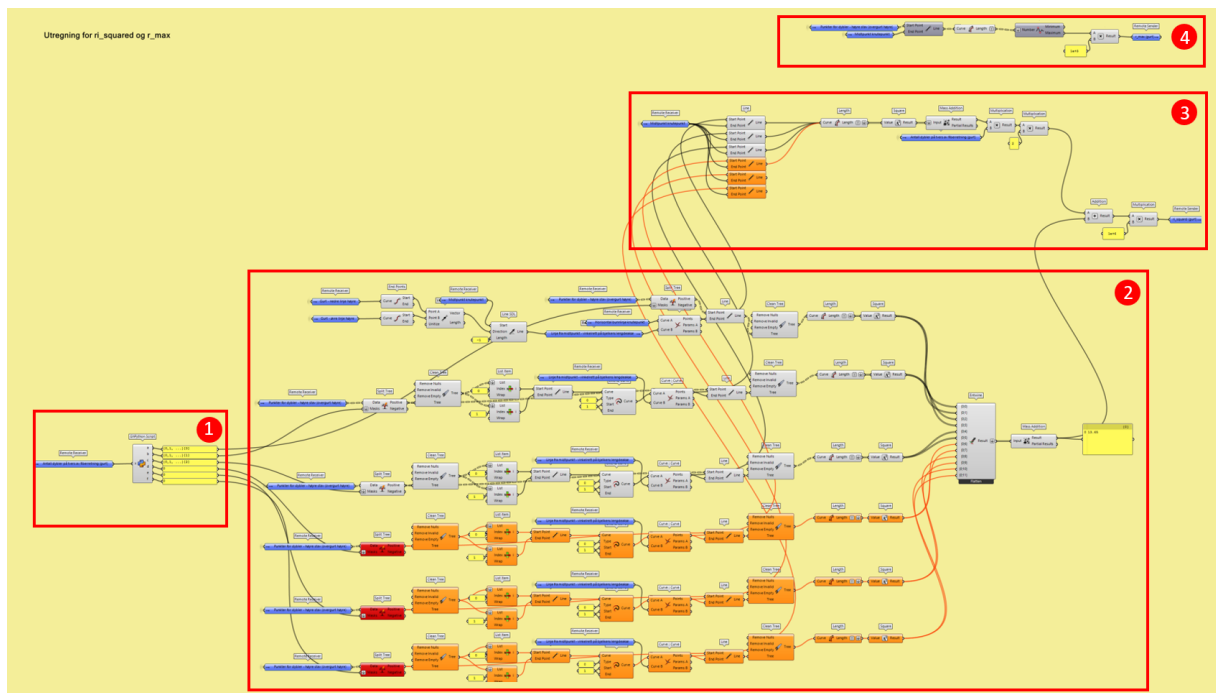
For de avgrensende linjene langs med stavens fiberretning kan «Offset Curve» benyttes på samme måte som i overgurtten. Likevel må det bygges et plan som følger stavens lengdeakse med «Construct Plane»-komponenten på samme måte som ved høydene i ramme 4. For linjene på tvers av fiberretning må imidlertid komponentene «Point Oriented» og «Line SDL» benyttes sammen, slik som ved breddene i ramme 4. For linjen som er nærmest overgurtens dybelgruppe, defineres startpunktet på samme måte i ramme 3. Linjen plasseres deretter med en avstand som defineres av bestemt endeavstand. Videre benyttes denne linjen videre til å plasseres linjen lengst borte fra overgurtens dybelgruppe. Den nye linjen forskyves dermed med en avstand som bestemmes av innbyrdes avstand i fiberretning multiplisert med antall mellomrom mellom dyblene.

Når rammen for dybelgruppens avgrensende linjer er etablert, er metoden identisk som metoden for overgurtens dybelgruppe.



Figur 6.102 Etablering av stavenes dybler i rutenett-struktur (ramme 8)

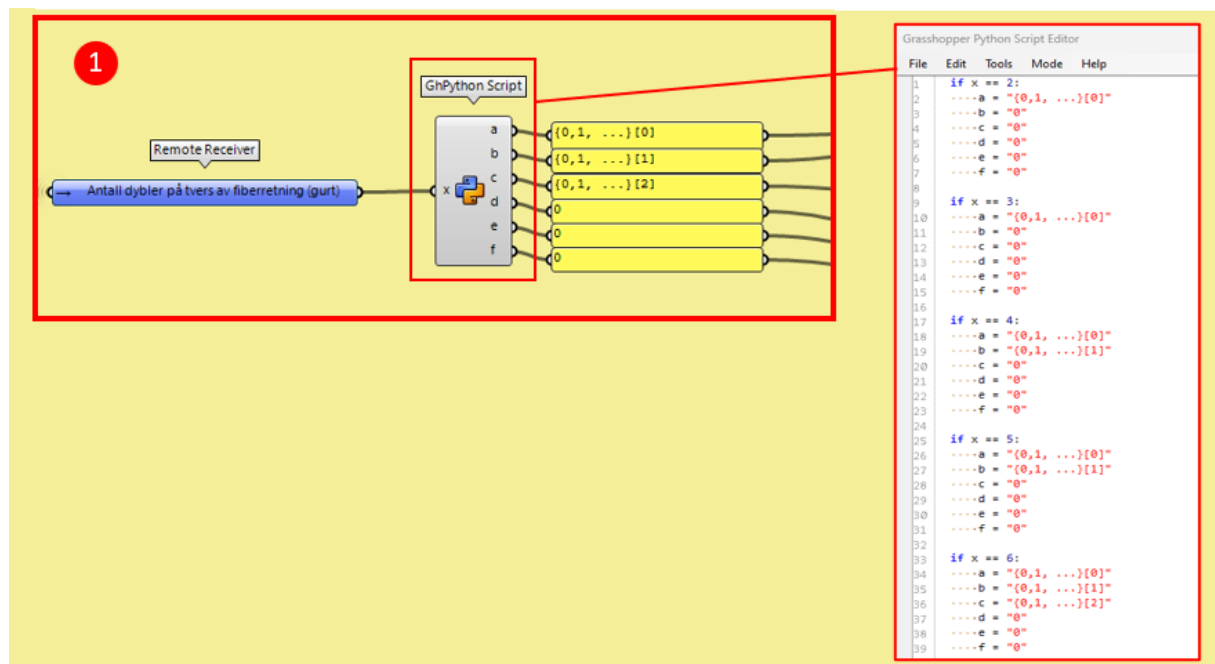
I ramme 9 måles avstander for beregning av $r_{i_squared}$ og r_{max} , som benyttes videre for beregning av resultatant på mest påkjente forbindere. Dette utføres på samme måte for overgurt og stavene, men utklippet i Figur 6.103 er hentet fra beregning i overgurt. Figur 6.103 er delt inn i fire grupper, og videre vil vi ta for oss del for del av denne oppbyggingen.



Figur 6.103 Måling av avstander for beregner $r_{i_squared}$ og r_{max} (ramme 9)

For å beregne ri_squad , er det ønskelig å måle avstanden fra dybelgruppens midtpunkt frem til hver av rekkene som går både vinkelrett på- og parallelt med fiberretning. Derfor er det bygget et system som henter ut punkttrekkene parallelt med fiberretningen. Det hentes imidlertid bare ut punkter fra ene siden av dybelgruppens midtpunkt, da avstandene vil være like i motsatt retning og kan derfor multipliseres med to. Punktgruppen som dannes for hver av dybelgruppene i ramme 8 videreføres i en liste, hvor hver gren representerer en kolonne med dybler (på tvers av fiberretningen). Nulte indeks i hver av grenene representerer gruppens nederste rekke med punkter, deretter representerer første indeks nest-nederste rekke med punkter og så videre.

I første gruppe i ramme 9 er det bygget et script som forteller hvilke punkter som skal hentes ut ved ulike antall dybler på tvers av fiberretningen (presentert i Figur 6.104). Ved to og tre dybler på tvers av fiberretningen, skal bare nederste punkttrekkene parallelt med fiberretningen hentes ut. Derimot, skal det ved fire og fem dybler på tvers av fiberretningen, hentes ut både nederste- og nest-nederste punkttrekke parallelt med fiberretningen. Slik fortsetter scriptet videre, og fungerer med opptil 12 antall dybler på tvers av fiberretningen.

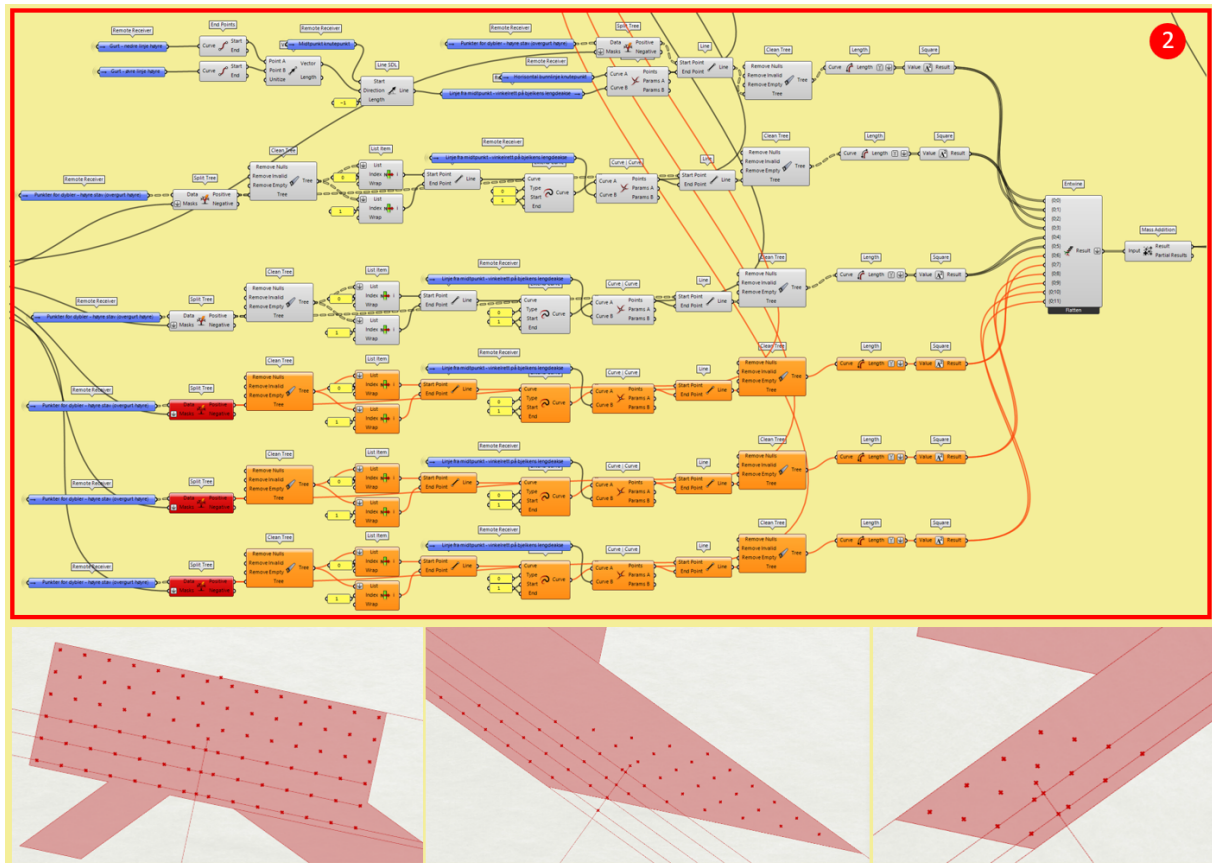


Figur 6.104 Sortering av punktgruppe (ramme 9)

De riktige punktene hentes ut ved hjelp av komponenten «split tree» i ramme 9s andre gruppe. I tilfellet som presenteres i Figur 6.105, er det seks dybler på tvers av fiberretning i overgurt, og dermed er det de tre nederste rekkene med punkter som hentes ut. Som vist i Figur 6.105, vil bare de tre øverste operasjonsrekkene fungere, mens de resterende operasjonene bare vil fungere dersom det tilføyes flere antall dybler på tvers av fiberretningen. I hver av punkttrekkene som hentes inn, blir første og andre punkt i rekken sortert ut med komponenten «split list». Videre blir det etablert en linje mellom punktene, som til slutt blir forlenget til en lengre linje med komponenten «Extend curve».

Samtidig hentes det inn en linje med komponenten «Line SDL», som starter ved dybelgruppens midtpunkt og har retning vinkelrett på elementets lengdeakse. Denne operasjonen synes i øvre,

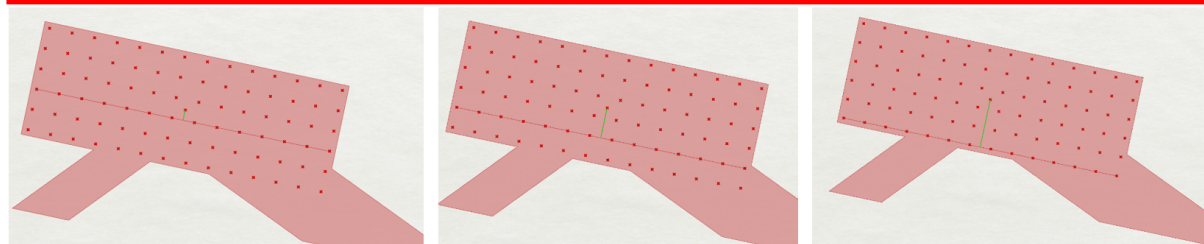
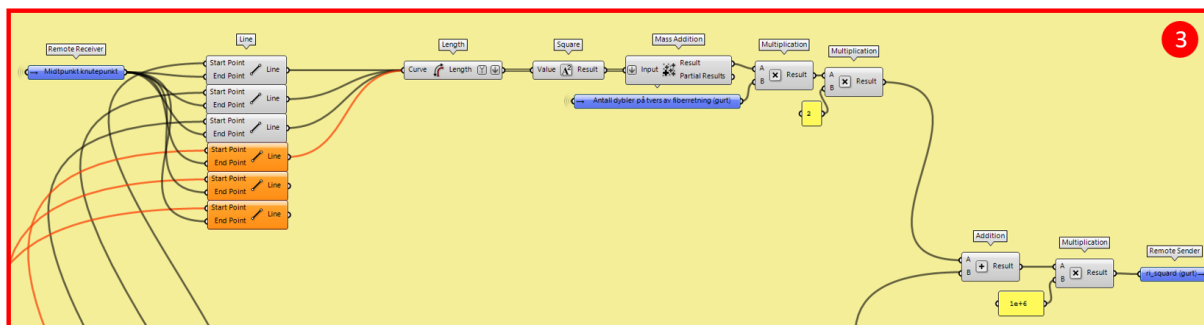
venstre hjørne av Figur 6.105. Linjen benyttes for å danne krysningspunkter med de forlengede linjene fra forrige avsnitt. Deretter benyttes «line»-komponenten til å tegne linjer mellom krysningspunktet i raden og hver av radens punkter. Alle disse linjene blir målt med «length»-komponenten og opphøyes i andre. Til slutt samles verdiene i en lang liste, og summeres med komponenten «Mass addition».



Figur 6.105 Måler avstander mellom dybelgruppens senterlinje og punkttrekkene på tvers av fiberretning (ramme 9)

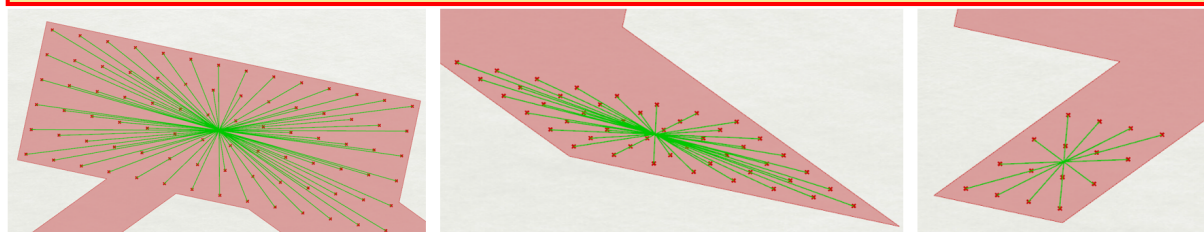
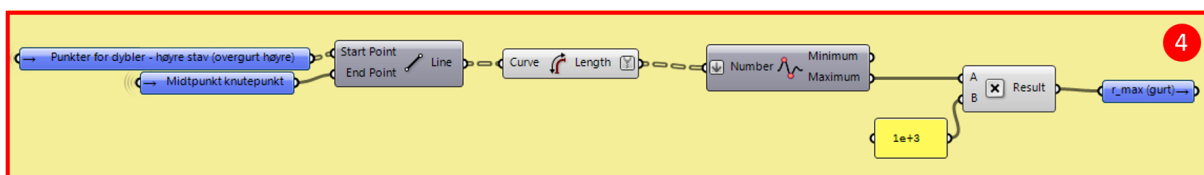
I tredje gruppe i ramme 9 måles linjene mellom dybelgruppens midtpunkt og punkttrekkene i fiberretning. Som presentert i Figur 6.106, hentes hver av krysningspunktene mellom dybelgruppens senterlinje og de forlengede linjene i fiberretning, som ble etablert i gruppe to. Deretter etableres en linje mellom dybelgruppens midtpunkt og hver av disse krysningspunktene. Til slutt summeres lengdene på alle linjene, og multipliseres med antall dybler på tvers av fiberretningen. I tillegg multipliseres verdien med to, for å også omfatte motstående side av dybelgruppen.

I siste operasjon summeres lengdene fra dybelgruppens midtpunkt frem til hver av rekkene som går både vinkelrett på- og parallelt med fiberretning. Dette gir verdien «ri_squared».



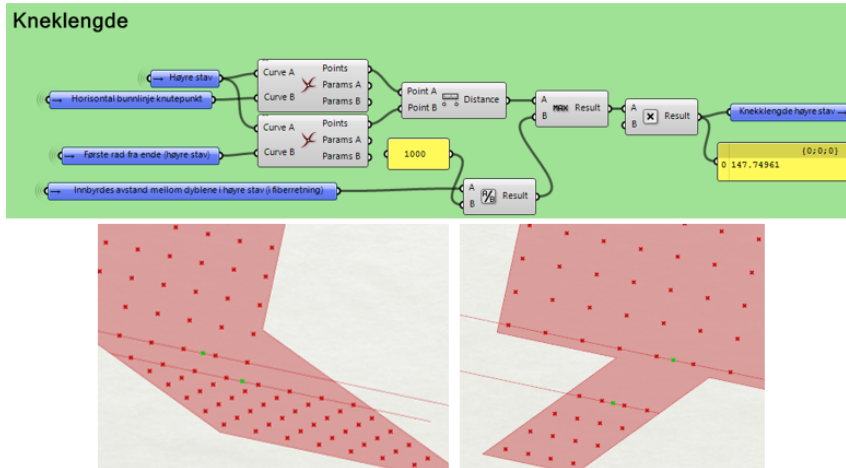
Figur 6.106 Måler avstander mellom dybelgruppens midtpunkt og punktrekkene i fiberretning (ramme 9)

For å måle «r_max», måles avstanden fra dybelgruppens midtpunkt og frem til hvert enkelt punkt i dybelgruppen. Den lengste linjen finnes ved hjelp av «Min/Max» komponenten, som vist i Figur 6.107.



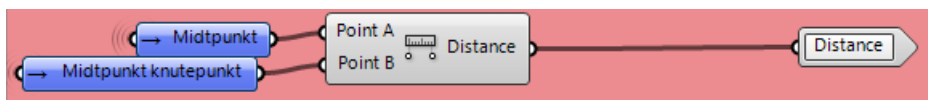
Figur 6.107 Beregning av «r_max» (ramme 9)

Til slutt måles stålpлатens lengste knekk lengde for å kunne kontrollere stålplassen for knekking. Programmeringen for dette er presentert i Figur 6.108. Dette utføres ved å måle avstanden mellom overgurtens nederste dybel-rekke og stavens øverste dybel-rekke. Det kan oppstå tilfeller hvor innbyrdes avstand mellom dyblene i fiberretning er lengre enn den målte knekk lengden. Derfor benyttes også «maximum»-komponenten for å hente ut den lengste lengden av de to.



Figur 6.108 Måling av knekklangde (ramme 10)

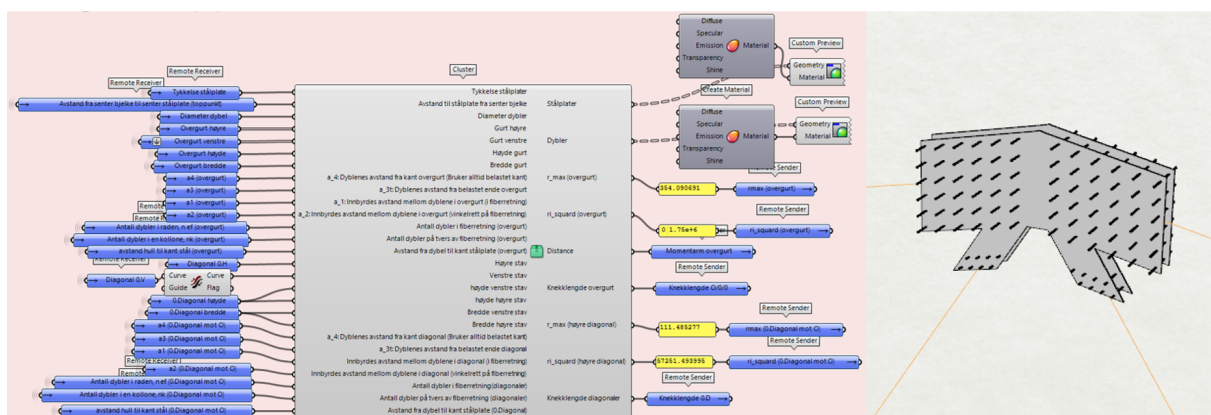
Output-verdiene listes opp i ramme 11. Nødvendige outputer omfatter visning av stålplatene (volumene) og visning av dyblene (sylinderne), samt verdier som er nødvendige for eurokodekontroller. En av verdiene som er nødvendige for eurokodekontroller er momentarm, som er behøves ved eksentrisitet i forbindelsen. Denne måles mellom midtpunktet hvor alle elementene møtes, og midtpunktet til dybelgruppen, som vist i Figur 6.109. I kontinuerlige gurter er denne avstanden lik null.



Figur 6.109 Måling av momentarm (ramme 11)

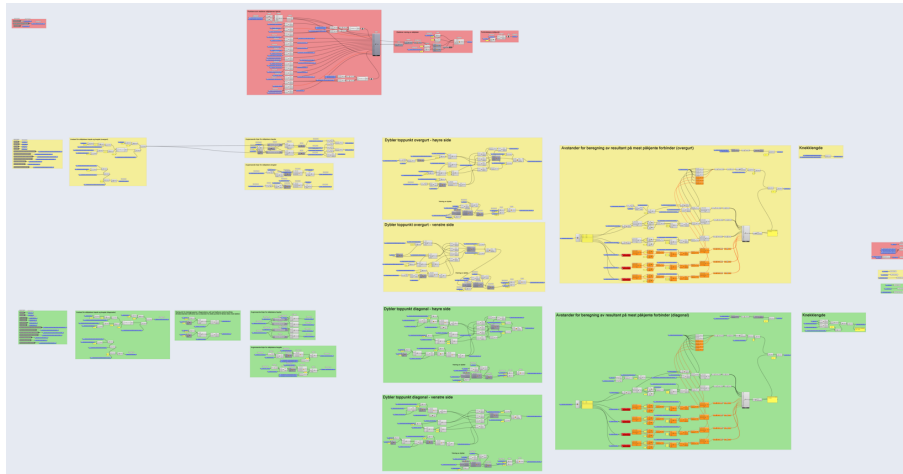
Toppunktforbindelse (Grønn cluster)

I motsetning til den oransje clusteren, har forbindelsen i toppunktet en endeavstand fra krysningspunktet hvor overgurtene og diagonalene møtes, og videre frem til dybelgruppens første rekke. Dette vil fungere på samme måte ved en skjøttet forbindelse. Denne clusteren er markert med grønn farge på ikonet, som vist i Figur 6.110.



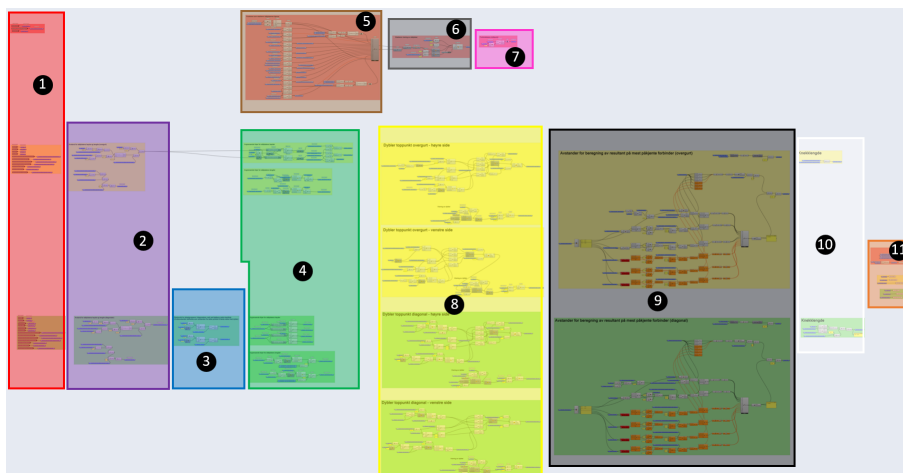
Figur 6.110 Cluster for skjøttet forbindelse (grønn)

Figur 6.111 viser oppbygging av den grønne clusteren. Her tilhører de gule gruppene dybelgruppen i overgurt, mens de grønne clusterene tilhører stavene. I tillegg viser de røde gruppene programmering som tilhører både stavene og overgurt.



Figur 6.111 Toppunktforbindelse (Grønn cluster)

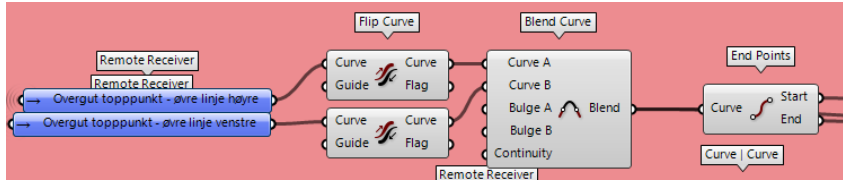
Figur 6.112 viser rekkefølgen på operasjonene som utføres for hver av elementene i den grønne clusteren. Den nummererte rammeinndelingen er identisk med inndelingen for den oransje clusteren, og innholdet i hver av rammene kan beskrives på samme måte som i Tabell 6.5. Det er imidlertid noen forskjeller innenfor rammene, som gjør at denne clusteren er tilpasset toppunktet.

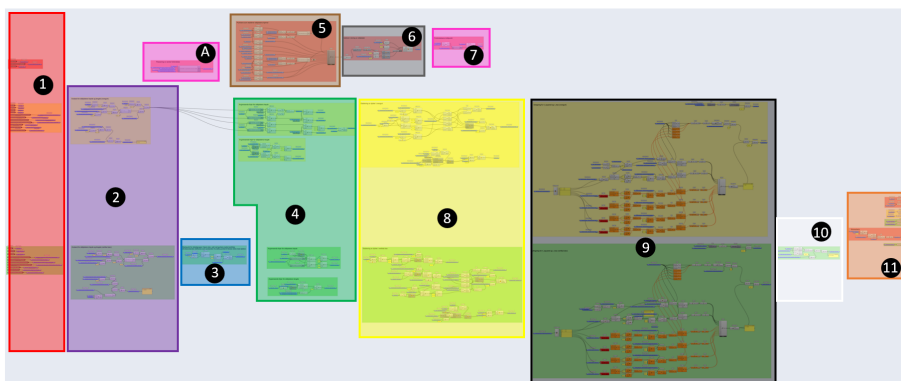


Figur 6.112 Nummerert fremgangsmåte for toppunktforbindelse (Grønn cluster)

Programmeringen som skiller modelleringen av toppunktet (grønn cluster) fra modelleringen av forbindelsen i kontinuerlig overgurt (oransje cluster), er presentert i Tabell 6.6. Fargene i tabellen viser til fargene i figuren.

Tabell 6.6 Programmering som er særegen for modellering av toppunkt (grønn cluster)

RAMME	Programmering som er særegen for modellering av toppunkt (grønn cluster)
1	Høyre- og venstre diagonal i toppunktforbindelsen er identiske, og har derfor like tverrsnittsegenskaper, samt like avstander i dybelgruppene. Derfor kommer det ikke inn to sett med inputverdier for hver av staven, men heller en samlet gruppe som inneholder all nødvendig informasjon.
2	I motsetning til forbindelsen i kontinuerlig overgurt, har forbindelsen i toppunktet en endeavstand fra krysningspunktet hvor overgurtene og diagonalene møtes og videre frem til dybelgruppens første rekke. For beregning av stålplattens lengde i overgurt, må derfor valgt endeavstand adderes til de resterende parameterne.
3	Ingen særegen programmering.
4	For den oransje clusteren, blir det hentet ut én linje som kalles «retning gurt». Denne ligger på tvers av overgurtens fiberretning, og benyttes ved senere modellering for å etablere linjer i samme retning. For modellering av toppunktet må det etableres en «retning gurt» for både høyre- og venstre overgurt.
5	<p>Overgurtens øvre stålplate-linjer vil ikke krysse hverandre, og det kan derfor ikke dannes et punkt her. Derfor benyttes komponenten «blend curve» for å skape en penere, avrundet overgang av stålplattens toppunkt. Programmeringen er presentert i Figur 6.113.</p>  <p style="text-align: center;"><i>Figur 6.113 Avrundet topp på stålplate</i></p> <p>Det trengs ingen tilleggsprogrammering for krysningspunktet mellom stavenes indre kantlinjer (som i oransje overgurt), da dette punktet alltid kan være et vendepunkt i stålplaten.</p>
6	Ingen særegen programmering.
7	Ingen særegen programmering.
8	I motsetning til forbindelsen i kontinuerlig overgurt, hvor det kun etableres en dybelgruppe i krysningspunktet mellom alle elementene, må det bygges dybelgruppe for overgurtens høyre- og venstre side i toppunktforbindelsen. I tillegg kan ikke «Offset curve» benyttes for første linje på tvers av fiberretning. Her må komponentene «Point Oriented» og «Line SDL» benyttes sammen, hvor linjen plasseres ved gitt endeavstand.
9	Ingen særegen programmering.
10	Ingen særegen programmering.
11	Ingen særegen modellering. Det vil opptre en momentarm som skaper eksentrisitet i forbindelsen i dette tilfellet.

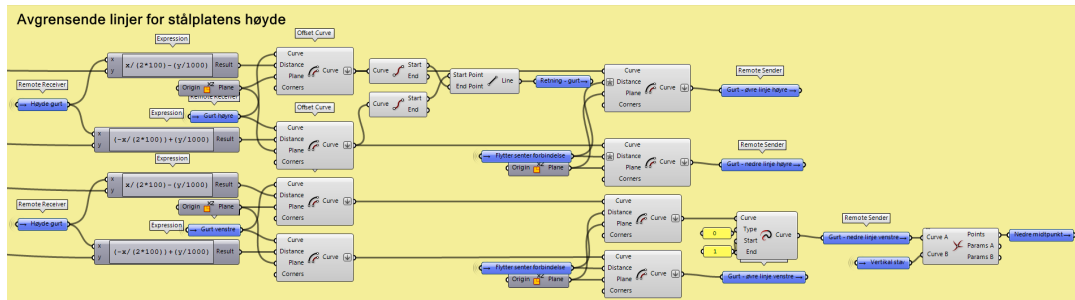


Figur 6.116 Nummerert fremgangsmåte for forbindelse i kontinuerlig overgurt tilknyttet vertikal stav (Lilla cluster)

Programmeringen som skiller modelleringen av forbindelse i kontinuerlig overgurt tilknyttet vertikal stav (lilla cluster) fra modelleringen av forbindelsen i kontinuerlig overgurt (oransje cluster), er presentert i Tabell 6.7. Fargene i tabellen viser til fargene i figuren.

Tabell 6.7 Programmering som er særegen for modellering av forbindelse i kontinuerlig overgurt tilknyttet vertikal stav

RAMME	Programmering som er særegen for modellering av forbindelse i kontinuerlig overgurt tilknyttet vertikal stav (grønn cluster)
1	I motsetning til oransje cluster, vil denne clusteren bare hente input-verdier for én stav.
2	Ingen særegen programmering.
3	Ingen særegen programmering.
A	<p>Ramme «A» definerer avstanden som dybelgruppen i overgurten skal flyttes nedover i tverrsnittet. Ved formelens bruk av formelen som vist i Figur 6.117, vil dybelgruppen plasseres slik at den overholder oppgitt kantavstand til overgurten nedre kant. Dermed vil denne clusteren kunne styres basert på nedre kantavstand, i motsetning til den oransje, som bare vil gi en tydelig visuell tilbakemelding, dersom kantavstanden overskrides.</p> <div data-bbox="373 1377 1374 1574" data-label="Diagram"> <p>The diagram shows a calculation process for determining the distance the dybel group should be moved. It includes inputs: 'Høyde gurt' (x), 'Innbyrdes avstand mellom dyblene i gurt (på tvers av fiberretning)' (y), 'Antall dybler på tvers av fiberretning (gurt)' (z), 'a_A: Dyblenes avstand fra kant gurt (Bruker alltid belastet kant)' (u), and 'Avstand fra dybel til kant ståloplate (vertikal stav)' (v). The formula used is $x / (2 * 100) - ((y * (z / 2)) + (u - v)) / 1000$. The result is used to 'Flytter senter forbindelse'.</p> </div>
4	<p>Den vertikale staven har identisk oppbygging som stavene i den oransje clusteren, og det samme gjelder ståloplaten lengde i overgurten. Men når det gjelder ståloplaten høyde i overgurten, må dybelgruppen forflyttes nedover i tverrsnittet i henhold til verdien fra resultatet av ramme A. Dette blir presentert i Figur 6.118.</p> <p>I tillegg til de opprinnelede «Offset Curve»-komponentene, blir alle linjene forflyttet en ekstra omgang med nye «Offset Curve»-komponenter. Avstanden de forflyttes, hentes fra ramme A.</p>

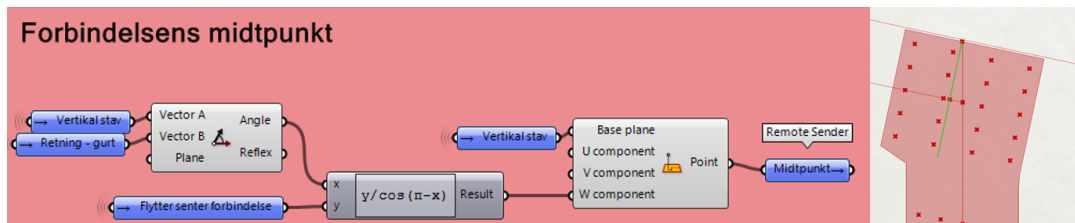


Figur 6.118 Forflytning av dybelgruppen i overgurten nedover i tverrsnittet

5 Det blir en del færre hjørner og vendepunkter, siden overgurten bare er tilknyttet en stav. Det vil blant annet ikke eksistere et krysningspunkt mellom stavenes indre kantlinjer (som i oransje overgurt). Derfor blir det også mindre tilleggsprogrammering for punkter som i noen tilfeller ikke skal eksistere.

6 Ingen særegen programmering.

7 Gruppe 7 inneholder i dette tilfellet dybelgruppens midtpunkt, og ikke krysningspunktet mellom elementene. Med komponenten «Point oriented», flyttes midtpunktet nedover fra den vertikale stavens endepunkt (i krysningspunktet mellom elementene). Avstanden den blir forflyttet, avhenger av resultatet fra ramme A. Men ettersom avstanden fra ramme A, måles langs den grønne linjen i Figur 6.119, brukes trigonometriske sammenhenger for å finne riktig avstand langs den vertikale linjen.



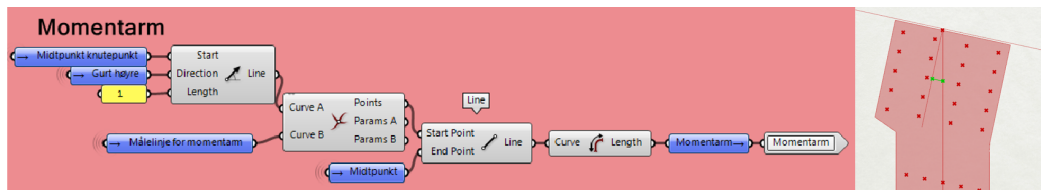
Figur 6.119 Dybelgruppens midtpunkt (lilla cluster)

8 Ingen særegen programmering.

9 Ingen særegen programmering.

10 Ingen særegen programmering.

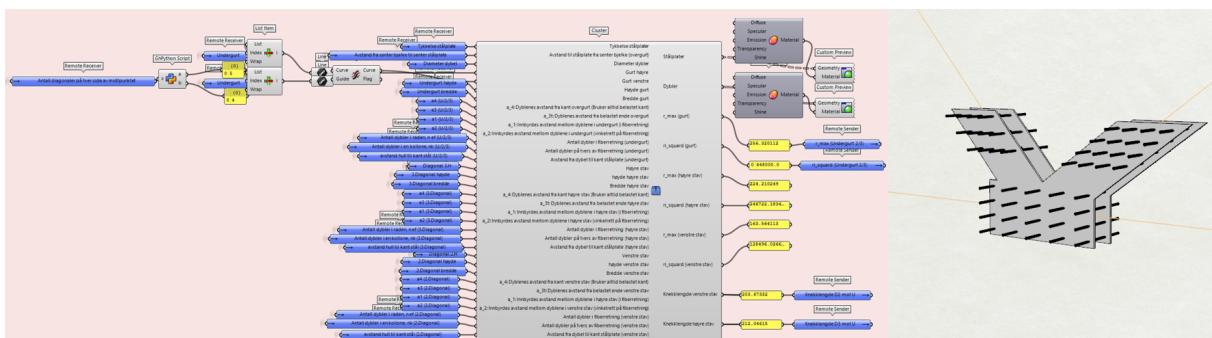
11 Det vil opptre en momentarm som skaper eksentrisitet i forbindelsen i dette tilfellet, fordi gruppen er rotert slik at dyblene ligger vinkelrett på overgurtens fiberretning. Linjen som starter ved forbindelsens midtpunkt og har retning vinkelrett på elementets lengdeakse, hentes fra gruppe 8. I tillegg lages en linje som starter i dybelgruppens midtpunkt og trekkes parallelt med overgurtens systemlinje. Det etableres et krysningspunkt mellom disse linjene med komponenten «Curve|Curve». Avstanden fra dybelgruppens midtpunkt til målepunktet, er dybelgruppens momentarm. Dette er vist i Figur 6.120.



Figur 6.120 Momentarm i vertikal stav.

Forbindelser i kontinuerlig undergurt (Blå cluster)

Forbindelsene i kontinuerlig undergurt, har dybelgruppens senter plassert midt i krysningpunktet hvor undergurten og diagonalene møtes. Denne clusteren er markert med blå farge på ikonet, som vist i Figur 6.121.



Figur 6.121 Cluster for forbindelse i kontinuerlig undergurt (blå)

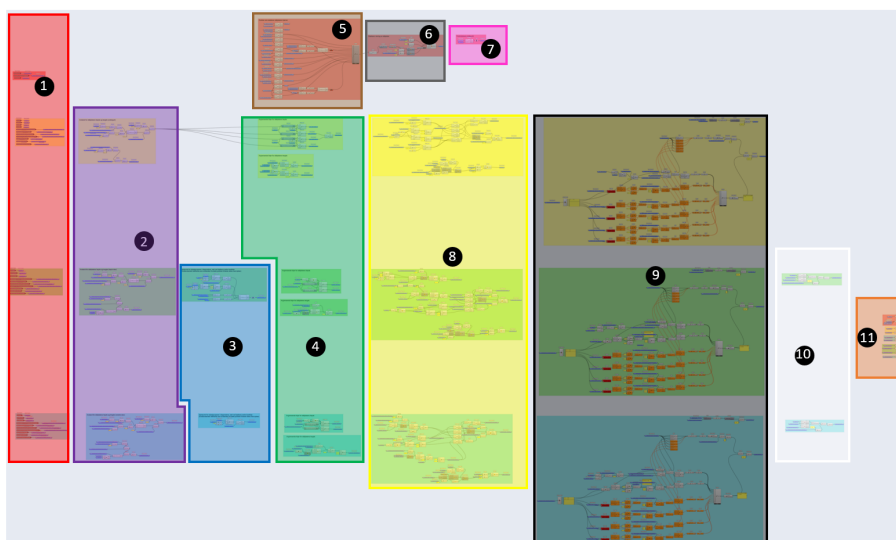
Innsiden av clusteren er vist i Figur 6.122. De gule gruppene tilhører dybelgruppen i undergurt, mens de grønne og blå clusterene tilhører henholdsvis høyre og venstre stav. I tillegg viser de røde gruppene programmering som tilhører både stavene og undergurt.



Figur 6.122 Forbindelser i kontinuerlig undergurt (Blå cluster)

Figur 6.123 viser rekkefølgen på operasjonene som utføres for hver av elementene i den blå clusteren. Den nummererte rammeinndelingen er identisk med inndelingen for den oransje

clusteren, og innholdet i hver av rammene kan beskrives på samme måte som i Tabell 6.5. Det er imidlertid noen forskjeller innenfor rammene, som gjør at denne clusteren er tilpasset denne forbindelsen.

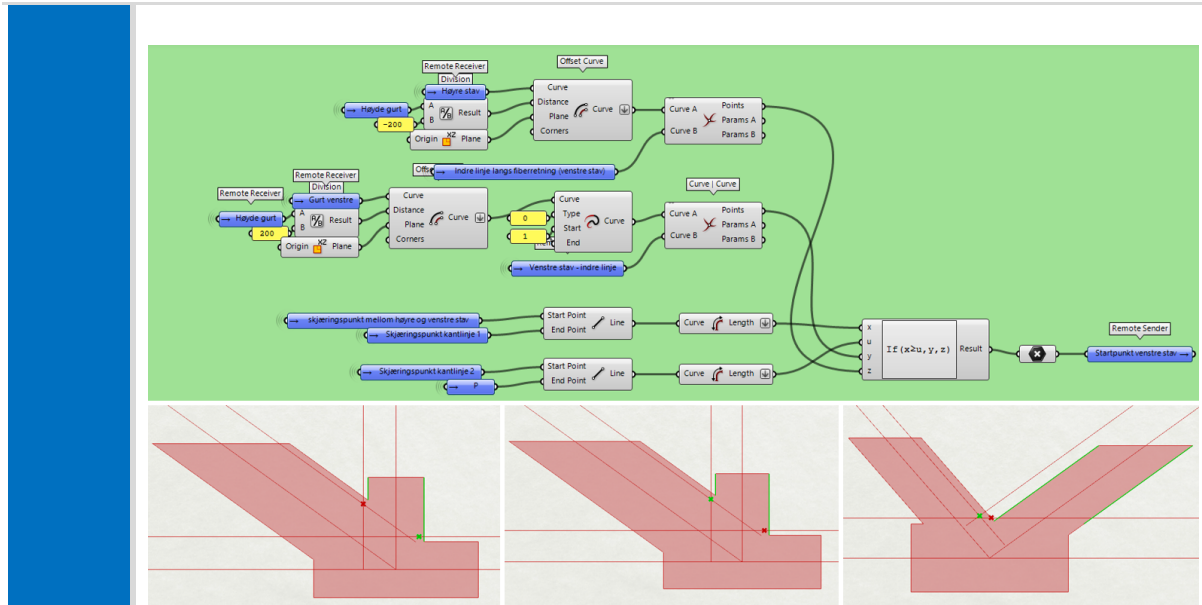


Figur 6.123 Nummerert fremgangsmåte for forbindelser i kontinuerlig undergurt (Blå cluster)

Programmeringen som skiller modelleringen av forbindelsene i kontinuerlig undergurt (blå cluster) fra modelleringen av forbindelsen i kontinuerlig overgurt (oransje cluster), er presentert i Tabell 6.8. Fargene i tabellen viser til fargene i figuren.

Tabell 6.8 Programmering som er særegen for modellering av forbindelsene i kontinuerlig undergurt

RAMME	Programmering som er særegen for modellering av forbindelsene i kontinuerlig undergurt (blå cluster)
1	Ingen særegen programmering.
2	Ingen særegen programmering.
3	<p>For forbindelsene i overgurt (oransje cluster), kuttet stavene alltid ved overgurtens nedre kantlinje, og dermed må dybelgruppens endeavstand måles fra dette punktet. I undergurt derimot, er det flere tilfeller hvor stavene må kuttet ved kantlinjen til den andre staven. Særlig i forbindelsen mot vertikal stav er dette tilfelle, som vist på bildene til venstre i Figur 6.124. Men det er også tilfeller hvor litt av staven må kappes mot den andre, som vist på bildet til høyre i Figur 6.124.</p> <p>Derfor er det etablert et system, hvor stavens startpunkt (hvor endeavstanden måles fra), justeres basert på forbindelsens utforming. Krysningspunktet hvor første rekke med dybler langs med fiberretningen i venstre stav krysser høyre stavs kantlinje, benyttes som startpunkt dersom høyre stavs indre linje er kortere enn den ytre linjen. På den andre siden, benyttes punktet ved overgurtens nedre kantlinje dersom høyre stavs indre linje er lengre enn den ytre linjen.</p> <p>På denne måten vil startpunktet alltid ligge der hvor staven kappes.</p>

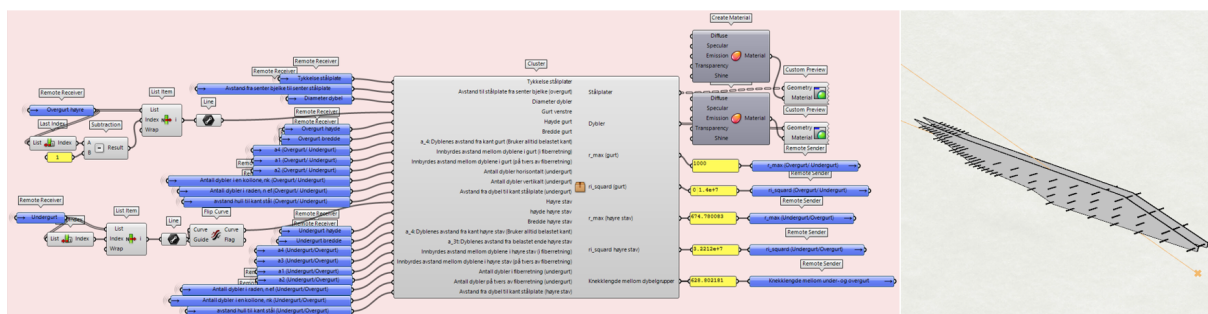


Figur 6.124 System for staves startpunkt (hvor endeavstanden måles fra) i blå cluster

4	Ingen særegen programmering.
5	Rekkefølgen på punktene vil endres. I tillegg trengs det ingen tilleggsprogrammering for kryssningspunktet mellom stavenes indre kantlinjer (som i oransje overgurt), da dette punktet alltid kan være et vendepunkt i stålplaten.
6	Ingen særegen programmering.
7	Ingen særegen programmering.
8	Ingen særegen programmering.
9	Ingen særegen programmering.
10	Ingen særegen programmering.
11	Det vil aldri være momentarm i kontinuerlig undergurt.

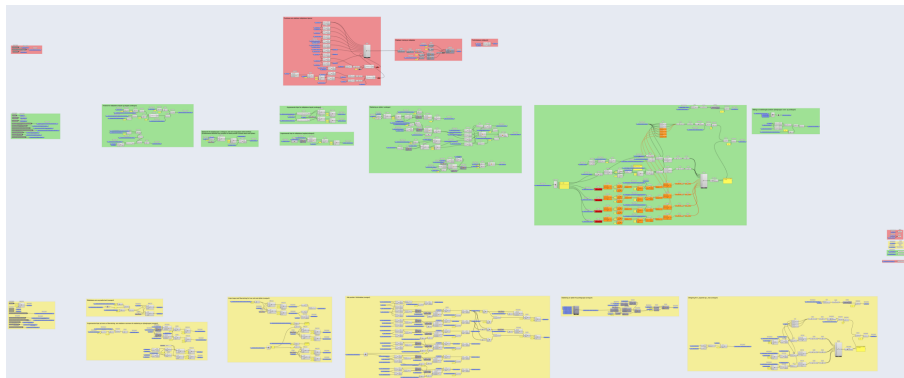
Forbindelse mellom over- og undergurt (Brun cluster)

Forbindelsen mellom over- og undergurt skiller seg mer ut fra de resterende forbindelsene. Oppbyggingen for dybelgruppen i undergurt kan modelleres på akkurat samme måte som staven i den oransje clusteren, mens dybelgruppen i overgurt har en helt enestående oppbygging. Denne clusteren er markert med brun farge på ikonet, som vist i Figur 6.125.



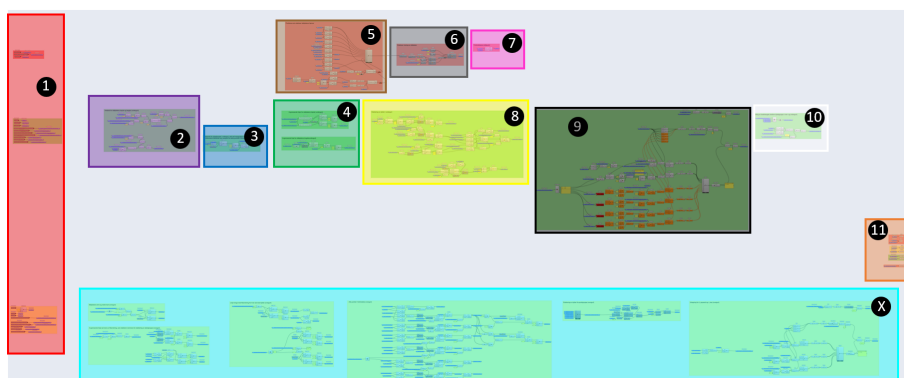
Figur 6.125 Cluster for forbindelse mellom over- og undergurt (brun)

Innsiden av clusteren er vist i Figur 6.126. De grønne gruppene tilhører dybelgruppen i undergurt, mens de gule tilhører gruppen i overgurt. I tillegg viser de røde gruppene programmering som tilhører både over- og undergurt.



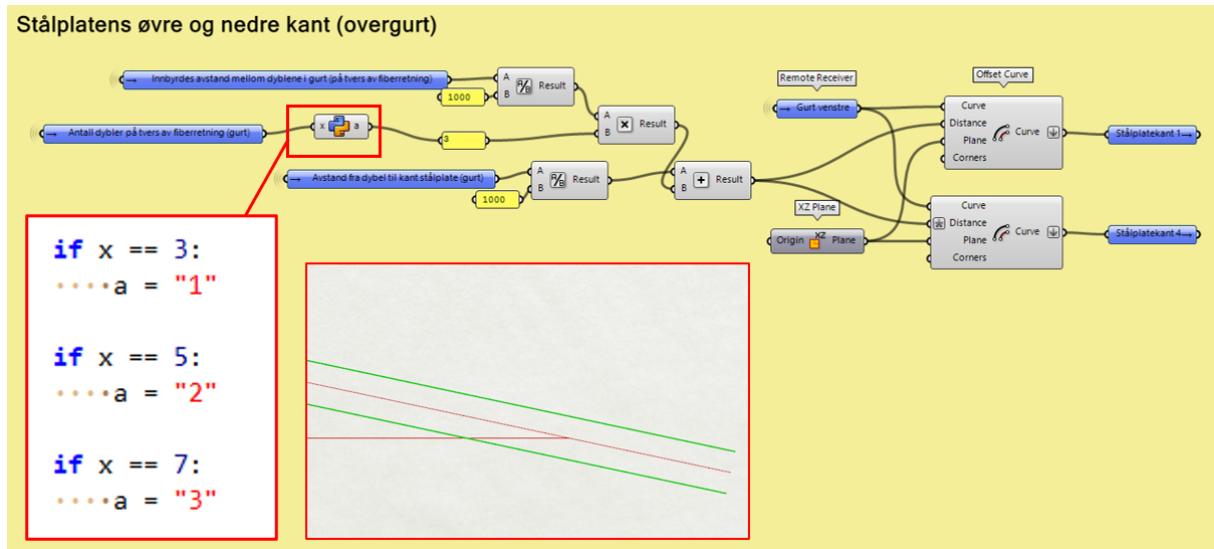
Figur 6.126 Forbindelse mellom over- og undergurt (Brun cluster)

Figur 6.127 viser rekkefølgen på operasjonene som utføres for hver av elementene i den brune clusteren. Dybelgruppen for undergurt har lik rammeinndeling som den oransje clusteren. Innholdet for ramme «1» til «11» kan derfor beskrives på samme måte som i Tabell 6.5. I tillegg er oppbyggingen for undergurt identisk med oppbyggingen for de vertikale staver i den oransje clusteren. Dybelgruppen i overgurten har imidlertid en helt unik oppbygging, og har derfor fått en ramme med navn «X». Videre forklares hvordan denne rammen er bygget opp.



Figur 6.127 Nummerert fremgangsmåte for forbindelse mellom over- og undergurt (Brun cluster)

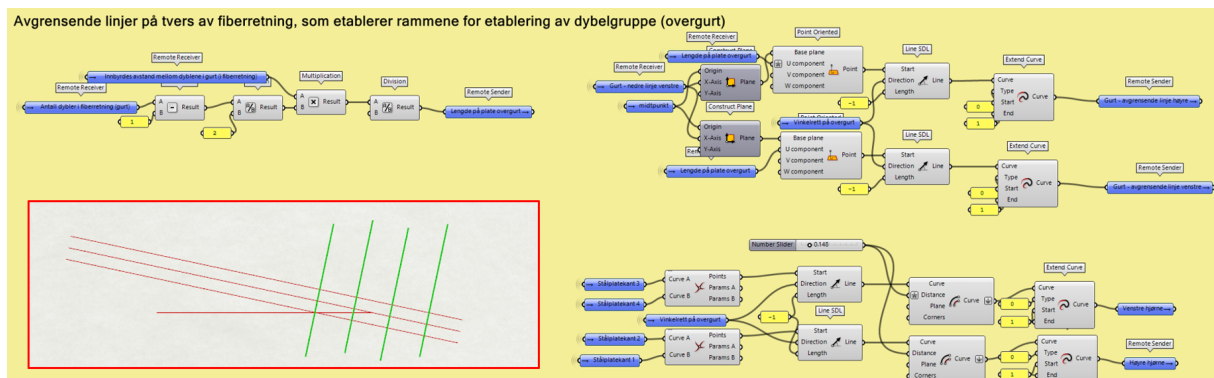
I ramme X sin første gruppe fra venstre, dannes stålplatens ytterkant parallelt med overgurten fiberretning. Gruppen, og dens resultater er presentert i Figur 6.128. Her benyttes «Offset Curve»-komponenten, hvor det defineres en avstand fra overgurten systemlinje. Avstanden er avhengig av antall dybler på tvers av fiberretningen, innbyrdes avstand mellom dyblene og avstand til kant stålplate. Derfor er det laget et script som systematiserer hvor mange «innbyrdes avstander» linjen skal forskyves, avhengig av antall dybler som ønskes på tvers av fiberretningen. Scriptet er dannet for inntil 9 dybler, og vil bare fungere for oddetall. På grunn av dette, er også justeringsmulighetene på utsiden av clusteren satt til å bare kunne oppgi oddetall.



Figur 6.128 Linjer som definerer ståplatens ytterkant parallelt med elementets fiberretning (brun cluster)

Videre dannes avgrensende linjer vinkelrett på fiberretning i neste gruppe, som presentert i Figur 6.129. For å etablere de to midterste grønne linjene, er komponentene «Point Oriented» og «Line SDL» benyttet. Disse linjene er dannet for å avgrense rekken med dybler som ligger i overgurtens systemlinje. Det blir tatt utgangspunkt i midtpunktet (krysningspunktet mellom over- og undergurt), og spesifisert en avstand til hver side av punktet. Avstanden utover til hver side fra midtpunktet, beregnes til øverst til venstre i figur. Beregningen avhenger av innbyrdes avstand mellom dyblene- og antall dybler i fiberretning.

Nederst til høyre i figur, dannes de to ytterste grønne linjene i Figur 6.129. Disse er til for å kappe unødvendig lange hjørner på ståplatene. Komponentene som benyttes er «Curve|Curve», «Line SDL» og «Offset curve». Plasseringen blir satt på grunnlag av linjer som blir etablert i en annen gruppe i ramme X, som forklares senere. Avstanden som er bestemt (0.148) er beregnet med trigonometriske sammenhenger, for å overholde minimumskravet på 25 mm fra dybel til kant ståplate.



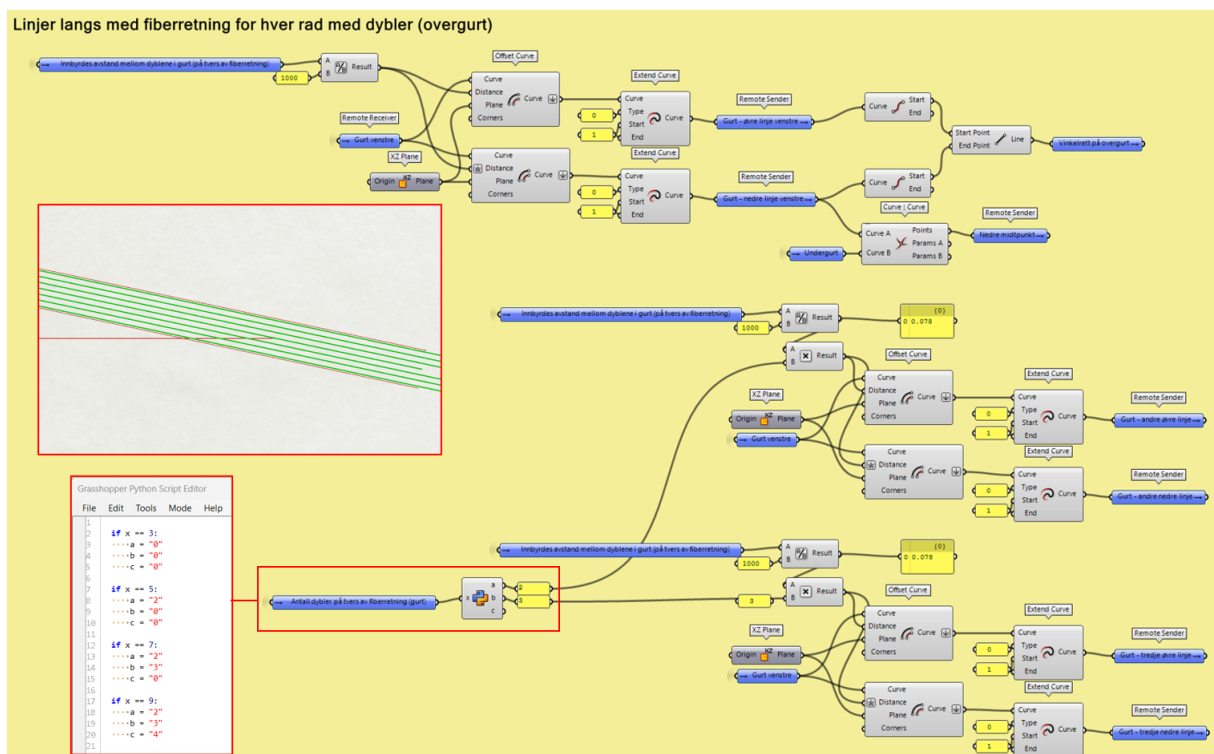
Figur 6.129 Etablering av avgrensende linjer vinkelrett på fiberretning (brun cluster)

På samme måte som i oransje cluster sin femte ramme, blir linjene fra begge gruppene som er omtalt ovenfor benyttet for å etablere ståplatens hjørner- og vendepunkter. Krysningspunktene mellom linjene blir sortert i løpende rekkefølge langs ståplatens ytterkant, sammen med hjørnene til undergurtens ståplate. I tillegg er det programmert funksjoner for noen av punktene som bare skal

eksistere i enkelte tilfeller. Videre blir stålpplatene og midtpunktene etablert på tilsvarende måte som den oransje cluster sin sjette og syvende ramme.

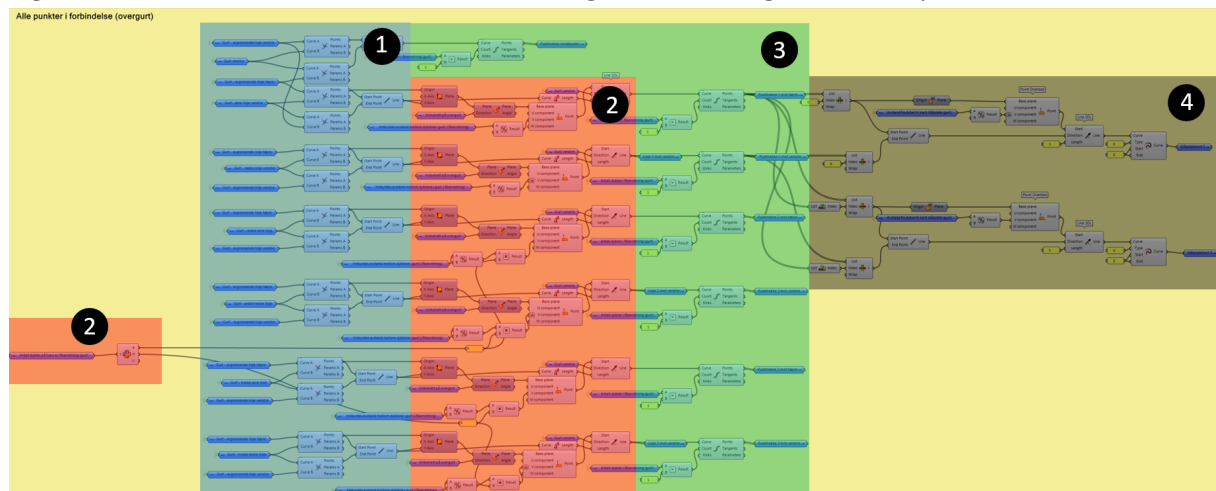
Videre i ramme X, blir det dannet linjer for hver av dybelrekkene langs med fiberretning. Gruppen og dens resultater er presentert i Figur 6.130. Den midterste grønne linjen, er overgurtens systemlinje. De to øverste rekkene med operasjoner, danner de to nærmeste linjene på hver side av systemlinjen. Komponentene som benyttes, gjenkjennes fra tidligere i metodekapittelet. Avstanden mellom linjene, «innbyrdes avstand mellom dyblene på tvers av fiberretning», er bestemt på utsiden av clusteren. Til slutt er det hentet informasjon fra overgurt-linjenes endepunkter, som «retning gurt» og «nedre midtpunkt», på samme måte som i den oransje clusteren.

Deretter gjentas disse to operasjonene to ganger, for de neste linjene på hver side av systemlinjen. Dette kan gjentas videre, for å kunne legge til flere dybler på tvers av fiberretningen. I dette tilfellet er systemet imidlertid avgrenset til å bare kunne inneholde 7 dybler på tvers av fiberretningen. Også dette er lagt begrensninger for på justeringsmulighetene på utsiden av clusteren. For at systemet skal fungere parametrisk, er det laget et script som sier hvilken av linjene som skal fungere ved ulikt antall dybler på tvers av fiberretningen. Ved tre rekker med dybler på tvers, skal bare de to nærmeste linjene på hver side av systemlinjen fungere. Derfor blir verdien «0» sendt til de andre operasjonene, slik at de får avstand «0» fra systemlinjen. Derimot, skal operasjonen som definerer andre linje fra systemlinjen ha avstanden «2 ganger innbyrdes avstand», dersom det er valgt fem rekker med dybler på tvers. I tillegg skal operasjonen som definerer tredje linje fra systemlinjen ikke eksistere, og dermed ha avstanden «0 ganger innbyrdes avstand». Slik fortsetter også scriptet videre for sju rekker med dybler.



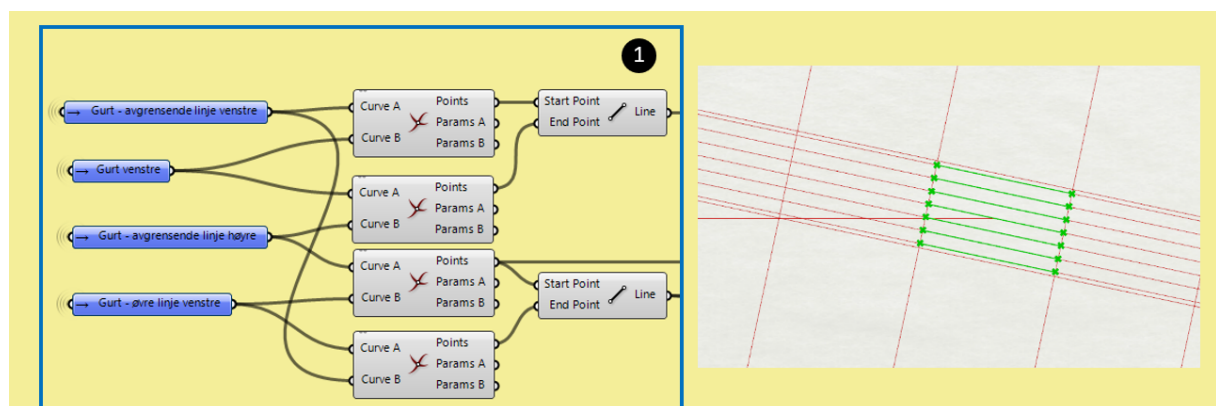
Figur 6.130 Linjer for hver av dybelrekkene langs med fiberretning (brun cluster)

Neste gruppe danner punktene i overgurtens dybelgruppe, og kan deles inn med 4 rammer som vist i Figur 6.131. Disse fem rammene blir forklart i stigende rekkefølge videre i kapittelet.



Figur 6.131 Gruppe som danner punktene i overgurtens dybelgruppe (brun cluster)

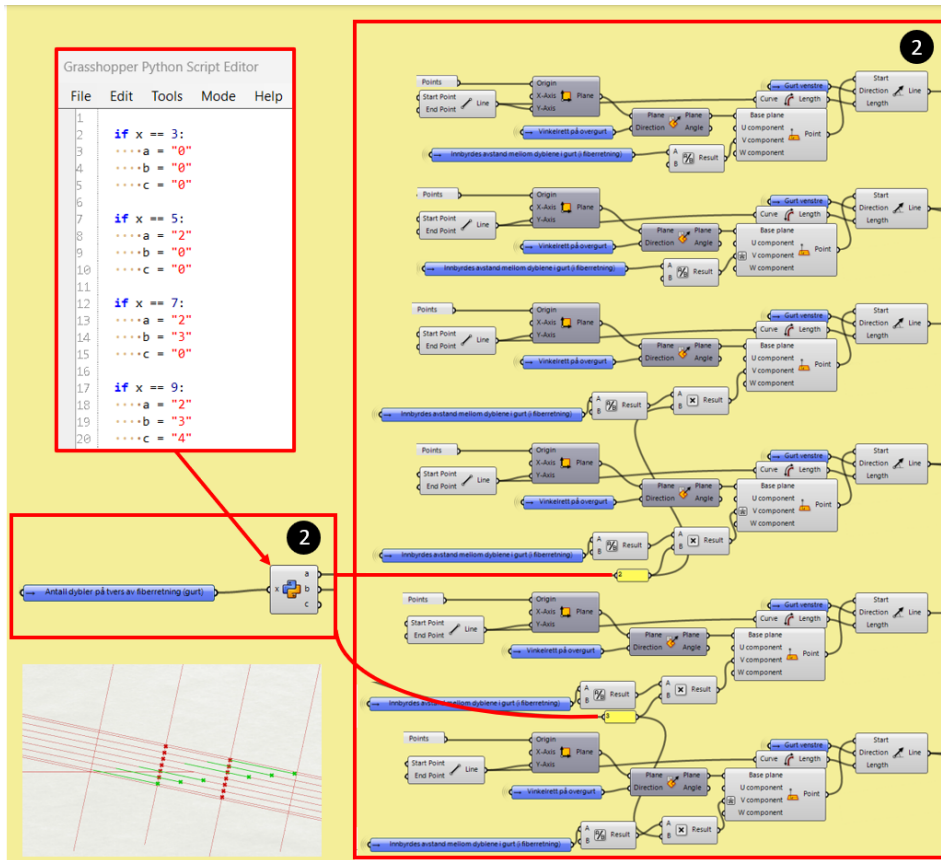
Et utklipp fra første del av gruppen, er presentert i Figur 6.132. Her benyttes de to midterste linjene vinkelrett på overgurtens fiberretning, som ble forklart tidligere. Disse er dannet for å avgrense rekken med dybler som ligger i overgurtens systemlinje. Det blir laget krysningspunkter mellom disse- og hver av linjene i fiberretning. I tillegg bygges en linje mellom krysningspunktene. Det er imidlertid bare rekken med dybler som ligger i overgurtens systemlinje som skal ligge på linjen som er laget her. De resterende linjene er laget som et utgangspunkt for å lage en forskjøvet linje langs med fiberretningen.



Figur 6.132 Linjer som utgangspunkt for forskjøvne linjer (Brun)

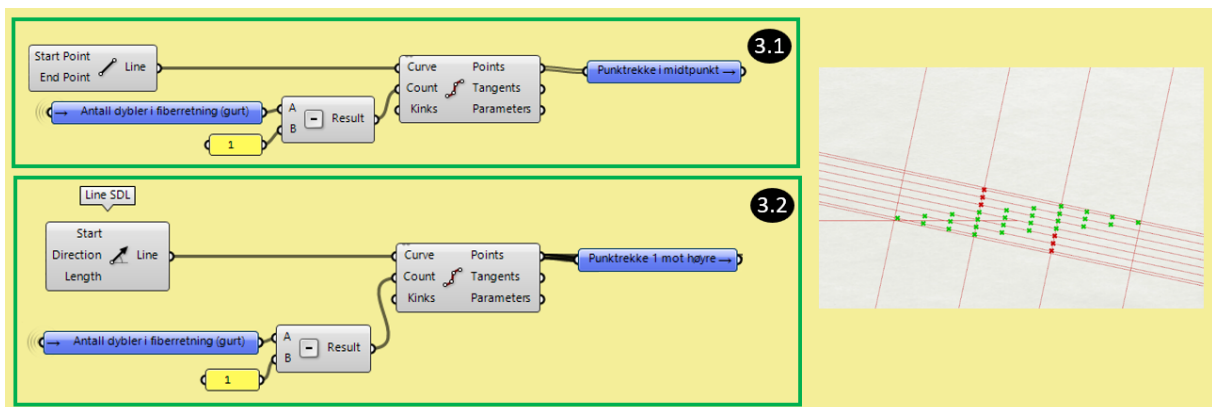
I neste ramme av gruppen dannes linjene som blir forskjøvet langs med fiberretningen, som vist i Figur 6.133. De to første linjene på hver side av systemlinjen, blir forskjøvet med valgt innbyrdes avstand i fiberretning. Den ene forskyves mot venstre, mens den andre forskyves mot høyre. Lengden av linjen fra rammen som ble forklart ovenfor, settes til å være den samme lengden i den forskjøvne linjen. Linjen brukes også for å bestemme hvilket plan den forskjøvne linjen skal ligge i. Deretter gjentas operasjonen for de resterende linjene på hver side av systemlinjen. Neste linje skal forskyves med 2 innbyrdes avstander, og neste deretter skal forskyves med 3 innbyrdes avstander osv. Også her er det benyttet script, for at systemet skal fungere parametriske. Scriptet sier hvor mange innbyrdes avstander linjene skal forskyves ved forskjellige antall dybler på tvers av fiberretningen. Ved tre rekker med dybler på tvers, skal bare de to nærmeste linjene på hver side av

systemlinjen flyttes 1 innbyrdes avstand. Ved fem rekker, skal også de neste linjene forskyves ett hakk. Slik fortsetter systemet videre for inntil 9 dybler (men systemet fungerer så langt bare for 7 dybler på tvers).



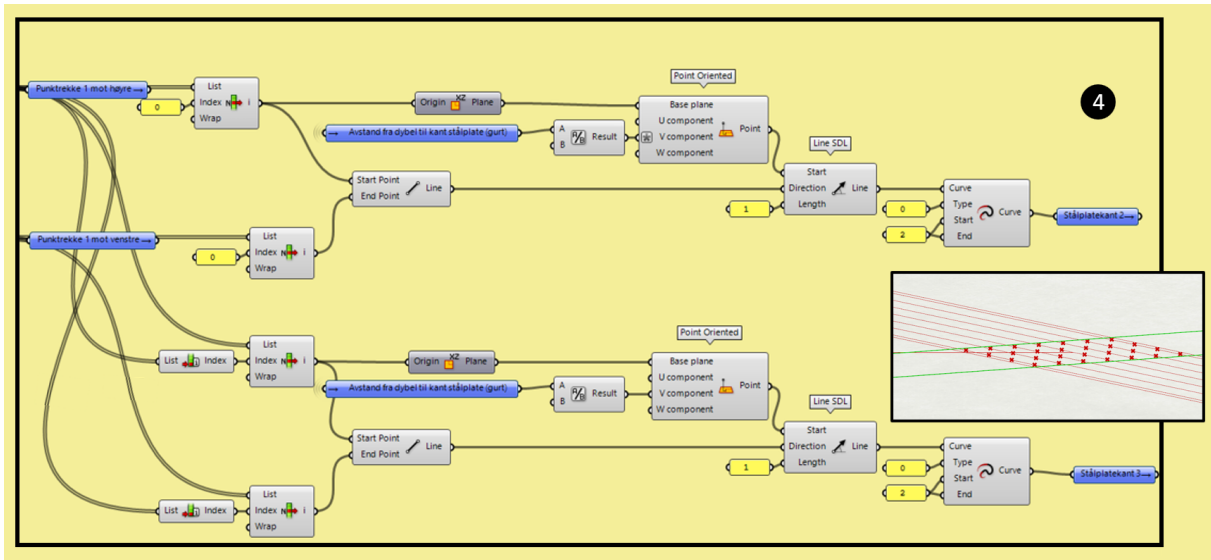
Figur 6.133 Etablering av forskjøvne linjer langs med overgurtens fiberretning (Brun cluster)

I tredje ramme dannes punktene i fiberretning hvor dyblene skal plasseres. For linjene på systemlinjen, som vist i ramme 3.1 i Figur 6.134, settes punktene direkte på linjen som ble dannet i første ramme. Komponenten «Divide curve», deler linjene i riktig antall deler (som bestemmes av antall dybler i fiberretning). Det samme utføres for alle de andre linjene, som vist i ramme 3.2 i Figur 6.134. Forskjellen er at punktene settes på de forskjøvne linjene fra ramme 2, og ikke linjene fra ramme 1.



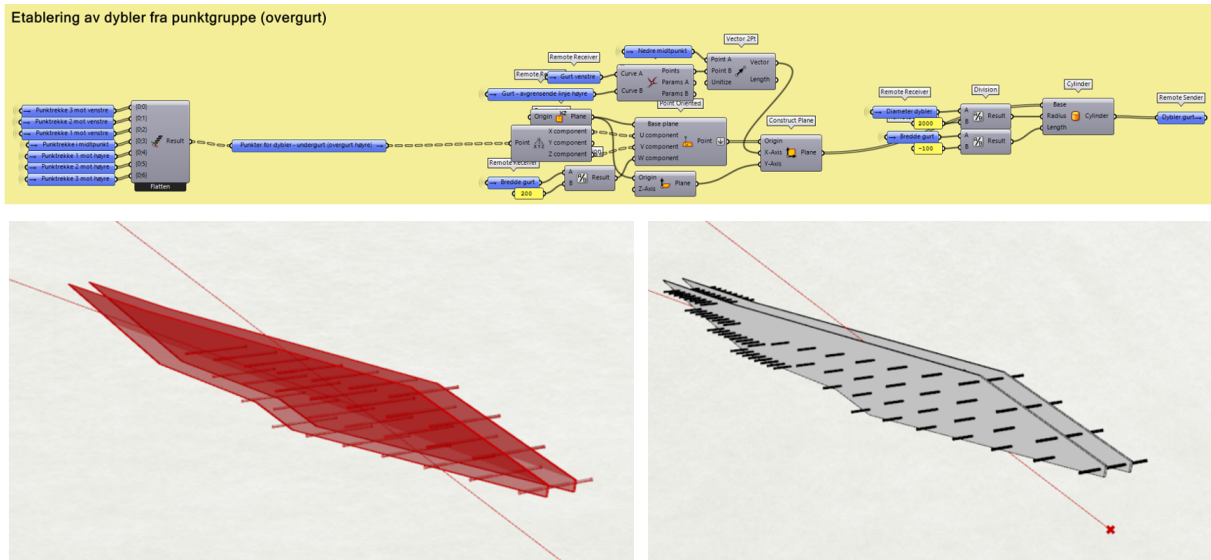
Figur 6.134 Etablering av punkter på linjene i fiberretning (Brun cluster)

Til slutt hentes de første og siste punktene i hver av dybelrekkene, for å danne ytre linjer som går parallelt med dyblene på tvers. Operasjonene er utført med komponenter som er kjent fra tidligere i metodekapittelet, som vist i Figur 6.135. Linjene benyttes for å lage stålplatenes hjørner og vendepunkter. I tillegg er de grunnlaget for plassering av linjene som kapper de unødvendig lange hjørnene på stålplaten, som forklart tidligere.



Figur 6.135 Ytre linjer som går parallelt med dyblene på tvers i overgurten (brun cluster)

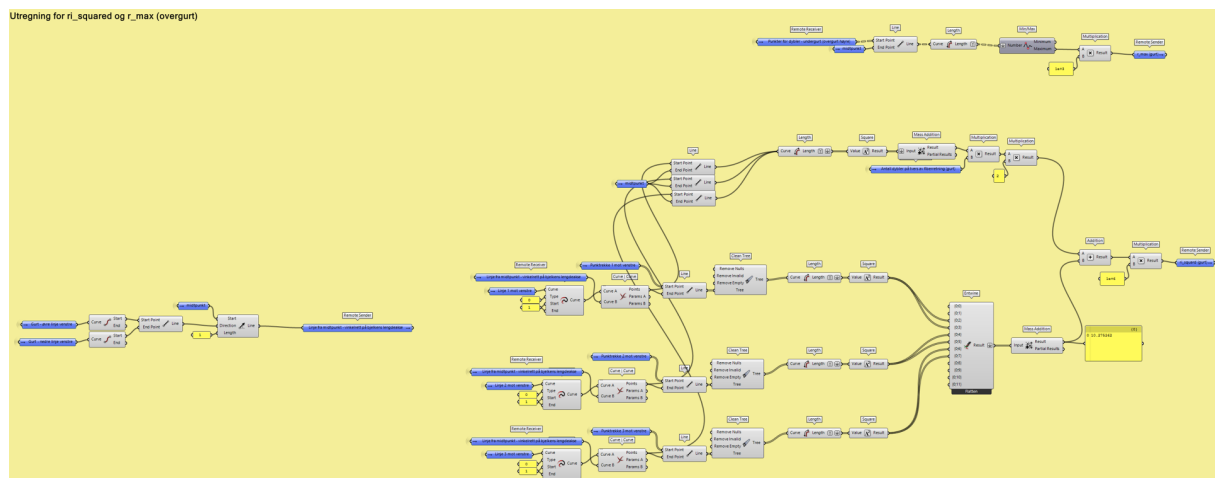
Dermed samles alle punktene med «Entwine»-komponenten, og dyblene bygges på samme måte som den oransje clusterens ramme 8. Programmeringen og resultatet av dette, er vist i Figur 6.136.



Figur 6.136 Etablering av dybler (Brun cluster)

For måling av avstander for beregner $r_{i_squared}$ og r_{max} , bygges systemet nesten på samme måte som i oransje cluster. Programmeringen er vist i Figur 6.137. Den eneste forskjellen, er at det ikke trengs å lage linjer i dybelgruppene parallelt med elementets fiberretning, fordi disse linjene allerede er laget i de tidligere gruppene. Scriptet fra den oransje clusteren kunne ikke benyttes for denne

dybelgruppen, fordi listen med dybler ikke har lik oppbygging som tidligere. Det må undersøkes ytterligere om hvordan dette systemet fungerer parametrisk.



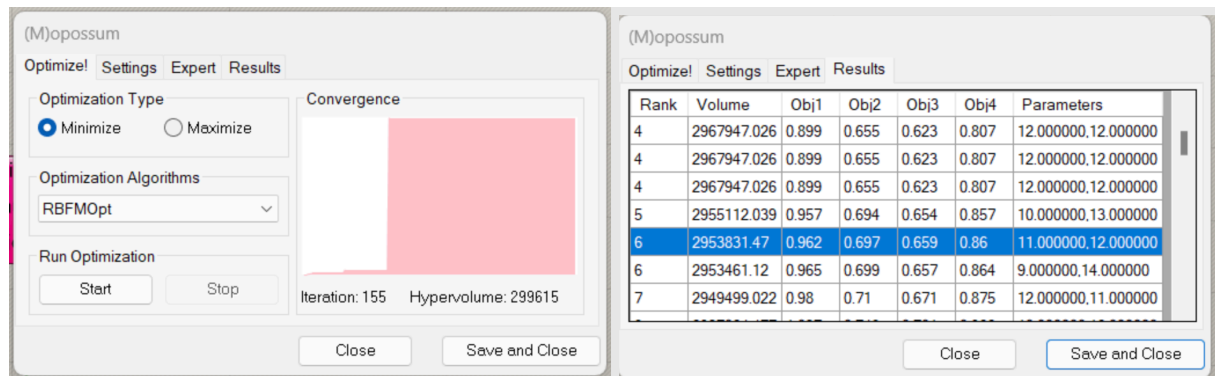
Figur 6.137 Måling av avstander for beregner r_i _squared og r _max (Brun cluster)

6.5 Metode for optimalisering

I dette delkapittelet forklares metoden som ble brukt for å optimaliseres limtrefagverket i Grasshopper. Formålet med optimaliseringen var å redusere totalvekten til fagverket, og komponenten «Assemble Model (Karamba3D)» gir totalvekten til fagverket i kilogram, ut ifra densiteten til det valgte limtrematerialet. Det ble gjort flere mislykkede forsøk før den endelige metoden ble valgt.

De fleste optimaliseringsverktøyene som kan brukes i Grasshopper kan kun ha ett formål, og ofte kan disse kun gjør formålet minst eller størst mulig. Det ble først forsøkt å bruke Galapagos som kun kan ha et formål, men siden det må tas hensyn til flere kontroller for hvert element, ble dette et mislykket forsøk. Det ble dermed prøvd å bruke verktøyet Octopus, ettersom denne kan ha flere enn ett formål. Problemet med dette verktøyet, var at det ikke fungerte og Grasshopper-modellen avsluttet flere ganger uten forvarsel. Grunnen til dette var at Octopus ikke kan brukes når verktøyet «Human UI» og «Wallacei» var nedlastet i Grasshopper. Octopus bruker «HelixToolkit», og når et annet verktøy bruker dette, oppstår det en programsvikt i Grasshopper [168]. I tillegg var det utfordrende å velge mellom optimaliserings alternativene i Octopus komponenten, noe som gjorde at vi heller valgte «Opossum» som optimaliseringsverktøy.

Opossum kan ha flere optimaliserings formål, og det var i tillegg enkelt å velge mellom løsningene den genererte. Dette er illustrert til høyre i Figur 6.138. Til venstre i samme figur vises optimaliseringstypen, og algoritmen som kan velges. I denne oppgaven er det kun brukt «RBFMOpt», ettersom det ikke var tid til å utforske flere algoritmer.



Figur 6.138: Opossum valg av optimalisering og resultater

Det ble også valgt å ha en egen Grasshopper-fil for optimaliseringsdelen, der de fleste av forbindelsesberegningene og visualiseringen av dem ble fjernet. Grunnen til dette er at beregningene av for hvert knutepunkt, gjorde Grasshopper-modellen veldig treg. For eksempel tok det over ti sekunder å kun endre navn på noe i modellen. I en helt ny Grasshopper-modellen vil endring av navn skje umiddelbart. Grasshopper-filen for optimalisering er gitt i Vedlegg 6.

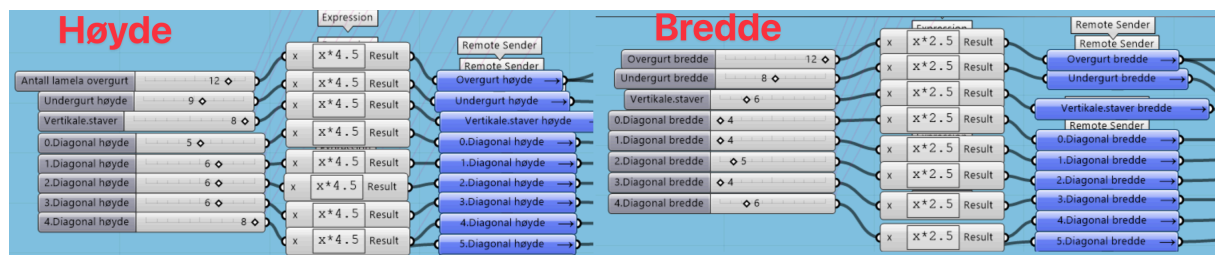
6.5.1 Metode for størrelsesoptimalisering av limtrefagverket

For å optimalisere størrelsen på tverrsnittene, ble det gjort en del forandringer på parameterne som bestemmer tverrsnittstørrelsen. Grunnen til dette var at Opossum skapte tverrsnitt som ikke var byggbare, når den var tilkoblet disse parameterne. Dette gjorde at antallet på variabler og resultater som skapte urealistiske tverrsnitt, måtte dermed fjernes fra disse parameterne.

Parameterne som bestemmer høyden til tverrsnittene, ble forandret fra å ha verdier mellom 1 og 100, til mellom 2 og 14 for gurtene, og mellom 2 og 8 for stavene. Hver av disse tallene representerer antallet limte lameller som elementet består av. Disse lamellene er 45 millimeter høye og antallet av lameller multipliseres med denne høyden. Dette er basert på Moelvns limtre [169]. I Figur 6.139 vises det at hver verdi fra parameterne blir multiplisert med 4.5 for høyde og 2.5 for bredden, siden Karamba3D opererer med centimeter.

Det er valgt å ha forskjellig antall verdier i parameterne som bestemmer tverrsnittet til stavene og gurtene, slik at Opossum ikke skal generere staver med høyere høyde enn gurtene. Dette er også utført for parameterne som bestemmer tverrsnitts-bredden, men her er verdien for gurtene, vertikale, og diagonalstavene 3 og 4, mellom 4 og 12. For diagonalstavene 0 og 1 er verdien mellom 4 og 9, ettersom disse stavene har mindre tverrsnitt enn alle andre elementene i fagverket fra casen.

Bredden bestemmes ved å multiplisere antallet av lameller med 25 millimeter, dette er vist i Figur 6.139, der hver lamell multipliseres med 2.5. Dette følger Moelvns limtre bredde, ettersom hvert limtre-element økes med 25 millimeter [169]. Dette vil derimot ikke gi samme bredde som Sweco sine elementer, ettersom i realiteten bestemmes bredden av antallet på limtreene med breddene på 90, 115 og 140 millimeter. For eksempel brukes det i Sweco sitt tilfelle brukes det to limtre med 90 og et med 115 millimeter, noe som gir en bredde på 295 millimeter. Det som er laget i Grasshopper-modellen vil gi en bredde på 300 millimeter.



Figur 6.139: Korrigert høyde og bredde parametere for tverrsnitt

I Opossum ble «minimize» avkrysset som optimaliseringstype. Dette gjorde at algoritmen vil søke etter det laveste målet. I dette tilfellet ble utnyttelsen til elementene brukt som mål for å redusere vekten til fagverket. Det ble gjort et forsøkt på å bruke «Mass» outputen til «Assemble Model»-komponenten til Karamba3D for formålet til optimaliseringen, men dette skapte unormalt tynne tverrsnitt, og elementene var lagt over utnyttelseskravene. Det ble dermed bestemt å heller bruke de største utnyttelses verdiene fra Case-modellen til Sweco som formålet til optimaliseringen. I Opossum kan det velges mellom alle alternativene som genereres, slik at den optimale utnyttelsen for elementet kan velges.

Dersom «maximize» hadde blitt valgt som optimaliseringstype, ville Opossum søkt etter den største utnyttelses verdien. Dette hadde dermed skapt tverrsnitt som hadde vært unormalt tynne, ettersom komponenten som kontrollerer utnyttelses verdiene gir kun verdier av utnyttelsene, og kan dermed ikke stoppe Opossum i å generere elementer med over 100% utnyttelse. Det er heller ikke mulig å gi Opossum en maksimal grenseverdi for utnyttelsen, dette gjør at å søke etter den minste verdien er det beste alternative.

Det er kun valgt å optimalisere fagverket fra casen. Dermed er ikke alle parameterne som bestemmer dimisjonene til de seks andre diagonalstavene som kan legges inn i fagverket korrigerte. For hvert element som skal optimaliseres, er det tilkoblet en egen Opossum-komponent. Alle iterasjonen som genereres av Opossum ble kopiert og limt inn i et Excel-ark.

For å undersøke om resultatet tilfredsstillende knutepunkts kontrollene, måtte tverrsnittene i Grasshopper-modellen med alle Knutepunktene oppdateres.

Optimalisering av overgurt

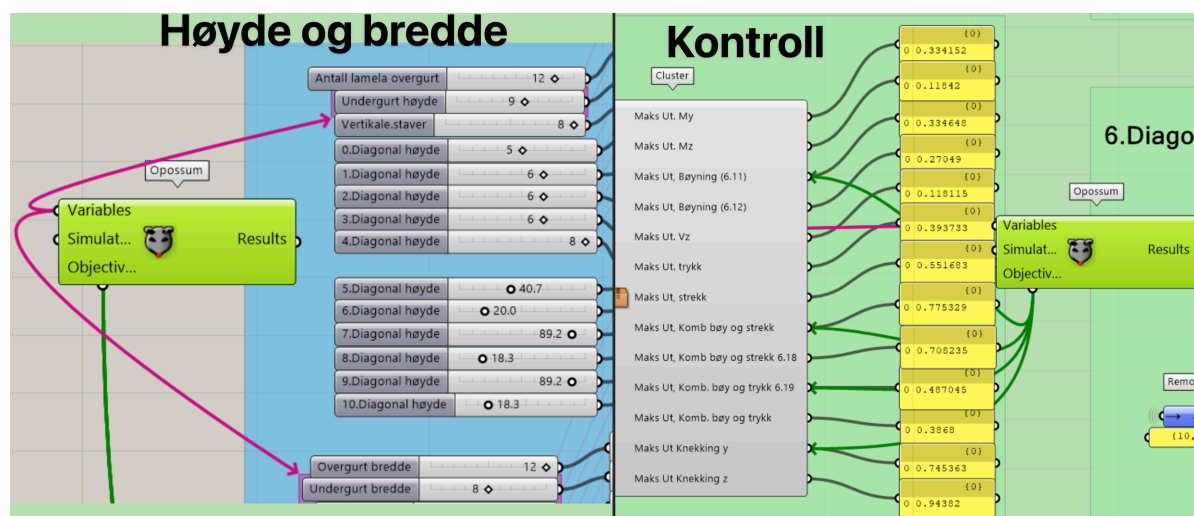
For størrelsesoptimalisering av overgurtene ble parameterne som bestemmer høyden og bredden til overgurtene valgt som «Variables» input i Opossum. Utnyttelsene som ble valgt som «Objective»-input i Opossum var bøyning 6.11, kombinasjon bøy og strekk 6.17, kombinasjon bøy og trykk 6.19 og knekking y-retningen. Disse ble valgt ettersom de hadde høyest utnyttelsesverdier fra sammenligningen med Sweco. Denne prosessen er vist i Figur 6.140. Totalt ble det utført 150 iterasjoner av Opossum for overgurtene, før vi stoppet den.



Figur 6.140: størrelsesoptimalisering av overgurtene

Optimalisering undergurt

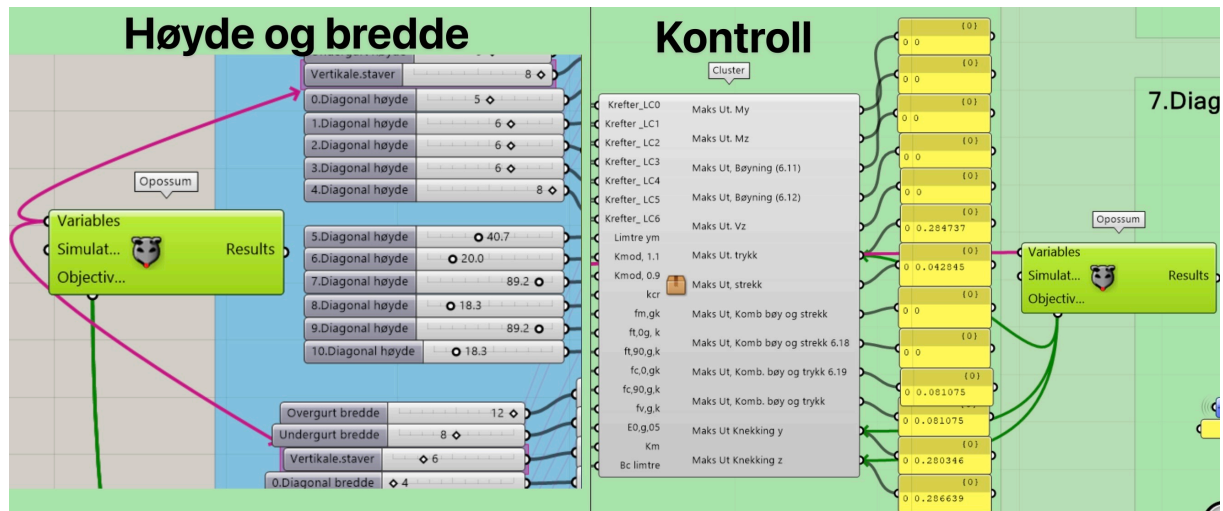
For størrelsesoptimalisering av undergurten ble parameteren som bestemmer høyde og bredde til undergurten valgt som variabel for Opossum. Målene som skulle reduseres var de sammen som overgurten. Det ble totalt utført 150 iterasjoner før vi stoppet Opossum, og denne prosessen er vist i Figur 6.141.



Figur 6.141: Størrelsesoptimalisering av undergurt

Optimalisering vertikale staver

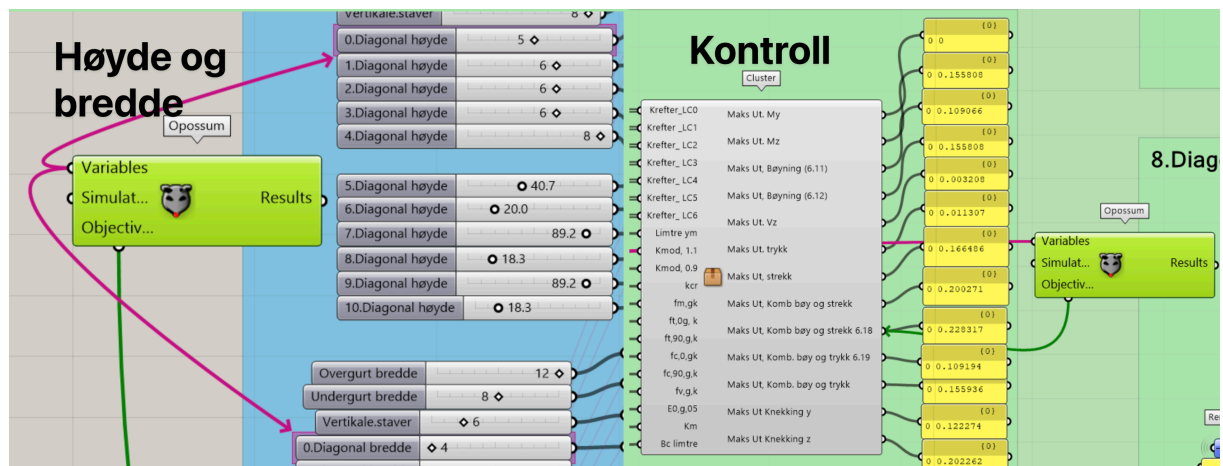
For de vertikale staver er variablene høyden og bredden til de vertikale staverne. Formålet var bøyning 6.11, kombinasjon bøy og strekk 6.17, kombinasjon bøy og trykk, og knekking y-akse. Det ble kun utført 5 iterasjoner før Opossum stoppet selv. Denne prosessen er vist i Figur 6.142.



Figur 6.142: Størrelsesoptimalisering av vertikale staver

Optimalisering av diagonal 0

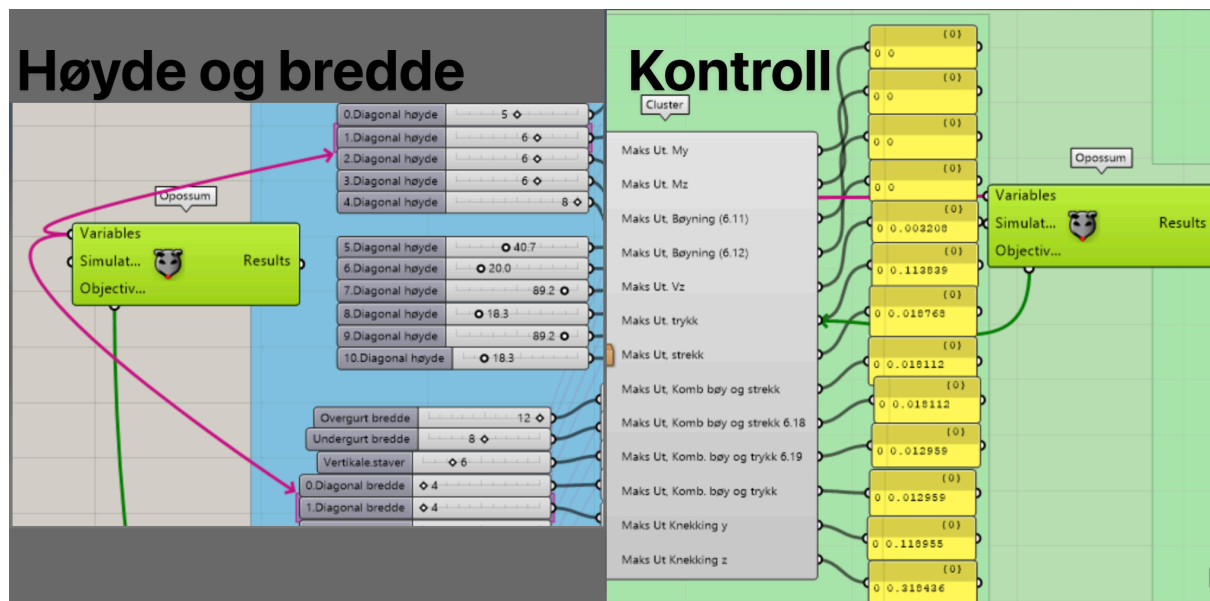
For diagonal 0, ble parameterens som bestemmer høyden og bredden valgt som variabler. Det ble kun valgt et objekt som skulle reduseres, ettersom alle andre utnyttelser var generelt lave. Kombinasjon bøyning og strekk 6.18 ble valgt som formålet for Opossum. Det ble utført totalt 38 iterasjoner før vi stoppet Opossum. Grunnen til dette var at utnyttelses verdien var allerede ganske lav, og det var dermed ikke nødvendig søke etter lavere utnyttelsesverdi. Prosessen er vist i Figur 6.143.



Figur 6.143: Størrelsesoptimalisering av diagonal 0

Optimalisering av diagonal 1

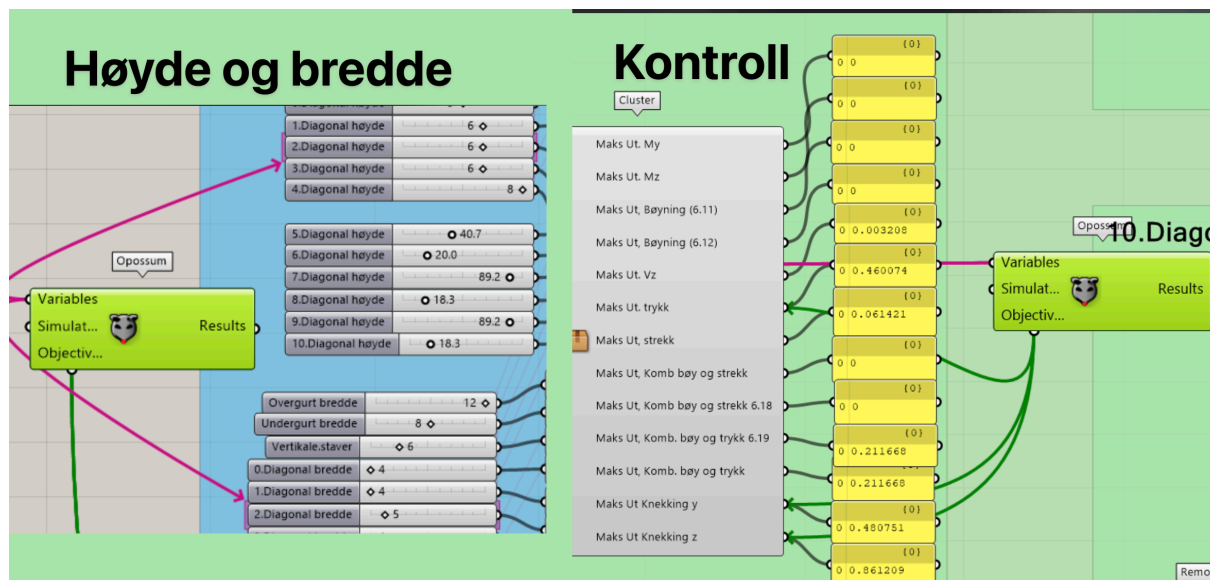
For størrelsesoptimalisering av diagonal 1, ble det også kun valgt et formål. Her ble utnyttelsen av trykk valgt som formål, og igjen ble parameterne som bestemmer høyden og bredde til disse staverne valgt som variabler for Opossum. Totalt ble det utført 37 iterasjoner før vi stoppet Opossum. Igjen var det ikke nødvendig i la Opossum redusere utnyttelsen mest mulig, siden det allerede var lav utnyttelse fra før av. Prosessen er vist i Figur 6.144.



Figur 6.144: Størrelsesoptimalisering av diagonal 1

Optimalisering av diagonal 2

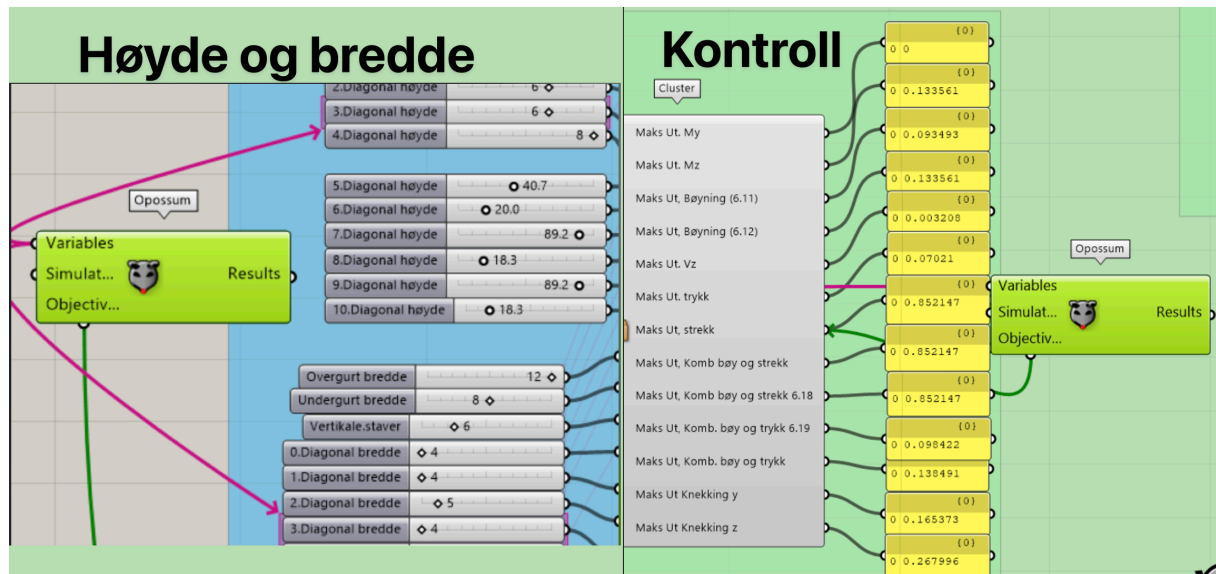
For diagonal stavene 2, ble utnyttelse trykk, og knekking både i y- og z-aksen valgt som optimaliseringsformål. Disse var de største utnyttelsesverdiene fra casen, og derfor valgt. Variablene var igjen høyden og bredden til stavene. Totalt ble det utført 143 iterasjoner, før vi stoppet Opossum. Prosessen er vist i Figur 6.145.



Figur 6.145: Størrelsesoptimalisering av diagonal 2

Optimalisering av diagonal 3

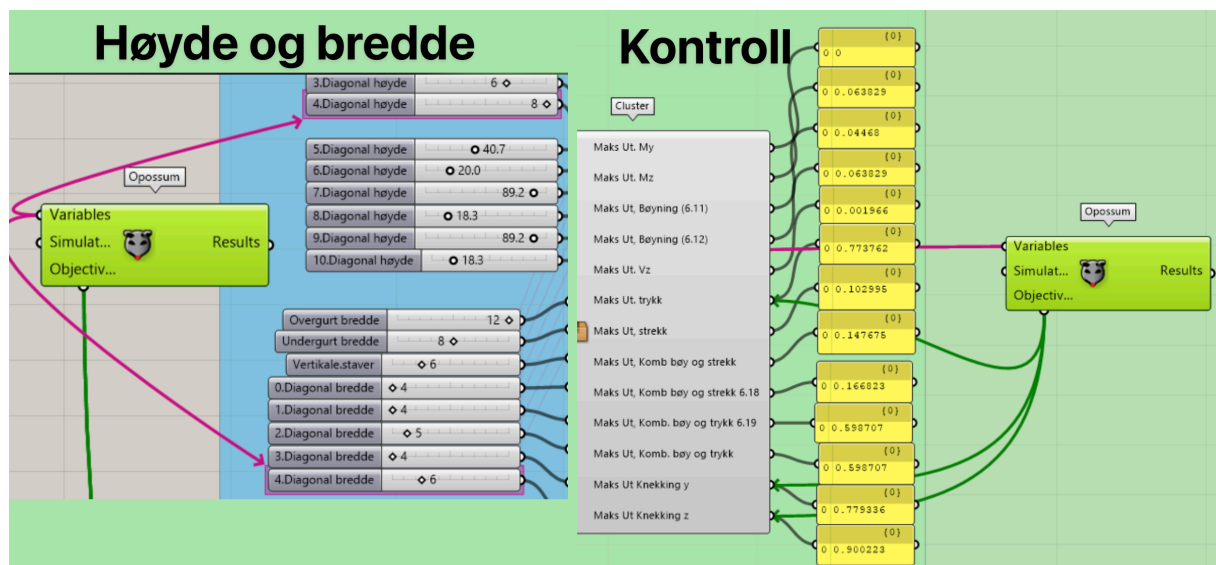
For diagonal 3, ble kun utnyttelse for strekk valgt som formål for Opossum, ettersom dette var den største utnyttelses verdien fra casen. Variablene var både høyden og bredden til disse stavene. Totalt ble det utført 56 iterasjoner før vi stoppet Opossum. Denne prosessen er vist i Figur 6.146.



Figur 6.146: Størrelsesoptimalisering av diagonal 3

Optimalisering av diagonal 4

Utnyttelse for trykk og knekking både i y- og z-retning ble valgt som formålet for Opossum. Ettersom disse var de største utnyttelsene fra casen. Variablene var høyde og bredden til tverrsnittet, og det ble totalt utført 143 iterasjoner, før vi stoppet Opossum. Denne prosessen er vist i Figur 6.147.



Figur 6.147: Størrelsesoptimalisering av diagonal 4

7 Resultat

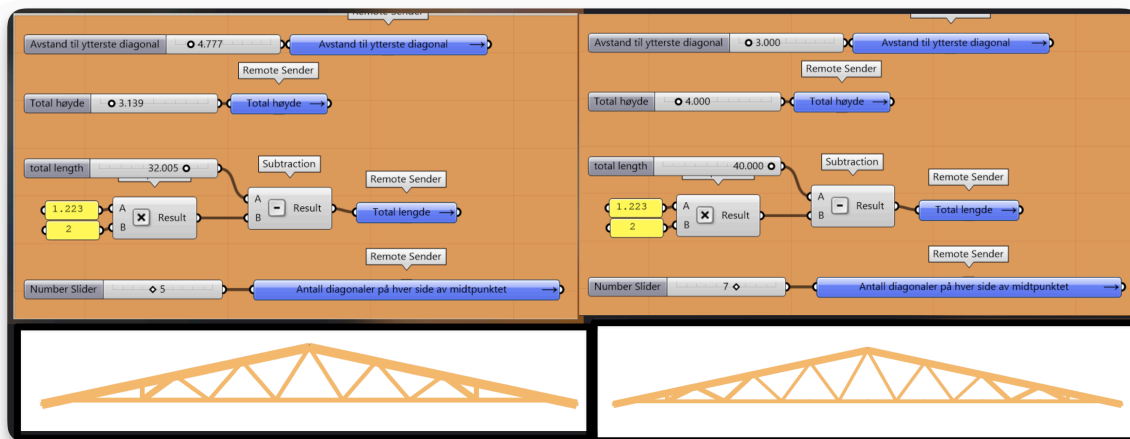
I dette kapittelet presenteres først det parametriske limtre-fagverket som ble programmert i Grasshopper. Deretter presenteres resultatet for sammenligningen mellom belastningskreftene fra Karamba3D og Robot modellen til Sweco. Videre presenteres sammenligningen av elementkontrollen. Deretter moment-, skjær- og aksialkraftdiagrammet og knutepunkts-kontrollene fra Grasshopper og Sweco. Deretter vises utnyttelsen av stålplatene i knutepunktene og til slutt resultatet fra optimaliseringen.

7.1 Parametrisk modell

I dette underkapittelet presenteres det parametriske limtre-fagverket som ble visuelt programmert i Grasshopper og visualisert i Rhino3D. Grasshopper-modellen ble programmert slik at, dersom det gjøres endringer i geometrien til limtre-fagverket, vil kontrollanalysen av strukturen oppdatert automatisk i henhold til endringen. Videre presenteres eksempler på hvordan strukturen endres basert på endringer av variablene til strukturen.

7.1.1 Endring av strukturen til fagverket

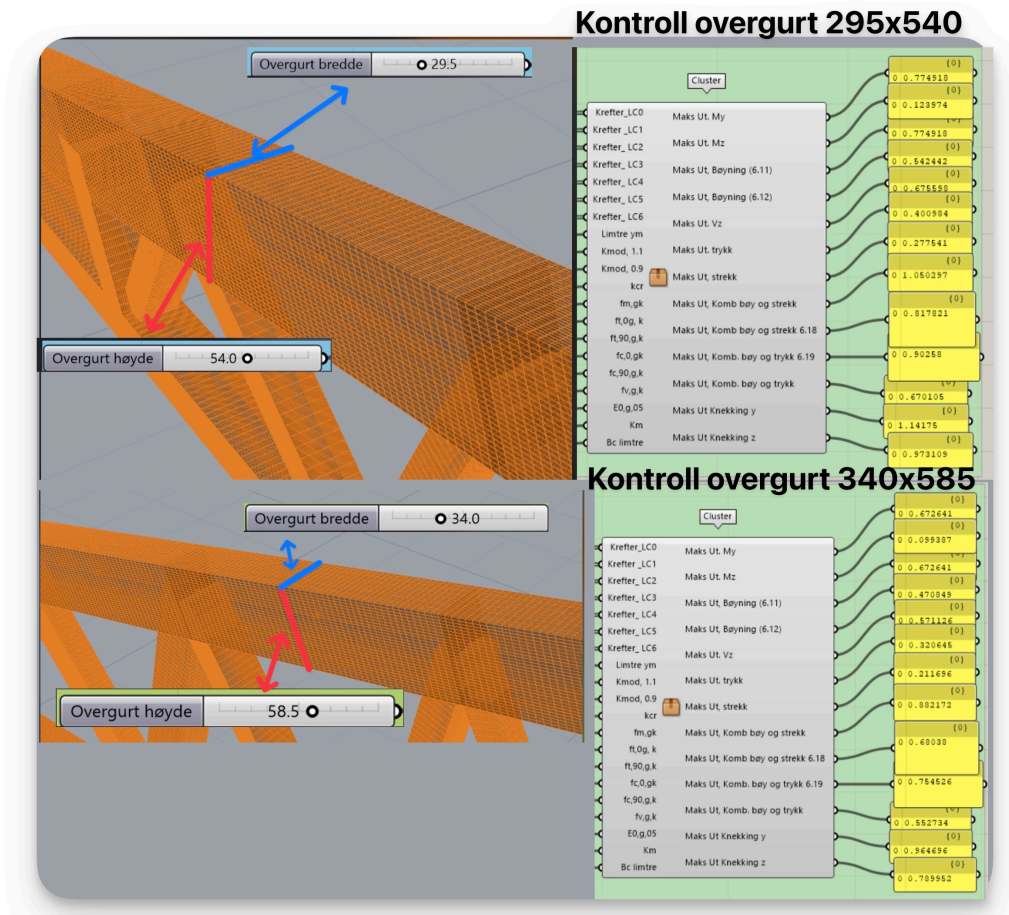
Strukturen til limtre-fagverket er definert av fire parametere, og oppdateres i henhold til verdiene i disse. Dette er parameter som bestemmer lengde, høyde, antall staver og avstander til oppleggene. I Figur 7.1 er det vist to fagverkt som er generert i henhold til verdiene i parameterne. Til venstre i samme figur vises limtre-fagverket fra Casen, mens til høyre vises et tilfeldig generert limtre-fagverkt, med lavere høyde, lengde spenn og flere antall staver.



Figur 7.1: Variablene for parametriske design av fagverket

7.1.2 Endringer av tverrsnitt

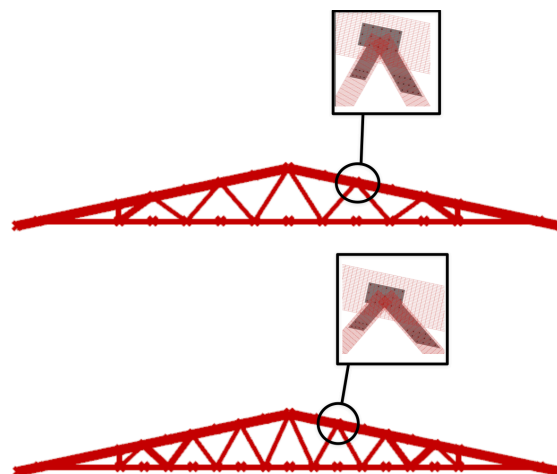
Det kan maksimalt være 11 staver i fagverket og dimensjonen på hvert tverrsnitt defineres av parametere. I Figur 7.2 vises to eksempel på å endre dimensjonen til overgurten. Når dimensjonen endres, utføres det en ny kontroll av elementene og knutepunktene automatisk. Dette gjør det mulig å prøve forskjellige dimensjoner på hvert tverrsnitt i fagverket og direkte få tilbakemelding på utnyttelsen.



Figur 7.2: Oppdatering av kontrollanalyse ved endring av parametere

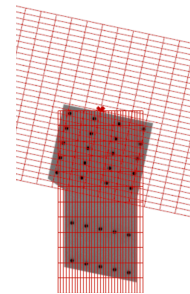
7.1.3 Endringer i forbindelse

Visualiseringen og kontrollen knutepunktene vil automatisk direkte oppdateres ved endringer av modellen. I Figur 7.3 vises hvordan visualiseringen av stålplatene og dyblene forandres basert på strukturen til limtre-fagverket. I dette tilfellet ble antall staver økt i fagverket og dybel-forbindelsen ble endres basert på dette.

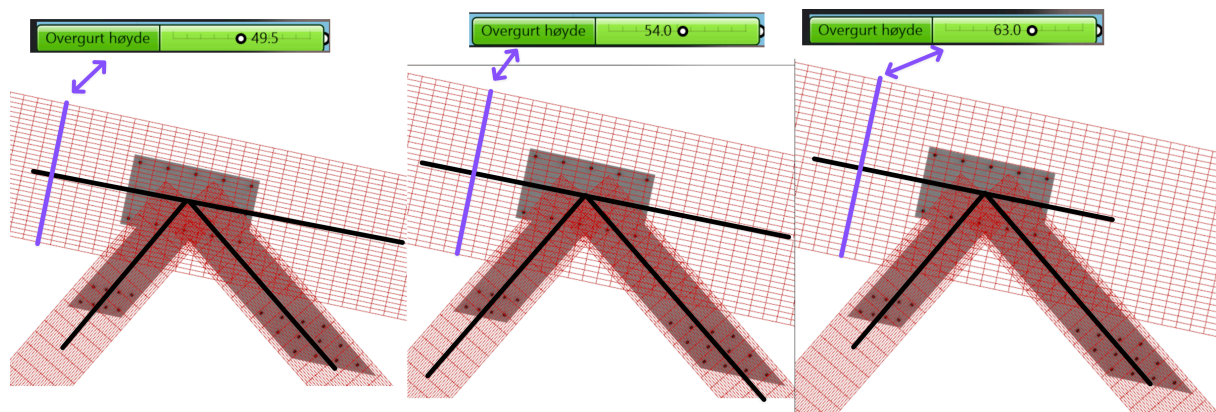


Figur 7.3: Oppdatering av stålplate og dybler når antallet av staver endres

I Figur 7.5 vises det at dybel-forbindelsen vil ta hensyn til å plassere seg slik at systemlinjene skjærer seg i samme punkt (svarte linjer) når høyden til elementet økes. Den eneste forbindelsen som ikke tar hensyn til dette er forbindelsen i den vertikale staven, se Figur 7.4. Forbindelsen i den vertikale staven er flyttet ned i overgurtene, for å redusere bruken av stål. Stålplaten og dybelene vil ta hensyn til eventuelle endringer av dimensjoner til elementene. Dette er også vist i Figur 7.5, der stålplatene forlenges og dybelene omplasseres i stavene når høyden til overgurtene forandres.

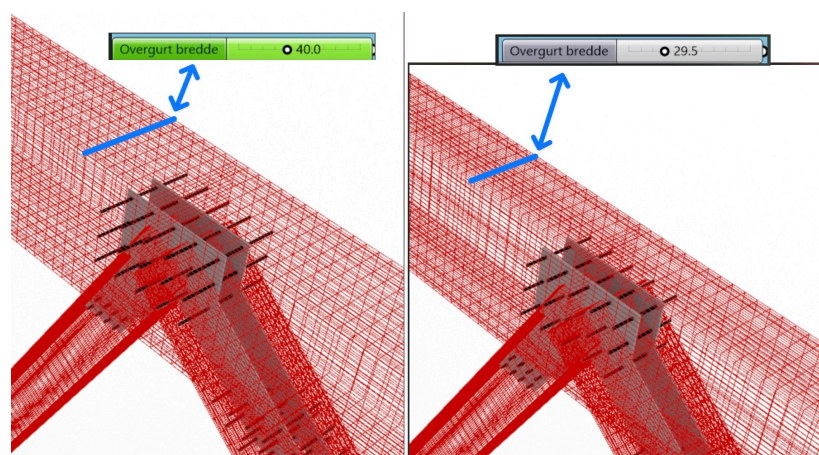


Figur 7.4:
Forbindelse i
vertikal stav



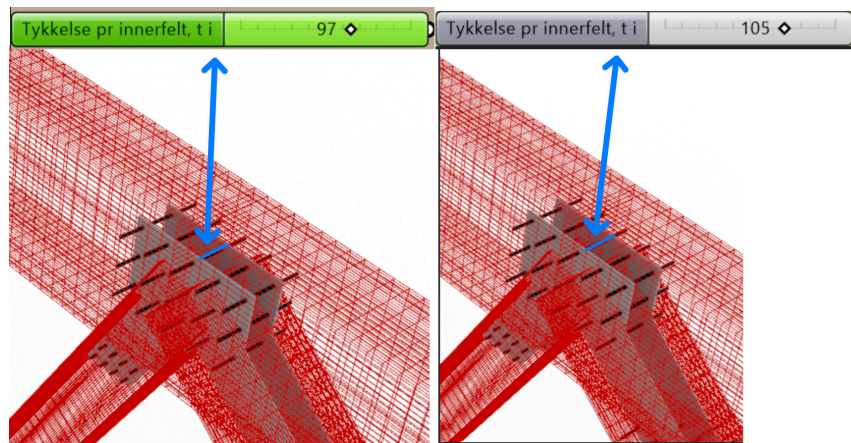
Figur 7.5: Oppdatering av dybeler og stålplater, ved endring av tverrsnitt høyde

Når bredden til tverrsnittet forandres, vil kun lengden til dybelene oppdateres i henhold til dette. Dette er vist i Figur 7.6, der bredden til overgurtene er endret fra 40 cm til 29.5 cm, og lengden til dybelene oppdateres. Avstanden mellom stålplatene oppdateres ikke, ettersom denne avstanden bestemmes av en egen parameter.



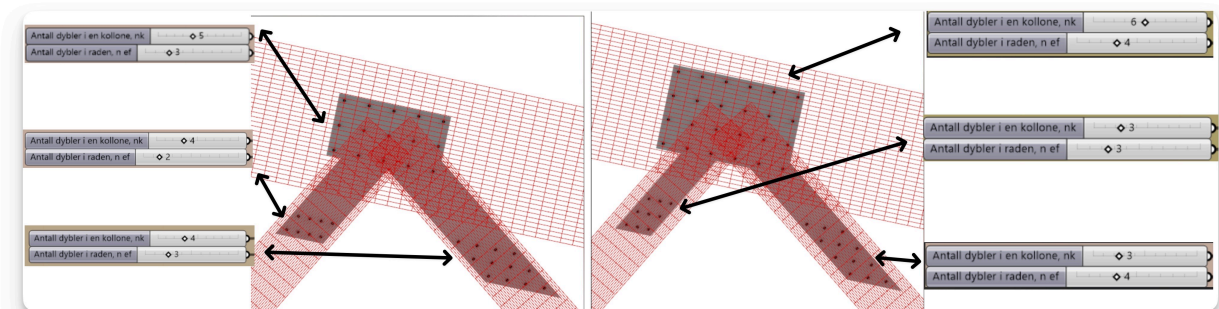
Figur 7.6: Endring av tverrsnitts bredde

I Figur 7.7 vises at ved å endre verdien på parameteren som bestemmer avstanden mellom stålplatene, oppdateres modellen og stålplatene blir omplassert i elementene. Denne parameteren bestemmer avstandene mellom stålplatene for alle elementene.



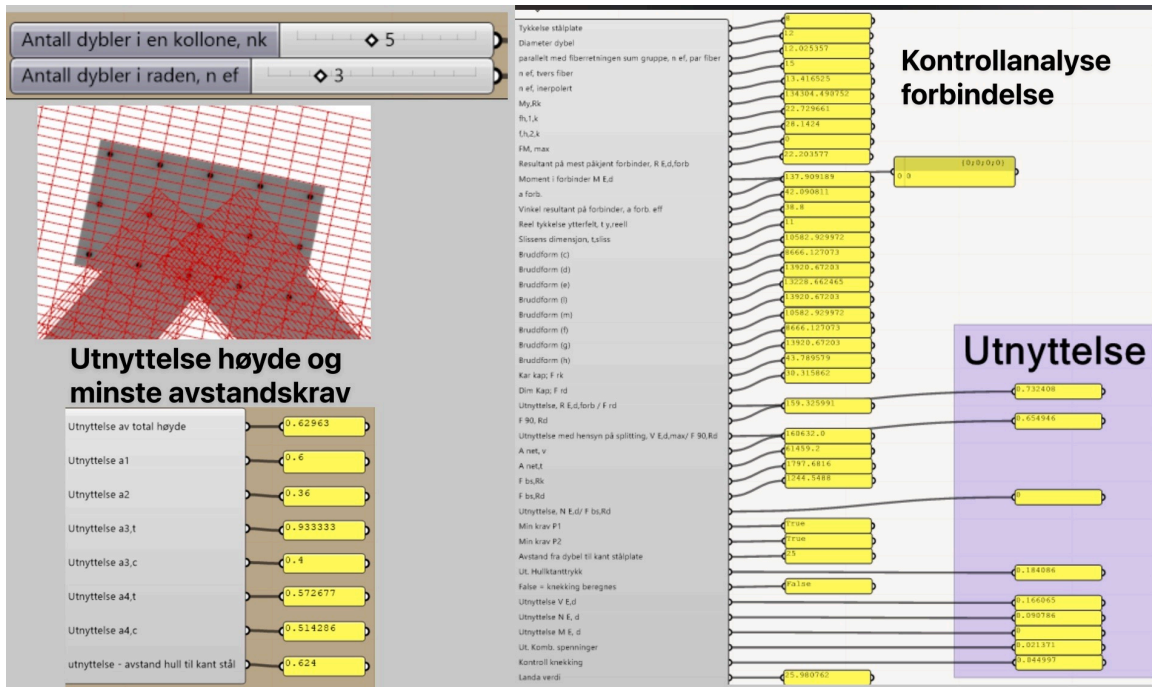
Figur 7.7: Endring av avstand mellom stålplater

Antall dybler bestemmes av to parameter, og stålplaten oppdateres direkte i henhold til dette. I Figur 7.8 vises det at stålplaten oppdateres i alle knutepunktene som er i forbindelsen når verdiene til parameterne endres.

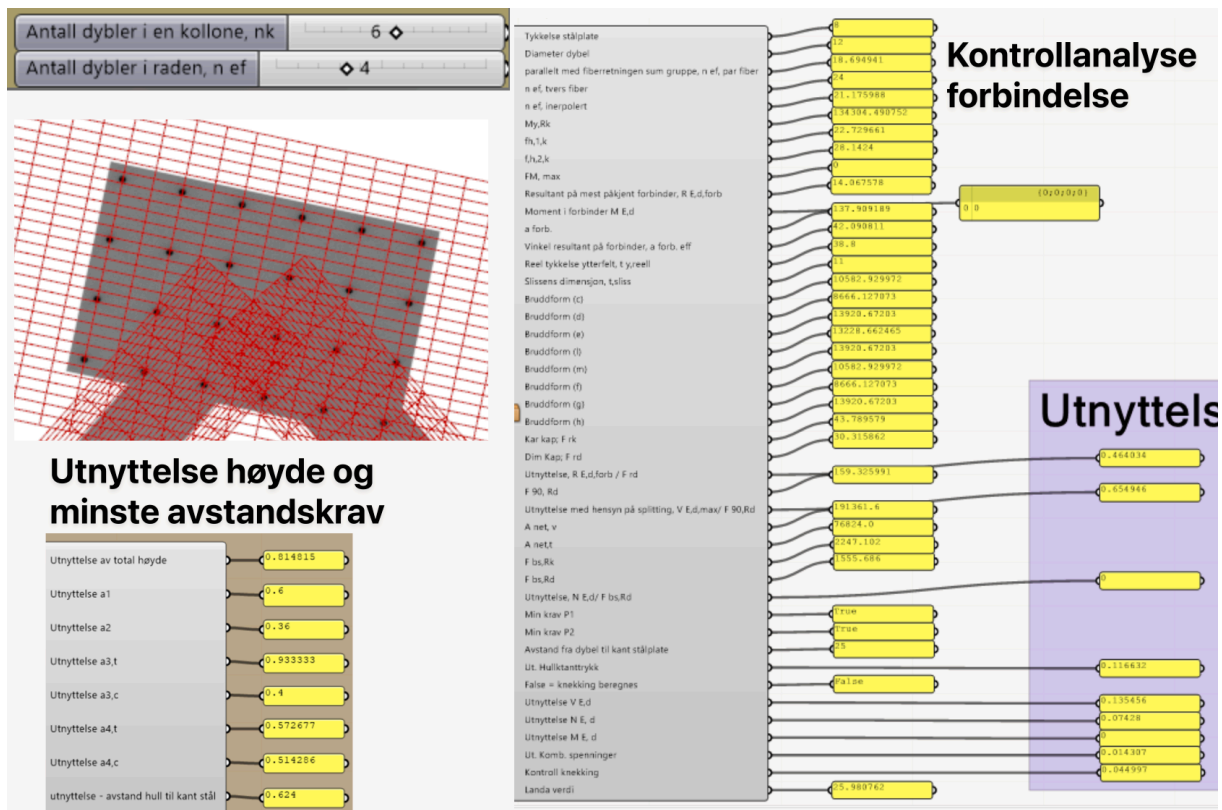


Figur 7.8: Endring av antall dybler

Kontrollanalysen til knutepunktene vil også oppdateres direkte, når antall dybler endres. Dette er vist i to figurer: Figur 7.9 og Figur 7.10. I disse figurene blir det vist at ved å endre antallet av dybler i knutepunktene, vil kontrollanalysen direkte oppdateres i henhold til dette. I tillegg vises også utnyttelsen av høyden til tverrsnittet og utnyttelsen til avstandskravene som er gitt i EK5 og EK3 [63, 115]. Dette gjør at man kan utforske flere varianter av antallet til dyblene og finne en optimal løsning både for utnyttelse og mengden av stålet i forbindelsen.



Figur 7.9: Oppdatering av kontrollen for knutepunktet i Overgurt



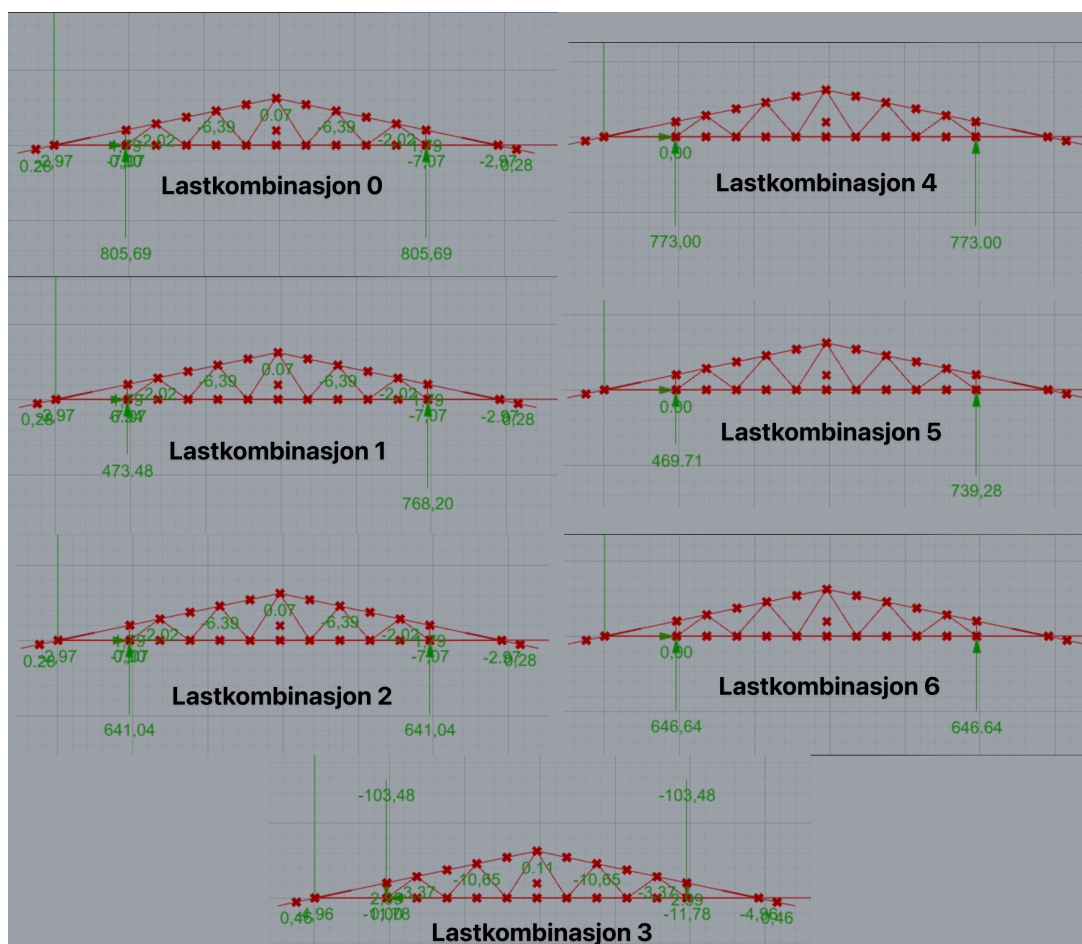
Figur 7.10: Oppdatering av kontrollen for knutepunktet i Overgurt

7.2 Største belastningskrefter på hvert element

I dette kapitlet sammenlignes belastningskreftene som er generert av Karamba3D med de tilsvarende belastningskreftene fra Robot Structural Analysis modellen til Sweco. Dette skal gi en indikasjon på påliteligheten til Karamba3D, ettersom lastene og fagverksmodellen som er programmert i Grasshopper er identisk med modellen til Sweco fra Robot. Videre i dette kapittel viser tabeller sammenligningen av kreftene for hvert element med de forskjellige lastkombinasjonene. Dette skal gi en oversikt over differansen mellom kreftene fra Karamba3D og Robot.

7.2.1 Oppleggs-krefter for alle lastkombinasjonene fra Karamba3D

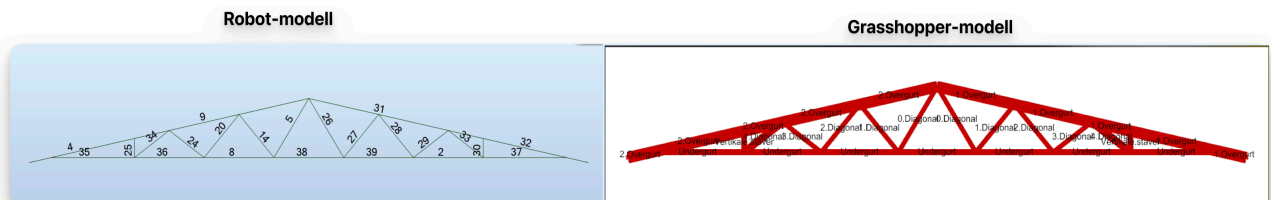
I Figur 7.11 vises resultatet for opplagerkreftene for alle lastkombinasjonene som er programmer i Grasshopper og generert av Karamba3D. I lastkombinasjonene 0, 1, 2 og 3 ser man vindlasten som er påført i y-retningen på fagverket. Lastkombinasjon 0 og 4 har jevnt fordelte laster hvor last 0 har med vindlast, mens last 4 har uten. I lastkombinasjon 1 og 5 er det skjevt fordelt snølast på Overgurt 1 (høyre overgurt i Figur 7.11), der last 1 har med vindlast, mens 5 er uten. Lastkombinasjon 3 er jevnt fordelt vindlast for oppløft. Lastkombinasjon 2 og 6 er snølasten sentrert mellom oppleggene, der 2 er med vindlast og 6 er uten vindlast.



Figur 7.11: Oppleggs-krefter for hver lastkombinasjon

7.2.2 Overgurt

I Tabell 7.1 presenteres sammenligningen av belastningskreftene som er påført overgurtene. I Grasshopper-modellen er overgurtene delt opp i hver forbindelse og består dermed totalt av ti elementer, altså fem elementer for hver overgurt. Utnyttelseskontrollen som analyserer bruddgrensetilstanden til elementene, bruker det lengste spennet av de fem elementene med de største belastningskreftene som er påført overgurtene. I tabellen presenteres de største skjær-, moment- (M_y og M_z), trykk- og strekkreftene som er påført på hver av de ti elementene som representerer overgurtene i Grasshopper. I Robot-modellen til Sweco er hver overgurt oppdelt i to elementer, og belastningskreftene som er presentert i tabellen, er de største belastningskreftene som er påført disse fire elementene som utgjør begge overgrutene. I Figur 7.12 vises element navnene fra Robot- og Grasshopper-modellen. Element 4, 9, 31 og 32 representerer overgurtene fra Robot, mens overgurt 1 og 2 representerer overgurtene i Grasshopper-modellen.



Figur 7.12: Element navn fra Robot og Grasshopper

Resultatet fra Tabell 7.1 viser at det er en relativt minimal forskjell mellom belastningskreftene fra Karamba3D og Robot innen skjærbelastningen på overgurtene. Robot har litt høyere skjærkraft innen seks av de sju lastkombinasjonene. Den største differansen er i lastkombinasjon 6, der skjærkraften fra Karamba3D er ca. 4% høyere enn skjærkraften fra Robot.

For momentbelastningen i y-retningen er momentkreftene fra Robot-modellen større innenfor alle lastkombinasjonene. Den største differansen er i lastkombinasjon 1, der momentkraften er 3,6% større i Robot-modellen. For moment i z-retningen er Karamba3D derimot større innen fire av de totalt sju lastkombinasjonene. Der den største differansen kommer fra lastkombinasjon 3, som er oppløft fra vindlasten.

I trykkraften viser resultatet at Karamba3D har større trykkrefter i seks av sju lastkombinasjoner, men flere av dem er relativt like. De største differansene er i lastkombinasjon 1 og 5, som er laster med skjev snølast med og uten vindlast. Her er differansen over 30 KN som utfør en forskjell på 7,6% for lastkombinasjon 1 og 6,8% forskjell i lastkombinasjon 5.

For strekkraften er Robot-modellen størst i fire av de sju lastkombinasjonene, der den største differansen er igjen i lastkombinasjon 1 og 5. Robot-modellen har 8,94 KN og 10 KN større strekkraft i lastkombinasjon 1 og 5, men siden disse kreftene er så store utgjør dette en forskjell på 1,23% i lastkombinasjon 1 og 1,4% i lastkombinasjon 5.

Tabell 7.1: Sammenligning av krefter overgurt

OVERGURT	SKJÆR (VZ) (KN)		MOMENT (MY) (KNm)		MOMENT (MZ) (KN)		TRYKK (KN)		STREKK (KN)	
	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
<i>Last. komb.</i>										
0	151,23	151,89	206,98	211	4,06	1,24	570,20	569,16	712	708,7
1	150,73	154,32	205	212,71	4,06	1,24	458,04	425,56	714	722,9
2	102,69	99,3	122,46	125,83	4,06	2,07	681,65	685,9	367	367,2
3	22,23	22,38	29,25	29,87	6,76	0	95,04	94,69	83	90,6
4	144,69	145,38	198,31	202,16	0	0	545,41	544,4	684	680,8
5	144,33	147,89	196,54	204,09	0	0	444,53	414,25	685	695
6	103,07	99,3	123,45	125,83	0	0	686,79	685,90	370	368
Differanse	0,66		4,02		2,82		1,04		3,33	
	3,59		7,71		2,82		32,48		8,94	
	3,39		3,37		1,99		4,25		0,23	
	0,15		0,62		6,76		0,35		7,57	
	0,69		3,85		0		1,01		3,25	
	3,56		7,55		0		30,28		10	
	3,77		2,38		0		0,89		2,02	

7.2.3 Undergurt

I Tabell 7.2 sammenlignes belastningskreftene som er påført undergruten. I Grasshopper-modellen er også undergruten oppdelt i hver forbindelse. Dette gjør at det er totalt sju elementer som representerer hele undergruten i fagverket. Resultatet fra denne tabellen viser dermed de største skjær-, moment- (My og Mz), trykk- og strekkreftene som er påført hver av de sju elementene i undergruten, slik som i overgurt. I Robot modellen er også undergruten oppdelt i sju elementer, og kreftene som presenteres i tabellen er de største kreftene på hver av dem. I Figur 7.12 er element 35, 36, 8, 38, 39, 2 og 37 undergruten, mens elementene med navn «Undergurt» er undergruten i Grasshopper.

Resultatet fra Tabell 7.2, viser at det er relativ liten forskjell mellom skjærkreftene fra Karamba3D og Robot-modellen. Den største differansen i lastkombinasjon 2, som er sentrert snølast og utgjør en forskjell på 5%, siden kreftene er så lave.

Momentkraften i y-retningen er størst i fem av de totalt sju lastkombinasjonene, der den største differansen er i lastkombinasjon 1 og 5. Denne forskjellen er på 1,66% og 1,73% for lastkombinasjon 1 og 5. Resultatet for momentkraften i z-retningen viser det samme som for overgurt. Karamba3D har momentkrefter i fire av sju lastkombinasjoner. Den største differansen er igjen lastkombinasjon 3, der differansen er på 6,63 KN og er en forskjell på 83% når kreftene er så små. Dette var også tilfellet for overgurt.

Trykkreftene fra Robot i undergruten er størst innen fire av sju lastkombinasjoner. Den største differansen er for lastkombinasjon 1 og 5, der trykkraften fra Robot er 8 KN større enn trykkraften fra Karamba3D. Ettersom disse kreftene er så store, er forskjellen i lastkombinasjon 1 på 1,22%, og 1,3% i lastkombinasjon 5.

For strekkreftene er også forskjellen relativ minimal. Robot har større strekkrefter i fire av sju lastkombinasjoner, men enkelte strekkrefter kan betraktes som like. De største differansene er i lastkombinasjon 1 og 5, der strekkreftene fra Robot er 1,23% og 1,33% enn strekkreftene fra lastkombinasjon 1 og 5 fra Karamba3D.

Tabell 7.2: Sammenligning av krefter undergrut

UNDERGRUT	SKJÆR (VZ) (KN)		MOMENT (MY) (KNm)		MOMENT (MZ) (KN)		TRYKK (KN)		STREKK (KN)	
	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
<i>Last. komb.</i>										
0	6,40	6,39	16,72	16,70	4,78	0,81	667,31	664,26	550	548,61
1	6,23	6,32	16,52	16,80	4,78	0,81	669,44	677,7	453	458,64
2	4,54	4,78	9,89	10,16	4,78	0,81	341,41	342,53	688	692,66
3	1,24	1,23	1,60	1,80	7,97	1,34	51,45	51,67	98	98,09
4	6,17	6,17	16,05	16,04	0	0	638,62	635,68	530	529,24
5	6,02	6,10	15,88	16,16	0	0	640,54	648,96	443	448,98
6	4,79	4,78	10,17	10,16	0	0	344,60	342,53	693	692,53
Differanse		0,01		0,02		3,97		3,05		1,39
		0,09		0,28		3,97		8,26		5,64
		0,24		0,27		3,97		1,12		4,66
		0,01		0,2		6,63		0,22		0,09
		0		0,01		0		2,94		0,76
		0,08		0,28		0		8,42		5,98
		0,01		0,01		0		2,07		0,47

7.2.4 Vertikale staver

Resultatet fra Tabell 7.3 viser at skjær- og momentkreftene i y-retningen er identiske i både Karamba3D og Robot. Momentkreftene i z-retningen er derimot forskjellige, og Karamba3D er størst i fire av sju lastkombinasjoner. Kraftene er relativt små, men basert på de tidligere resultatene har Karamba3D alltid større momentkrefter i z-retningen.

I trykkreftene viser resultatet at Robot har større trykkrefter innen alle lastkombinasjonene. Den største differansen er i lastkombinasjon 1 og 5, der trykkraften er ca. 9,5 KN større i begge lasttilfellene. Dette er en forskjell på over 3,3% for begge lastene. For strekk, er resultatene nesten identiske, der Robot er neste 1 KN større enn kraften fra Karamba3D i lastkombinasjon 3.

Tabell 7.3 viser sammenligningen mellom belastningskreftene fra Karamba3D og Robot-modellen til Sweco for de vertikale stavene. Resultatet som presenteres i denne tabellen er de største skjær-, moment- (My og Mz), trykk- og strekkreftene som er påført de to vertikale stavene i limtre-fagverket fra Karamba3D og Robot. I Robot-modellen er de vertikale stavene element 25 og 30, og i Grasshopper er de vertikale stavene navngitt som «vertikale staver», se Figur 7.12.

Resultatet fra Tabell 7.3 viser at skjær- og momentkreftene i y-retningen er identiske i både Karamba3D og Robot. Momentkreftene i z-retningen er derimot forskjellige, og Karamba3D er størst i fire av sju lastkombinasjoner. Kraftene er relativt små, men basert på de tidligere resultatene har Karamba3D alltid større momentkrefter i z-retningen.

I trykkreftene viser resultatet at Robot har større trykkrefter innen alle lastkombinasjonene. Den største differansen er i lastkombinasjon 1 og 5, der trykkraften er ca. 9,5 KN større i begge lasttilfellene. Dette er en forskjell på over 3,3% for begge lastene. For strekk, er resultatene nesten identiske, der Robot er neste 1 KN større enn kraften fra Karamba3D i lastkombinasjon 3.

Tabell 7.3: Sammenligning av krefter vertikale staver

VERTIKAL STAV	SKJÆR (VZ) (KN)		MOMENT (MY) (KNm)		MOMENT (MZ) (KN)		TRYKK (KN)		STREKK (KN)	
	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
Last. komb.										
0	0	0	0	0	1,44	0	288,85	293,39	0	0
1	0	0	0	0	1,44	0	285,97	295,80	0	0
2	0	0	0	0	1,44	0	183,43	187,55	0	0
3	0	0	0	0	2,4	0	0	0	41	41,91
4	0	0	0	0	0	0	276,67	281,03	0	0
5	0	0	0	0	0	0	274	283,73	0	0
6	0	0	0	0	0	0	184,83	187,64	0	0
Differanse					1,44		4,54		0	
					1,44		9,83		0	
					1,44		4,12		0	
					2,4		0		0,91	
					0		5,03		0	
					0		9,73		0	
					0		2,81		0	

7.2.5 Diagonal 0

Tabell 7.4 viser resultatet for sammenligningen av belastningskrefter i diagonal 0 fra Karamba3D og Robot. I Grasshopper er det to staver representerer diagonal 0, og resultatet som presenteres i tabellen, representerer de største belastningskreftene som påføres disse stavene. Dette er også tilfellet for Robot-modellen. I Figur 7.12 representerer elementene 5 og 26 fra Robot-modellen diagonal 0, mens diagonal 0 er navngitt «0.Diagonal» i Grasshopper-modellen.

Resultatet for diagonal 0 i Tabell 7.4, viser at nesten identiske for skjær- og momentkreftene i y-retningen. Momentkreftene i z-retningen er derimot forskjellige i fire av de sju lastkombinasjonene, som er samme som de tidligere resultatene. I trykkraften resultatene identiske i fem av de sju lasttilfellene, og forskjellen er relativ minimal i de to ulike lastkombinasjonene. I strekkreftene er derimot større forskjeller. I seks av sju lasttilfellet er strekkreftene større i Robot, der de største forskjellene er i lastkombinasjon 1 og 5. Her er forskjellen på 2,6% for lastkombinasjon 1 og 3% for lastkombinasjon 5.

Tabell 7.4: Sammenligning av krefter Diagonal 0

DIAGONAL 0	SKJÆR (VZ) (KN)		MOMENT (MY) (KNm)		MOMENT (MZ) (KN)		TRYKK (KN)		STREKK (KN)		
	Last. komb.	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
0		0,19	0,19	0	0	0,84	0	0	0	31	31,23
1		0,19	0,19	0	0	0,84	0	0,72	0	49	50,3
2		0,14	0,19	0	0	0,84	0	0	0	62	62,66
3		0,14	0,14	0	0	1,40	0	6,25	6,33	0	0
4		0,19	0,19	0	0	0	0	0	0	29	29,36
5		0,19	0,19	0	0	0	0	0	0	45	46,40
6		0,19	0,19	0	0	0	0	0	0	62	62,66
Differanse		0,05	0			0,84		0		0,23	
				0,84		0,72		1,3			
				0,84		0		0,66			
				1,4		0,08		0			
				0		0		0,36			
				0		0		1,4			
				0		0		0,66			

7.2.6 Diagonal 1

Tabell 7.5 viser sammenligningen mellom skjær-, moment- (My og Mz), trykk- og strekkreftene fra Karamba3D og Robot-modellen til Sweco. Resultatet som presenteres i denne tabellen er de maksimale belastningskreftene som påføres staverne som defineres som diagonal 1. Både Grasshopper- og Robot- modellen er det to staver som defineres som diagonal 1. I Figur 7.12 er element 14 og 27 fra Robot-modellen diagonal 1, mens «1.Diagonal» er diagonal 1 i Grasshopper modellen.

Resultatene fra tabell 7.5 er relativt likt for diagonal 1. Skjær- og momentkreftene i y-retningen er nesten identiske. Moment-kreftene er igjen forskjellige for de fire første lastkombinasjonene, der Karamba3D er har størst krefter. For strekkreftene har Karamba3D gitt 5 KN og 3 KN fra den skjeve snølasten som er i lastkombinasjon 1 og 5, selv om dette er en trykkstav.

For trykkreftene er Robot større i fem av de sju last tilfellene. De største differansene kommer fra den skjeve snølasten, som er lastkombinasjon 1 og 5. Her der det en differanse gjennomsnittlig differanse på 1,5 KN, som utgjør en forskjell på 3,5% mellom Robot og Kramba3D.

Tabell 7.5: Sammenligning av krefter Diagonal 1

DIAGONAL 1	Skjær (Vz) (KN)		Moment (My) (KNm)		Moment (Mz) (KN)		Trykk (KN)		Strekk (KN)	
	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
0	0,19	0,19	0	0	0,84	0	31	30,48	0	0
1	0,19	0,19	0	0	0,84	0	50,1	51,49	5	0
2	0,14	0,19	0	0	0,84	0	65,38	65,76	0	0
3	0,14	0,14	0	0	1,4	0	0	0	9	8,89
4	0,19	0,19	0	0	0	0	28,13	28,45	0	0
5	0,19	0,19	0	0	0	0	46,08	47,74	3	0
6	0,19	0,19	0	0	0	0	65,49	65,76	0	0
Differanse	0		0		0,84		0,52		0	
					0,84		1,39		5	
					0,84		0,38		0	
					1,4		0		0,11	
					0		0,32		0	
					0		1,66		3	
				0		0,27		0		

7.2.7 Diagonal 2

I Tabell 7.6 sammenlignes belastningskreftene for diagonal 2. Resultatene som presenteres er stavene 20 og 28 fra Robot-modellen, og stavene «2.Diaongal» fra Grasshopper-modellen fra Figur 7.12. Resultatet fra Tabell 7.6 viser relativt minimal forskjell på skjærkreftene i diagonalstaven. Momentkraften i y-retningen i identiske, men Kramba3D har større krefter i z-retningen i de fire første lastkombinasjonene. For trykkreftene er det igjen en forskjell i lastkombinasjon 1 og 5, der Robot 2% større trykkrefter i diagonal 2. Strekkreftene er nesten identiske.

Tabell 7.6: Sammenligning av krefter Diagonal 2

DIAGONAL 2	SKJÆR (VZ) (KN)		MOMENT (MY) (KNm)		MOMENT (MZ) (KN)		TRYKK (KN)		STREKK (KN)	
	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
0	0,31	0	0	0	2,03	0	305,84	308,24	0	0
1	0,31	0,31	0	0	2,03	0	278,17	283,95	0	0
2	0,23	0,31	0	0	2,03	0	240,33	243,56	0	0
3	0,23	0,23	0	0	3,38	0	0	0	39	39,41
4	0,31	0,31	0	0	0	0	293,84	296,14	0	0
5	0,31	0,31	0	0	0	0	268,96	274,68	0	0
6	0,31	0,31	0	0	0	0	241,92	243,56	0	0
Differanse	0,31				2,03		2,4		0	
	0				2,03		5,78		0	
	0,08				2,03		3,23		0	
	0		0		3,38		0		0,41	
	0				0		2,3		0	
	0				0		5,72		0	
0				0		1,64		0		

7.2.8 Diagonal 3

Sammenligningen av de største skjær-, moment- (M_y og M_z), trykk- og strekkreftene som er påført diagonal 3 er vist i Tabell 7.7. Denne diagonalen er stav 24 og 29 i Robot-modellen, og «3.Diagonal» i Grasshopper-modellen vist i Figur 7.12.

Resultatet viser skjærkreftene er nesten identiske. Momentkreftene i y-retningen identisk, men igjen har Karamba3D større momentkrefter i z-retningen for de fire første lastkombinasjonene. Trykkreftene har en liten forskjell på 1,2% i lastkombinasjon 3, der Robot er størst. For strekkreftene er den største differansen i lastkombinasjon 1 og 5. Her er Robot 2% større for begge disse last tilfellene.

Tabell 7.7: Sammenligning av krefter Diagonal 3

DIAGONAL 3	SKJÆR (VZ) (KN)		MOMENT (MY) (KNm)		MOMENT (MZ) (KN)		TRYKK (KN)		STREKK (KN)	
	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
Last. komb.										
0	0,31	0,31	0	0	2,97	0	0	0	386	388,6
1	0,31	0,31	0	0	2,97	0	0	0	351	358,8
2	0,23	0,31	0	0	2,97	0	0	0	304	307,9
3	0,23	0,23	0	0	4,95	0	45,67	46,18	0	0
4	0,31	0,31	0	0	0	0	0	0	371	373,5
5	0,31	0,31	0	0	0	0	0	0	340	347,2
6	0,31	0,31	0	0	0	0	0	0	307	308,4
		0				2,97		0		2,6
		0				2,97		0		7,8
		0,08				2,97		0		3,9
Differanse		0		0		4,95		0,51		0
		0				0		0		2,5
		0				0		0		7,2
		0				0		0		1,4

7.2.9 Diagonal 4

Sammenligningen av de største skjær-, moment- (M_y og M_z), trykk- og strekkreftene som er påført diagonalstavene 4 er vist i Tabell 7.8. I Robot-modellen er diagonal 4 definert som stav 33 og 34, og «4.Diaognal» er definer som diagonal 4 i Grasshopper-modellen i Figur 7.12.

Resultatet viser relativ små forskjeller mellom skjærkreftene fra Kramba3D og Robot-modellen. Momentkreftene i y-retningen er identiske, men i z-retningen er det forskjeller i de fire første lastkombinasjonene. Når det kommer til trykkreftene, har Karamba3D større trykkrefter i fem av sju lastkombinasjoner. De største differansene er i lastkombinasjon 0 og 4, som er jevnt fordelt snølast, med og uten vindlast. Her har Karamba3D en større trykkraft på ca. 8 KN i begge lasttilfellene. Dette utgjør en forskjell på 1%, siden kreftene er så store. I strekk er det kun en forskjell, som er i lastkombinasjon 3. Her der Karamba3D 1 KN større enn strekkraften fra Robot.

Tabell 7.8: Sammenligning av krefter Diagonal 4

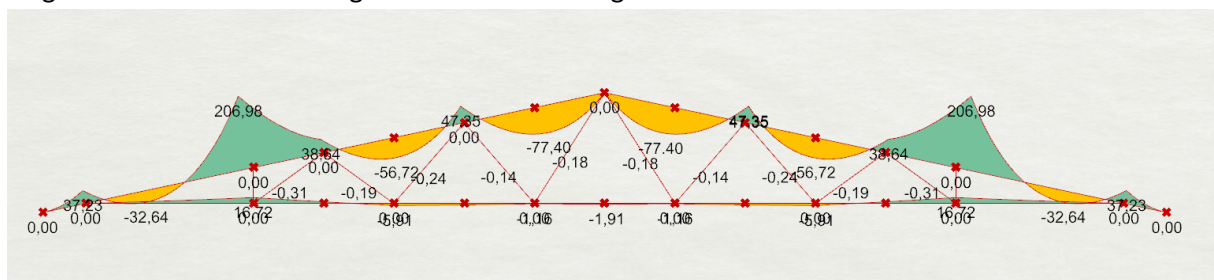
DIAGONAL 4	Skjær (Vz) (KN)		Moment (My) (KNm)		Moment (Mz) (KN)		Trykk (KN)		Strekk (KN)		
	Last. komb.	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot	Karamba	Robot
0		0,50	0,49	0	0	3,89	0	865,74	856,8	0	0
1		0,50	0,49	0	0	3,89	0	806,85	811,6	0	0
2		0,37	0,49	0	0	3,89	0	770,52	770,5	0	0
3		0,37	0,37	0	0	6,48	0	0	0	108	107
4		0,50	0,49	0	0	0	0	831,23	822,38	0	0
5		0,50	0,49	0	0	0	0	778,26	783,34	0	0
6		0,50	0,49	0	0	0	0	776,63	770,5	0	0
Differanse		0,01				3,89		8,94		0	
		0,01				3,89		4,75		0	
		0,12				3,89		0,02		0	
		0		0		6,48		0		1	
		0,01				0		8,85		0	
		0,01				0		5,08		0	
		0,01				0		6,13		0	

7.3 Resultater moment-, skjær-, og aksialkraftdiagram

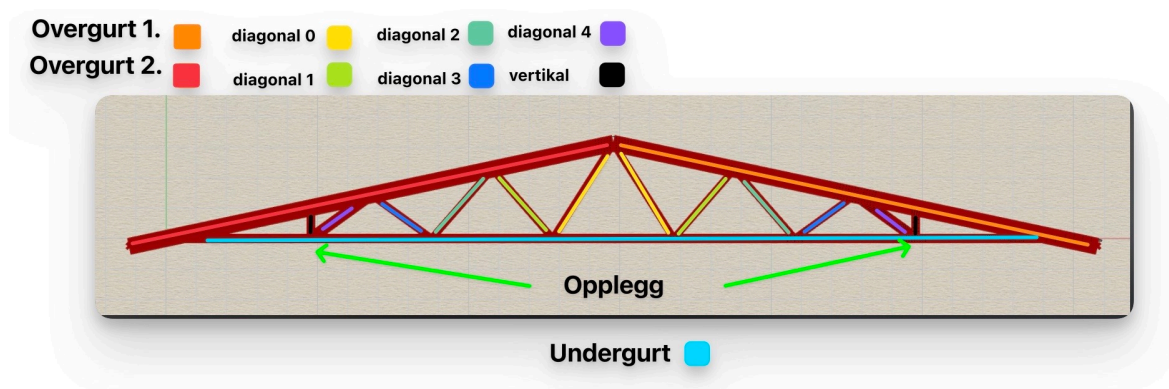
I dette underkapittelet presenteres moment-, skjær-, og aksialkraftdiagrammet fra karamba3D analysen i Grasshopper-modellen. Diagrammene er i henhold til lastkombinasjon 0, som også brukes diagrammene til Sweco, vist i kapittel 5.1.5. Karamba3D ble justert for å inkludere et momentfrie ledd i toppunktet og i hver ende av alle stavnene i limtre-fagverket. I tillegg ble det lagt inn momentstive ledd i forbindelsene som kobler sammen under- og overgurtene.

7.3.1 Momentkraftdiagram

I Figur 7.13 vises momentdiagrammet for limtre-fagverket som er visualisert i Rhnio3D.



Figur 7.13: Moment- diagram (My) fra Karamba3D



Figur 7.16: Limtrefagverk med oversikt over elementer

Videre i dette underkapittelet presenteres maksimal utnyttelse for skjær-, strekk-, trykk-, bøyekrefter og kombinasjonen av dem i henhold til EK5. Det er valgt å presentere utnyttelsene fra alle elementene limtre-fagverket og sammenligne dem med resultatet til Sweco. I beregningsrapporten til Sweco presenteres kun de største utnyttelsene fra kontrollen, derfor er ikke alle feltene i tabellen utfylt. I tillegg har Sweco undersøkt hvert element som utgjør overgurtene, og har ikke gjort forenklingen som er i Grasshopper. Dette gjør at enkelte sammenligner ikke reelt sett, stemmer med hverandre. Det valgt å inkludere alle kontrollene som er utført i Grasshopper modellen, både for å gi en oversikt over resultatene til hvert element og for å vise mulighetene til den programmerte komponenten i Grasshopper.

7.4.1 Overgrut

I Tabell 7.9 oppsummeres dimensjoner og lengden til overgruten fra Grasshopper modellen og fra bergeningssrapporten til Sweco. Lengden som er oppgitt, er den lengste lengden til den oppdelte overgruten i Grasshopper. Sweco gitt lengden har gitt den horisontale lengden til elementet, mens Grasshopper gir den reelle lengden. I Grasshopper-modellen er det utført kontroll for alle lastkombinasjonen, og utnyttelsen som vises er den høyeste fra lastkombinasjonene. Tabell 7.10 oppsummerer de maksimale utnyttelse innen; bøyning, kombinasjon bøyning, skjær, trykk, strekk, kombinasjon bøyning og strekk, kombinasjon bøyning og trykk, og knekking for både sterk og svak akse. Siden resultatet til begge overgrutene er like, representerer resultatet i tabellen begge overgrutene.

Tabell 7.9: Geometri overgrut

GEOMETRI	OVERGRUT FRA GRASSHOPPER MODELL		OVERGURT FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	540	mm	540	mm
Bredde	295	mm	295	mm
Lengde	4883,55	mm	4777	mm

Resultatene som er vist for Sweco i Tabell 7.10, er de maksimale utnyttelsene for elementet som har størst påkjenning av lastene i overgruten. Resultatet i denne tabellen viser at det store forskjeller på utnyttelses verdiene fra Grasshopper modellen og fra beregningsrapporten til Sweco. Spesielt knekking etter formel 6.23, der forskjellen er på over 30%. I Grasshopper er lengden på elementet i

overgurten lengre enn lengden som er gitt i beregningsrapporten til Sweco, noe som vil gi forskjeller i utnyttelsesverdiene.

Tabell 7.10: Kontroll overgurt

	Overgrut fra Grasshopper modell	Overgurt fra Sweco beregningsrapport
KONTROLL I HENHOLD TIL EKS	UTNYTTELSE (%)	UTNYTTELSE (%)
6.1.6 bøyning My	63,12	
6.1.6 bøyning Mz	3,22	
6.1.6 kombinasjon bøyning (6.11)	63,12	
6.1.6 kombinasjon bøyning (6.12)	44,19	
6.1.7 Skjær	62,18	49
6.1.4 trykk i fiberretning	22,95	
6.1.2 strekk i fiberretningen	28,19	
6.2.3 kombinert bøyning og strekk (6.17)	91,24	100
6.2.3 kombinert bøyning og strekk (6.18)	72,31	
6.2.4 kombinert bøyning og trykk (6.19)	43,46	
6.2.4 kombinert bøyning og trykk (6.20)	66,45	
6.3.2 knekking (6.23)	81,77	50
6.3.2 knekking (6.24)	65,49	

7.4.2 Undergurt

I Tabell 7.11 vises dimensjonen og lengden til undergruten både fra Grasshopper modellen og fra beregningsrapporten til Sweco. I Tabell 7.12 vises utnyttelsene for bruddgrensetilstandene både fra Grasshopper og Sweco.

Tabell 7.11: Geometri undergrut

GEOMETRI	UNDERGRUT FRA GRASSHOPPER MODELL		UNDERGRUT FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	540	mm	540	mm
Bredde	295	mm	295	mm
Lengde	4777	mm	4777	mm

Resultatet fra Tabell 7.12 viser at det er relativt store forskjeller i utnyttelseskontrollene, selv om geometrien fra Grasshopper og fra beregningsrapporten er samme. Strekkreftene som ble vist i Tabell 7.2 var litt større i Robot-modellen til Sweco, men det bemerkes at trykk- og strekkraft i utnyttelseskontrollen til Sweco er forstørret.

Tabell 7.12: Kontroll undergurt

KONTROLL I HENHOLD TIL EK5	UNDERGRUT - GRASSHOPPER MODELL	UNDERGRUT - SWECO BEREGNINGSRAPPORT
	UTNYTTELSE (%)	UTNYTTELSE (%)
6.1.6 bøyning My	20,4	
6.1.6 bøyning Mz	7,59	
6.1.6 kombinasjon bøyning (6.11)	20,6	
6.1.6 kombinasjon bøyning (6.12)	16,75	
6.1.7 Skjær	5,30	
6.1.4 trykk i fiberretning	42,82	
6.1.2 strekk i fiberretningen	57,04	
6.2.3 kombinert bøyning og strekk (6.17)	69,99	78
6.2.3 kombinert bøyning og strekk (6.18)	66,11	
6.2.4 kombinert bøyning og trykk (6.19)	38,66	
6.2.4 kombinert bøyning og trykk (6.20)	32,53	
6.3.2 knekking (6.23)	72,73	80
6.3.2 knekking (6.24)	63,56	

7.4.3 Vertikale staver

I Tabell 7.13 presenteres dimensjonene til de vertikale stavene i limtre-fagverket. Resultatene fra kontrollanalysen av disse staven er oppsummert i Tabell 7.14.

Tabell 7.13: Geometri vertikale staver

GEOMETRI	VERTIKAL STAV FRA GRASSHOPPER MODELL		VERTIKAL STAV FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	270	mm	270	mm
Bredde	295	mm	295	mm
Lengde	1014,58	mm	1015	mm

Resultatet fra Tabell 7.14 viser en relativ stor forskjell i utnyttelse av strekk. Strekkraften fra Tabell 7.3 var relativt identiske fra Karamba3D og Robot-modellen. Det bemerkes igjen at i beregningsrapporten til Sweco er det brukt større strekk- og trykkrefter enn av som ble gitt i Robot-modellen, noe som kan forklare forskjellene i utnyttelsene.

Tabell 7.14: Kontroll vertikale staver

KONTROLL I HENHOLD TIL EK5	VERTIKAL STAV FRA GRASSHOPPER MODELL	VERTIKAL STAV - SWECO BEREGNINGSRAPPORT
	UTNYTTELSE (%)	UTNYTTELSE (%)
6.1.6 bøyning My	0	
6.1.6 bøyning Mz	2,29	
6.1.6 kombinasjon bøyning (6.11)	1,16	
6.1.6 kombinasjon bøyning (6.12)	2,28	
6.1.7 Skjær	0	
6.1.4 trykk i fiberretning	18,49	22
6.1.2 strekk i fiberretningen	2,80	4
6.2.3 kombinert bøyning og strekk (6.17)	4,39	
6.2.3 kombinert bøyning og strekk (6.18)	5,08	
6.2.4 kombinert bøyning og trykk (6.20)	3,05	
6.2.4 kombinert bøyning og trykk (6.19)	3,086	
6.3.2 knekking (6.23)	18,30	
6.3.2 knekking (6.24)	18,27	

7.4.4 Diagonal 0

I Tabell 7.15 vises den valgte dimensjonen til Diagonal 0 stavene fra Grasshopper-modellen og fra Sweco. Det bemerkes at Sweco har presisert at denne staven er 180x225 i Figur 5.3 fra beregningsrapporten, men er 215x270 i utnyttelseskontrollen i samme rapport. Det er valgt å bruke 180x225 i Grasshopper modellen, men kontrollen til Sweco er 215x270. Oppsummeringen av resultatet for bruddgrensetilstanden til diagonal 0 er presentert i Tabell 7.16.

Tabell 7.15: Geometri diagonal 0

GEOMETRI	DIAGONAL 0 FRA GRASSHOPPER MODELL		DIAGONAL 0 FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	225	mm	270	mm
Bredde	180	mm	215	mm
Lengde	3722	mm	3772	mm

Resultatet vist i Tabell 7.16 viser relativ minimal forskjell på utnyttelsesverdiene. Det bemerkes likevel at utnyttelsene fra Grasshopper er større, selv om det er brukt litt større strekk og trykkrefter i rapporten til Sweco. Disse kreftene er større enn kreftene fra Karamba3D, og i tillegg er det brukt større tverrsnitt på disse stavene, noe som kan forklare forskjellen i utnyttelsesverdiene.

Tabell 7.16: Kontroll diagonal 0

KONTROLL I HENHOLD TIL EKS	DIAGONAL 0 - GRASSHOPPER MODELL	DIAGONAL 0 - SWECO BEREGNINGSRAPPORT
	UTNYTTELSE (%)	UTNYTTELSE (%)
6.1.6 bøyning My	0	
6.1.6 bøyning Mz	4,30	
6.1.6 kombinasjon bøyning (6.11)	3,30	
6.1.6 kombinasjon bøyning (6.12)	4,30	
6.1.7 Skjær	0,26	
6.1.4 trykk i fiberretning	0,67	
6.1.2 strekk i fiberretningen	10,10	7
6.2.3 kombinert bøyning og strekk (6.17)	10,10	
6.2.3 kombinert bøyning og strekk (6.18)	10,74	
6.2.4 kombinert bøyning og trykk (6.19)	3,01	
6.2.4 kombinert bøyning og trykk (6.20)	4,31	
6.3.2 knekking (6.23)	3,80	
6.3.2 knekking (6.24)	4,60	1

7.4.5 Diagonal 1

I Tabell 7.17 vises dimensjonen på diagonal 1 fra Grasshopper-modellen og Sweco rapporten. Sweco har kontrollert denne diagonalen med en lengde som ikke er mulig for fagverket, dette betraktes som en skrivefeil.

Tabell 7.17: Geometri diagonal 1

GEOMETRI	DIAGONAL 1 FRA GRASSHOPPER MODELL		DIAGONAL 1 FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	225	mm	225	mm
Bredde	180	mm	180	mm
Lengde	3040	mm	3772	mm

I Tabell 7.18 oppsummeres bruddgrensekontrollen for diagonal 1 fra Grasshopper-modellen og Sweco rapporten. Resultatet viser at det er en relativ stor forskjell på utnyttelsen for strekk i fiberretningen. Dette antas å være en feil, ettersom denne staven er i trykk. Strekkraften fra lastkombinasjon 3 er veldig lav, og vil dermed ikke utgjøre 11%.

Tabell 7.18: Kontroll diagonal 1

KONTROLL I HENHOLD TIL EK5	DIAGONAL 1 - GRASSHOPPER MODELL	DIAGONAL 1 - SWECO BEREGNINGSRAPPORT
	UTNYTTELSE (%)	UTNYTTELSE (%)
6.1.6 bøyning My	0	
6.1.6 bøyning Mz	0	
6.1.6 kombinasjon bøyning (6.11)	0	
6.1.6 kombinasjon bøyning (6.12)	0	
6.1.7 Skjær	0,20	
6.1.4 trykk i fiberretning	8,6	
6.1.2 strekk i fiberretningen	1,17	11
6.2.3 kombinert bøyning og strekk (6.17)	4,34	
6.2.3 kombinert bøyning og strekk (6.18)	4,34	
6.2.4 kombinert bøyning og trykk (6.19)	0,74	
6.2.4 kombinert bøyning og trykk (6.20)	0,74	
6.3.2 knekking (6.23)	9,3	
6.3.2 knekking (6.24)	10,19	1

7.4.6 Diagonal 2

I Tabell 7.19 er vises dimensjonen til diagonal 2 fra Grasshopper og Swecos rapport.

Tabell 7.19: Geometri diagonal 2

GEOMETRI	DIAGONAL 2 FRA GRASSHOPPER MODELL		DIAGONAL 2 FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	225	mm	225	mm
Bredde	295	mm	295	mm
Lengde	3040	mm	3037	mm

I Tabell 7.20 oppsummeres bruddgrensekontrollen for diagonal 2 fra Grasshopper-modellen og Sweco rapporten. Resultatet for strekk i fiberretningen er neste det sammen, ettersom strekkraften fra lastkombinasjon 3 var nesten identiske i Tabell 7.6. Resultatet for knekking derimot har en forskjell på ca. 4%, noen som kommer av forskjellen i trykkraften som ble vist i Tabell 7.6.

Tabell 7.20: Kontroll diagonal 2

	DIAGONAL 2 - GRASSHOPPER MODELL	DIAGONAL 2 - SWECO BEREGNINGSRAPPORT
KONTROLL I HENHOLD TIL EK5	UTNYTTELSE (%)	UTNYTTELSE (%)
6.1.6 bøyning My	0	
6.1.6 bøyning Mz	3,86	
6.1.6 kombinasjon bøyning (6.11)	2,27	
6.1.6 kombinasjon bøyning (6.12)	3,86	
6.1.7 Skjær	0,32	
6.1.4 trykk i fiberretning	23,57	
6.1.2 strekk i fiberretningen	3,15	3
6.2.3 kombinert bøyning og strekk (6.17)	5,85	
6.2.3 kombinert bøyning og strekk (6.18)	7,01	
6.2.4 kombinert bøyning og trykk (6.19)	5,65	
6.2.4 kombinert bøyning og trykk (6.20)	6,35	
6.3.2 knekking (6.23)	25,44	29
6.3.2 knekking (6.24)	24,38	

7.4.7 Diagonal 3

Tabell 7.21 presenterer dimensjon til diagonal 3 fra Grasshopper-modellen og fra rapporten til Sweco.

Tabell 7.21: Geometri diagonal 3

GEOMETRI	DIAGONAL 3 FRA GRASSHOPPER MODELL		DIAGONAL 3 FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	225	mm	225	mm
Bredde	295	mm	295	mm
Lengde	2464	mm	2468	mm

Tabell 7.22 viser resultatet til bruddgrensekontrollen fra Grasshopper-modellen og rapporten til Sweco. Resultatene er like for strekk i fiberretningen, men det er en 5% forskjell på knekking i z-akse.

Tabell 7.22: Kontroll diagonal 3

	DIAGONAL 3 - GRASSHOPPER MODELL	DIAGONAL 3 - SWECO BEREGNINGSRAPPORT
	UTNYTTELSE (%)	UTNYTTELSE (%)
<i>KONTROLL I HENHOLD TIL EK5</i>		
6.1.6 bøyning My	0	
6.1.6 bøyning Mz	5,66	
6.1.6 kombinasjon bøyning (6.11)	3,96	
6.1.6 kombinasjon bøyning (6.12)	5,66	
6.1.7 Skjær	0,32	
6.1.4 trykk i fiberretning	0,30	
6.1.2 strekk i fiberretningen	36,66	37
6.2.3 kombinert bøyning og strekk (6.17)	36,66	
6.2.3 kombinert bøyning og strekk (6.18)	36,66	
6.2.4 kombinert bøyning og trykk (6.19)	4,05	
6.2.4 kombinert bøyning og trykk (6.20)	5,75	
6.3.2 knekking (6.23)	7,08	
6.3.2 knekking (6.24)	8,71	3

7.4.8 Diagonal 4

Tabell 7.23 viser dimensjonen til diagonal 4 fra Grasshopper-modellen og rapporten til Sweco.

Tabell 7.23: Geometri diagonal 4

GEOMETRI	DIAGONAL 4 FRA GRASSHOPPER MODELL		DIAGONAL 4 FRA SWECO BEREGNINGSRAPPORT	
	VERDI	ENHET	VERDI	ENHET
Høyde	360	mm	360	mm
Bredde	295	mm	295	mm
Lengde	2464	mm	2468	mm

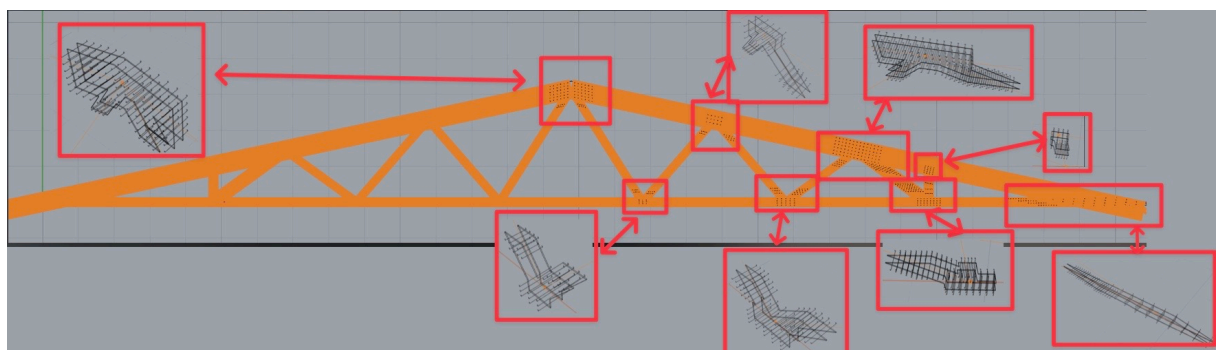
Tabell 7.24 oppsummerer bruddgrensekontrollen fra Grasshopper og rapporten til Sweco. Resultatet for strekk i fiberretningen er relativt likt, men det er en forskjell på 6% for knekking i y-aksen. I Tabell 7.8 vise det at Kramba3D har størst trykkraft, og denne er større enn kraften som er brukt i rapporten til Sweco. Utnyttelsen som er beregnet i Grasshopper-modellen skulle dermed vært størst.

Tabell 7.24: Kontroll diagonal 4

	DIAGONAL 4 - GRASSHOPPER MODELL	DIAGONAL 4 - SWECO BEREGNINGSRAPPORT
KONTROLL I HENHOLD TIL EK5	UTNYTTELSE (%)	UTNYTTELSE (%)
6.1.6 bøyning My	0	
6.1.6 bøyning Mz	0	
6.1.6 kombinasjon bøyning (6.11)	0	
6.1.6 kombinasjon bøyning (6.12)	0	
6.1.7 Skjær	0,32	
6.1.4 trykk i fiberretning	41,67	
6.1.2 strekk i fiberretningen	5,45	7
6.2.3 kombinert bøyning og strekk (6.17)	8,49	
6.2.3 kombinert bøyning og strekk (6.18)	9,79	
6.2.4 kombinert bøyning og trykk (6.19)	17,36	
6.2.4 kombinert bøyning og trykk (6.20)	17,36	
6.3.2 knekking (6.23)	41,97	48
6.3.2 knekking (6.24)	42,41	

7.5 Resultater for forbindelsene

I dette underkapittelet presenteres forbindelsene som knytter sammen limtre-fagverket. I Figur 7.17 vises visualiseringen av knutepunktene i limtre-fagverket. Det er valgt å kun vise for halve fagverket på grunn av at Grasshopper blir veldig treg ved å visualisere alle dybelene og stålplatene i hver forbindelse. Ettersom forbindelsene er symmetriske på begge sidene av fagverket, representerer resultatene begge sidene.



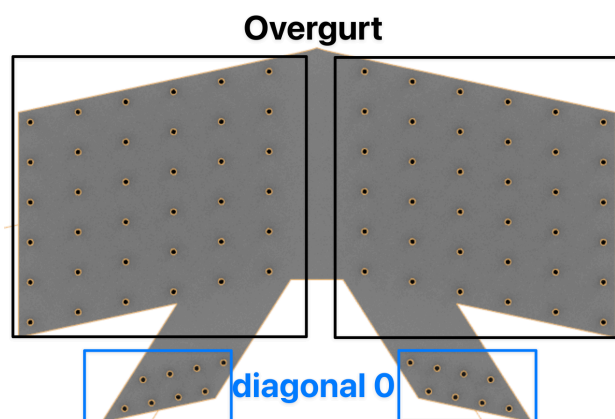
Figur 7.17: Illustrasjon av stålplater og dybler programmert i Grasshopper

Videre fremstilles løsningene for hvert knutepunkt som er vist i Figur 7.17. Løsningene inneholder; belastningskraftene, antall festere, valgte avstander, tykkelse ytterfelt, innerfelt og stålplate, utnyttelse for forbinde, skjær-blokk brudd, og splitting. Alle resultatene fra beregninger og verdier fra Grasshopper-filen er gitt i vedlegg 5. Resultatet til Sweco ble tildelt og refereres til [116].

7.5.1 Forbindelse toppunkt

I Figur 7.18 vises visualiseringen fra Rhino3D av forbindelsen som er i toppunktet til limtrefagverket. Resultatet som er oppsummert i Tabell 7.25 representerer begge overgurts knutepunktene som er markert i Figur 7.18. Etersom knutepunktene i begge endene på stavene er like, representerer resultatet i

Tabell 7.26 knutepunktene som er i diagonal 0 stavene, og er markert både i Figur 7.18 og Figur 7.25. I overgurten er det et eksentrisitetsmoment på grunn av at stivhetssentere til dybel-gruppen ikke er i krysningspunktet der alle elementer møtes.



Figur 7.18: Visualisering av forbindelse i toppunkt fra Rhino3D

Resultatet fra Tabell 7.25 viser at resultatet fra Grasshopper-modellen og rapporten til Sweco er relativt det samme. Det er noen forskjeller på moment-, normal- og skjærkreftene, men disse er minimale og i tillegg blir resultatkraften på den mest påkjente forbinder det samme. Den største forskjellen er utnyttelsen til forbinderen. Her er utnyttelsen fra Grasshopper-modellen 11,6% større enn Sweco sin utnyttelse. Dette kommer av den effektive tykkelsen til ytterdelen som brukes i Grasshopper modellen. Denne er ca. 44 mm mindre enn det Sweco har brukt.

Tabell 7.25: Kontroll overgurt i toppunkt

KNOTEPUNKT	OVERGURT	TOPPUNKT	GRASSHOPPER MODELL		SWECO	
			VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv		29,2	KNm	30	KNm
$N_{E,d}$	Strekk positiv		-647	KN	-646	KN
$V_{E,d}$	Nedover positiv		83,5	KN	86	KN
	Skjær for splittingsberegning		83,5	KN	80	KN
	Endeavstand		100	mm	100	mm
	Kantavstand (både øvre og nedre)		70	mm	70	mm
	Innbyrdes avstand mellom radene		80	mm	80	mm
	Innbyrdes avstand		100	mm	100	mm
	Antall forbindere (dybler)		36	stk	36	stk
	Resultantkraft gruppe		652	KN	652	KN
	Resultantkraft forbinder (mest påkjent forbinder)		29,5	KN	29,5	KN
	Stålplate tykkelse		8	mm	8	mm
	Tykkelse innerfelt		97	mm	97	mm
	Tykkelsen ytterfelt		38,8	mm	83	mm
	Kontroll forbinder (utnyttelse)		85,6	%	74	%
	Kontroll skjærblokk brudd (utnyttelse)		0	%	0	%
	Kontroll splitting (utnyttelse)		52,4	%	50	%

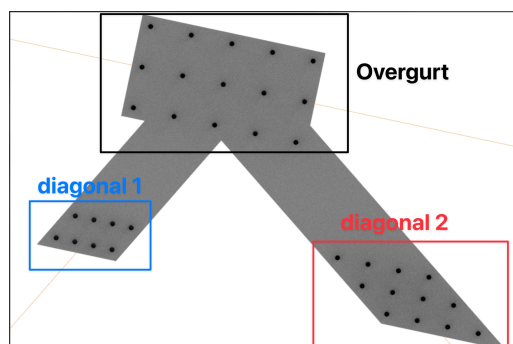
Resultatet fra Tabell 7.26 viser at resultatet fra Grasshopper-modellen er relativt likt med resultatet fra rapporten til Sweco. Den største forskjellen er normalkraften som er større enn normalkraften fra Karamba3D. I dette tilfellet er det ikke brukt den effektive tykkelsen til ytterfeltet, ettersom dette ytterfeltet ikke blir definert som tykt i Grasshopper-modellen. Dette ble beskrevet i formel 6.7 fra kapittel 6.4.9.

Tabell 7.26: Kontroll knutepunkter i diagonal 0

KNUTEPUNKT	DIAGONAL 0	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$ Medurs = positiv		0	KNm	0	KNm
$N_{E,d}$ Strekk positiv		62,45	KN	70	KN
$V_{E,d}$ Nedover positiv		0,20	KN	0	KN
Skjær for splittingsberegning		0,20	KN	0	KN
Endeavstand		130	mm	130	mm
Kantavstand (både øvre og nedre)		52,5	mm	52,5	mm
Innbyrdes avstand mellom radene		40	mm	40	mm
Innbyrdes avstand		70	mm	70 <td mm	
Antall forbindere (dybler)		8	stk	8	stk
Resultantkraft gruppe		62,45	KN	70	KN
Resultantkraft forbinder (mest påkjent forbinder)		10,22	KN	11,5	KN
Stålplate tykkelse		8	mm	8	mm
Tykkelse innerfelt		97	mm	97	mm
Tykkelse ytterfelt		25,5	mm	26	mm
Kontroll forbinder (utnyttelse)		30,6	%	33	%
Kontroll skjærblokk brudd (utnyttelse)		24,63	%	28	%
Kontroll splitting (utnyttelse)		0,45	%	0	%

7.5.2 Forbindelse overgurt, diagonal 1 og 2

I Figur 7.19 vises forbindelsen som sammenkobler overgurten og diagonalstavene 1 og 2. I Tabell 7.27 oppsummeres resultatet fra Grasshopper modellen og beregningsrapporten til Sweco for overgurten i denne forbindelsen. Tabell 7.28 viser resultatet av knutepunktene som er i hver ende av diagonal 1, mens Tabell 7.29 viser resultatet til knutepunktene i diagonal 2. I Figur 7.19 og Figur 7.25 vises de visualiserte knutepunktene som er i diagonal 1, mens Figur 7.19 og Figur 7.26 viser knutepunktene i diagonal 2.



Figur 7.19: Forbindelse overgurt, diagonal 1 og 2 visualisert i Rhino3D

Den største forskjellen mellom resultatet fra Grasshopper-modellen og rapporten fra Sweco er normalkraften. Tabell 7.27 viser at det er en forskjell på 79 KN på normalkraften, noe som gjør at resultantkraften på dybel-gruppen og mest påkjente forbinder har store forskjeller.

Det bemerkes også at det er forskjeller på tykkelsen på innerfeltet i knutepunktene som er i diagonal 1. Dette kommer av at Grasshopper modellen bruker samme innertykkelse i alle forbindelser, og det ble ikke lagt merke til at alle andre forbindelser i rapporten til Sweco hadde en innertykkelse på 105 mm, før resultatene ble sammenlignet i denne oppgaven.

Tabell 7.27: Kontroll overgurt i forbindelsen med diagonal 1 og 2

KNUTEPUNKT	OVERGURT	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	-221	KN	-300	KN
$V_{E,d}$	Nedover positiv	199,7	KN	201	KN
	Skjær for splittingsberegning	104	KN	104	KN
	Endeavstand	90	mm	90	mm
	Kantavstand (både øvre og nedre)	70	mm	70	mm
	Innbyrdes avstand mellom radene	100	mm	100	mm
	Innbyrdes avstand	100	mm	100	mm
	Antall forbindere (dybler)	15	stk	15	stk
	Resultantkraft gruppe	297,89	KN	361	KN
	Resultantkraft forbinder (mest påkjent forbinder)	22,2	KN	28,3	KN
	Stålplate tykkelse	8	mm	8	mm
	Tykkelse innerfelt	97	mm	105	mm
	Tykkelse ytterfelt	38,8	mm	79	mm
	Kontroll forbinder (utnyttelse)	73,24	%	76	%
	Kontroll skjærblokk brudd (utnyttelse)	0	%	0	%
	Kontroll splitting (utnyttelse)	65,5	%	65	%

Resultatet som er oppsummert i Tabell 7.28 er relativt like, ettersom det kun er minimal forskjell på normalkraften og ytterfeltet.

Tabell 7.28: Kontroll diagonal 1

KNUTEPUNKT	DIAGONAL 1	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	-65	KN	-70	KN
$V_{E,d}$	Nedover positiv	0,19	KN	0	KN
	Skjær for splittingsberegning	0,19	KN	0	KN
	Endeavstand	130	mm	130	mm
	Kantavstand (både øvre og nedre)	52,5	mm	52,5	mm
	Innbyrdes avstand mellom radene	40	mm	40	mm
	Innbyrdes avstand	70	mm	70	mm
	Antall forbindere (dybler)	8	stk	8	stk
	Resultantkraft gruppe	65	KN	70	KN
	Resultantkraft forbinder (mest påkjent forbinder)	10,64	KN	11,5	KN
	Stålplate tykkelse	8	mm	8	mm
	Tykkelse innerfelt	97	mm	105	mm
	Tykkelse ytterfelt	25,5	mm	26	mm
	Kontroll forbinder (utnyttelse)	31,8	%	33	%
	Kontroll skjærblokk brudd (utnyttelse)	28	%	28	%
	Kontroll splitting (utnyttelse)	0,456	%	0	%

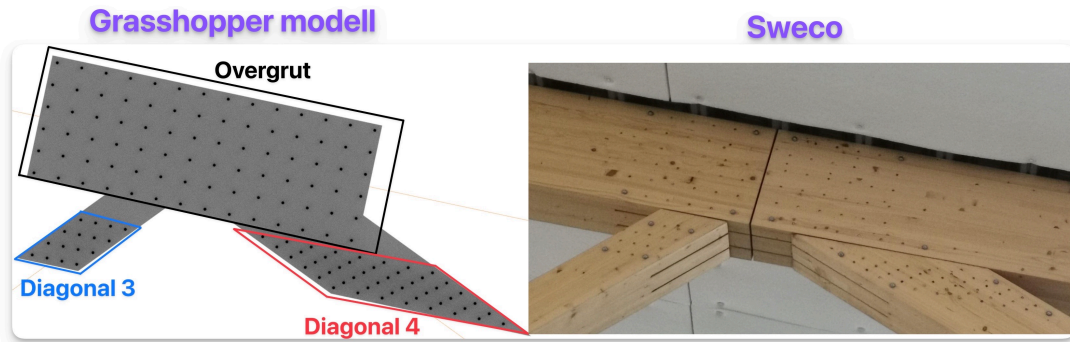
Sweco har gitt forskjellige endeavstander i knutepunktene som er i diagonal 2 (Tabell 7.29). Dette er ikke utført i knutepunktene som er i Grasshopper-modellen, der har begge knutepunktene samme endeavstand på 135 mm. I Sweco sine resultater som er presentert i Tabell 7.29, er knutepunktet med endeavstand på 135 mm i forbindelsen tilkoblet med overgruten, mens knutepunktet i forbindelsen med undergruten har 120 mm i endeavstanden. Resultatet er relativt likt, selv som det er forskjeller i endeavstandene. Det er en forskjell i utnyttelsen til forbinder, som er på 10%, men igjen kommer dette av den effektive tykkelsen til ytterfeltet.

Tabell 7.29: Kontroll diagonal 2

KNUTEPUNKT	DIAGONAL 2	GRASSHOPPER		SWECO					
		MODELL		VERDI	ENHET	VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	-293	KN	-296	KN	-296	KN	-296	KN
$V_{E,d}$	Nedover positiv	0,311	KN	0	KN	0	KN	0	KN
	Skjær for splittingsberegning	0,311	KN	0	KN	0	KN	0	KN
	Endeavstand	135	mm	135	mm	120	mm	120	mm
	Kantavstand (både øvre og nedre)	45	mm	45	mm	45	mm	45	mm
	Innbyrdes avstand mellom radene	45	mm	45	mm	45	mm	45	mm
	Innbyrdes avstand	90	mm	90	mm	90	mm	90	mm
	Antall forbindere (dybler)	12	stk	12	stk	12	stk	12	stk
	Resultantkraft gruppe	293	KN	296	KN	296	KN	296	KN
	Resultantkraft forbinder (mest påkjent forbinder)	31,27	KN	31,6	KN	31,6	KN	31,6	KN
	Stålplate tykkelse	8	mm	8	mm	8	mm	8	mm
	Tykkelse innerfelt	97	mm	105	mm	105	mm	105	mm
	Tykkelse ytterfelt	38,8	mm	79	mm	79	mm	79	mm
	Kontroll forbinder (utnyttelse)	89	%	79	%	79	%	79	%
	Kontroll skjærblokk brudd (utnyttelse)	0	%	0	%	0	%	0	%
	Kontroll splitting (utnyttelse)	0,39	%	0	%	0	%	0	%

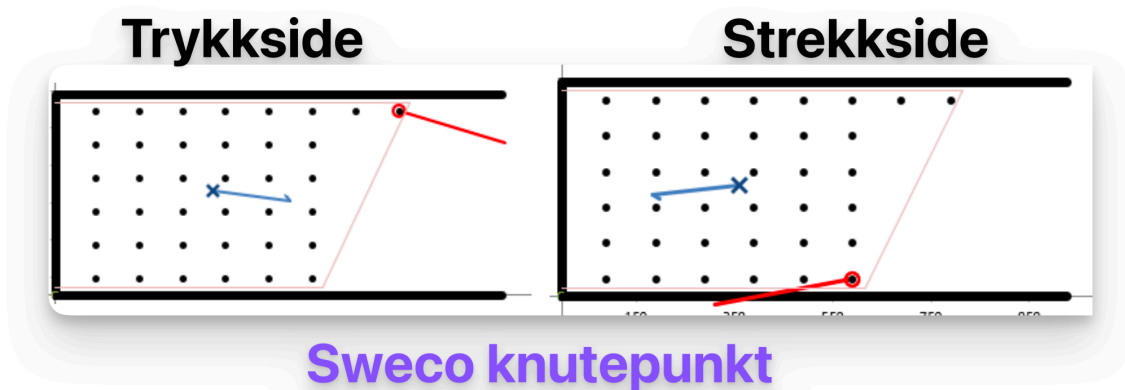
7.5.3 Forbindelse overgrut, diagonal 3 og 4

Forbindelsen som kobler sammen overgurt med diagonal 3 og 4, er ikke utformet på samme måte i Grasshopper som Sweco har i limtre-fagverket. I Figur 7.20 vises den visuelt programmerte forbindelsen fra Grasshopper-modellen, og hvordan forbindelsen ble prosjektet av Sweco. Sweco har skjøtet overgurtene i denne forbindelsen, noe som ikke er utført i Grasshopper modellen. Resultatet som er oppsummert i Tabell 7.30, viser knutepunktet i overgruten fra Grasshopper og begge knutepunktene i den skjøtete overgurtene til Sweco. Det første knutepunktet til Sweco er strekksiden, mens det andre er i trykksiden i overgurtene.



Figur 7.20: Forbindelse fra Grasshopper og Sweco

Det er også forskjell i antall dybler i forbindelsen. Sweco har 8 dybler i den nederste raden, som er vist øverst i Figur 7.21. Forbindelsen i Grasshopper har samme antall dybler i hver rad. Det bemerkes at det er en skrivefeil i beregningsrapporten til Sweco, siden det ikke er 8 dybler i noen av radene til knutepunktet i overgurtene til venstre (trykkside) i Sweco sin forbindelse i Figur 7.20.



Figur 7.21: Sweco overgurt forbindelse

Siden Sweco har skjøtet forbindelsen, oppstår det et eksentrisitetsmoment i denne forbindelsen på grunn av at stivhetssenteret til dybel-gruppen ikke er i krysningspunktet til elementene. Dette vil ikke oppstå i forbindelsen som er programmert i Grasshopper, siden her er stivhetssenteret til dybel-gruppen på linje med krysningspunktet til alle elementene i forbindelsen.

Siden Sweco har skjøtet overgurtene er resultatet som er vist i Tabell 7.30 forskjellig. Det er brukt samme avstander mellom dyblene, men det er et større antall dybler i Grasshopper-modellen enn summen av begge knutepunktene i overgurtene til Sweco. Dette antallet kunne enkelt blitt redusert i Grasshopper-modellen, ettersom den største utnyttelsen er 73%.

Tabell 7.30: Kontroll overgurt forbindelse med diagonal 3 og 4

KNOTEPUNKT	OVERGRUT	GRASSHOPPER MODELL		SWECO			
		VERDI	ENHET	VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	40	KNm	33	KNm
$N_{E,d}$	Strekk positiv	1009,9	KN	641	KN	-561	KN
$V_{E,d}$	Nedover positiv	58,96	KN	90	KN	90	KN
Skjær for splittingsberegning		91,25	KN	90	KN	90	KN
Endeavstand		90	mm	90	mm	90	mm
Kantavstand (både øvre og nedre)		45	mm	45	mm	45	mm
Innbyrdes avstand mellom radene		90	mm	90	mm	90	mm
Innbyrdes avstand		100	mm	100	mm	100	mm
Antall forbindere (dybler)		84	stk	38	stk	38	stk
Resultantkraft gruppe		1011,7	KN	647	KN	568	KN
Resultantkraft forbinder (mest påkjent forbinder)		17,23	KN	28,8	KN	25,9	KN
Stålplate tykkelse		8	mm	8	mm	8	mm
Tykkelse innerfelt		97	mm	105	mm	105	mm
Tykkelse ytterfelt		38,8	mm	79	mm	79	mm
Kontroll forbinder (utnyttelse)		49	%	73	%	67	%
Kontroll skjærblokk brudd (utnyttelse)		73,25	%	31	%	0	%
Kontroll splitting (utnyttelse)		44,7	%	44	%	44	%

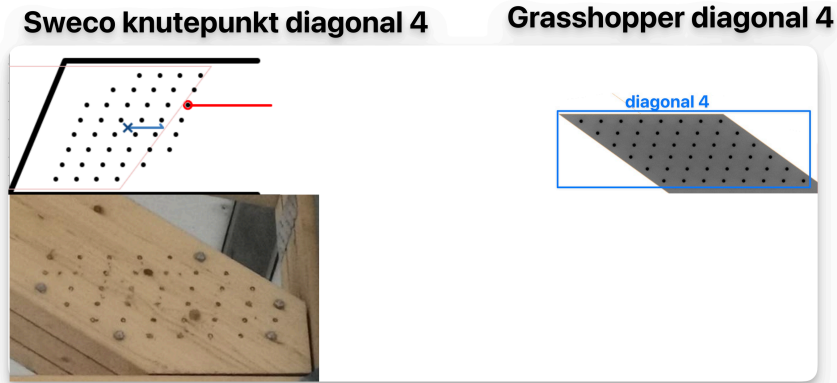
I Tabell 7.31 er det oppsummerte resultatet for hvert knutepunkt i diagonal 3, og i Figur 7.20 vises dette knutepunktet i forbindelsen med overgurten, mens i Figur 7.26 vises dette knutepunktet i forbindelsen med undergurten. Sweco har utformet knutepunktene i diagonal 3 med forskjellige endeavstander. I Tabell 7.31 presenteres begge disse knutepunktene som Sweco har i diagonal 3.

Selv om Sweco har forskjellige endeavstander på knutepunktene i diagonal 3, er alle andre avstander og antallet av dybler likt med knutepunktene i samme stav i Grasshopper-modellen. Resultatet fra Tabell 7.31 viser derimot at utnyttelsen for skjærblokkbrudd overstiger 100% i Grasshopper modellen. Dette kommer av forskjellene mellom størrelsen på inner- og ytterfeltet som brukes i Grasshopper-modellen. Dersom innerfeltet i Grasshopper-modellen økes til 105 mm, blir utnyttelsen 98%, som vil si at forbindelsen er under 100%, selv om den effektive tykkelsen på ytterfeltene brukes.

Tabell 7.31: Kontroll knutepunkter i diagonal 3

KNUTEPUNKT	DIAGONAL 3	GRASSHOPPER MODELL		SWECO			
		VERDI	ENHET	VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	371,4	KN	373	KN	373	KN
$V_{E,d}$	Nedover positiv	0,311	KN	0	KN	0	KN
	Skjær for splittingsberegning	0,311	KN	0	KN	0	KN
	Endeavstand	135	mm	120	mm	135	mm
	Kantavstand (både øvre og nedre)	45	mm	45	mm	45	mm
	Innbyrdes avstand mellom radene	45	mm	45	mm	45	mm
	Innbyrdes avstand	80	mm	80	mm	80	mm
	Antall forbindere (dybler)	16	stk	16	stk	16	stk
	Resultantkraft gruppe	371,4	KN	373	KN	373	KN
	Resultantkraft forbinder (mest påkjent forbinder)	31,5	KN	31,6	KN	31,6	KN
	Stålplate tykkelse	8	mm	8	mm	8	mm
	Tykkelse innerfelt	97	mm	105	mm	105	mm
	Tykkelse ytterfelt	38,8	mm	79	mm	79	mm
	Kontroll forbinder (utnyttelse)	89,9	%	79	%	79	%
	Kontroll skjærblokk brudd (utnyttelse)	105,99	%	71	%	71	%
	Kontroll splitting (utnyttelse)	0,39	%	0	%	0	%

Sweco har utformet knutepunktet som er tilkoblet undergurten i diagonal 4 på en annen måte enn det som er visuelt programmert i Grasshopper. I Figur 7.22 vises utformingen av Swecos knutepunktet og utforming på knutepunktet fra Grasshopper-modellen. Sweco har fjernet dybelen som er nærmest den vertikale staven, ettersom enden på diagonalen er kuttet for å tilpasse forbindelsen. I Tabell 7.32 oppsummeres resultatet for knutepunktene i diagonal 4. Resultatet fra Grasshopper modellen representerer begge knutepunktene i diagonal 4, siden de er likt utformet i begge ender av staven. Resultatet til Sweco viser først knutepunktet som er i tilkoblet overgurt, og deretter knutepunktet som er tilkoblet undergurten.



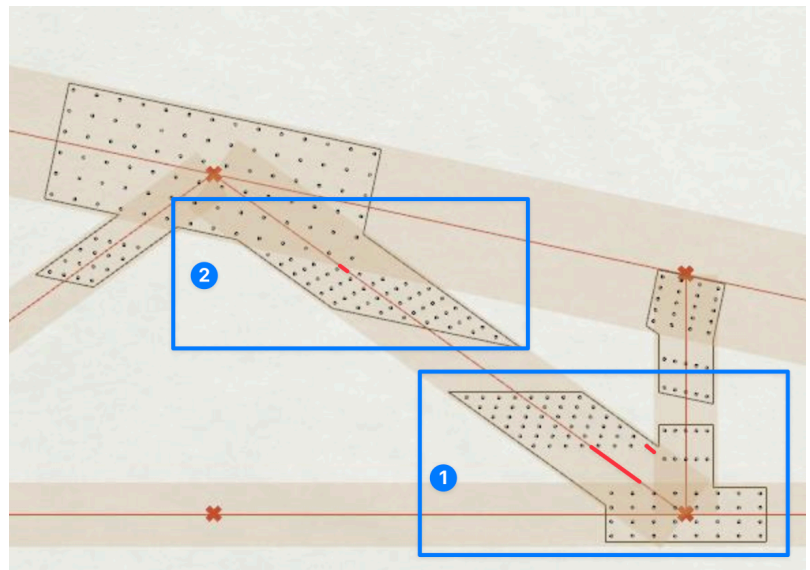
Figur 7.22: Knutepunkt i diagonal 4, koblet til undergurt

Resultatene som er oppsummert i Tabell 7.32 viser veldig store forskjeller for endeavstandene som er brukt i Grasshopper og rapporten til Sweco. I Grasshopper er det valgt å bruke minimumsavstanden for belastet ende, slik at avstanden på stålplatene mellom knutepunktet i diagonal 4 og undergruten ikke blir veldig lang, og at avstanden i fiberretningen mellom knutepunktet i diagonal 4 og den vertikale staven ikke blir for kort. Dette er illustrert i Figur 7.23-1 med røde streker.

Tabell 7.32: Kontroll knutepunkt diagonal 4

KNUTEPUNKT	DIAGONAL 4	GRASSHOPPER MODELL		SWECO			
		VERDI	ENHET	VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	-830	KN	-822	KN	-822	KN
$V_{E,d}$	Nedover positiv	0,497	KN	0	KN	0	KN
	Skjær for splittingsberegning	0,497	KN	0	KN	0	KN
	Endeavstand	84	mm	160	mm	140	mm
	Kantavstand (både øvre og nedre)	40	mm	40	mm	40	mm
	Innbyrdes avstand mellom radene	40	mm	40	mm	40	mm
	Innbyrdes avstand	70	mm	70	mm	70	mm
	Antall forbindere (dybler)	48	stk	40	stk	42	stk
	Resultantkraft gruppe	830	KN	822	KN	822	KN
	Resultantkraft forbinder (mest påkjent forbinder)	25,28	KN	29,5	KN	28,6	KN
	Stålplate tykkelse	8	mm	8	mm	8	mm
	Tykkelse innerfelt	97	mm	105	mm	105	mm
	Tykkelse ytterfelt	38,8	mm	79	mm	79	mm
	Kontroll forbinder (utnyttelse)	72	%	74	%	72	%
	Kontroll skjærblokk brudd (utnyttelse)	0	%	0	%	0	%
	Kontroll splitting (utnyttelse)	0,35	%	0	%	0	%

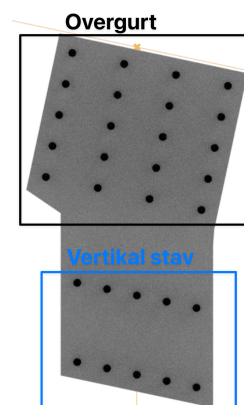
Det oppstår likevel et problem med knutepunktet som er i forbindelsen med overgurten, siden knutepunktene i diagonal 4 bestemmes av samme parametere, blir avstanden på stålplatene mellom knutepunktet til overgurten og diagonal 4 veldig liten. Dette vises i Figur 7.23-2. Avstanden mellom disse er fortsatt innenfor minimumskravet for endeavstandene, men her er det ikke tatt hensyn til at tredelene blir kuttet, for å unngå at tredelene til overgurten og diagonal 4 treffer hverandre. I tillegg ved å øke endeavstanden, minimeres avstandene mellom stålplatene mellom knutepunktene i diagonal 4.



Figur 7.23: Knutepunkter i diagonal 4

7.5.4 Forbindelse overgrut og vertikal stav

I Figur 7.24 vises forbindelsen som er visuelt illustrert i Rhino3D. I denne forbindelsen oppstår det et eksentrisitetsmoment i knutepunktet som er i overgurten. Dette kommer av at stivhetssenteret til dybel-gruppen i overgurten, ikke er på systemlinjene til krysningspunktet til overgurten og den vertikale staven. Tabell 7.33 oppsummeres resultatet for overgurten som har eksentrisitetsmoment, og i Tabell 7.16 er resulterte for knutepunktene i den vertikale staven oppsummert. I Figur 7.23 vises knutepunktet i den vertikale staven som er tilkoblet overgruten, mens Figur 7.27 viser knutepunktet i den vertikale staven som koblet til undergurten.



Figur 7.24: Forbindelse overgurt og vertikal stav

Resultatet som er oppsummert i Tabell 7.33 er relativt like, bortsett eksentrisitetsmomentet i forbindelsen til Sweco er større, dette gjør at resultatkräften på den mest påkjente forbinder er større for Sweco. Det bemerkes at tykkelsen på ytterfeltet i knutepunktet til Sweco er mindre enn der effektive tykkelsen som brukes i Grasshopper, noe som er den største grunnen for at resultatet er nesten det samme.

Tabell 7.33: Kontroll overgurt med vertikal stav

KNOTEPUNKT	OVERGURT	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	7,18	KNm	12,4	KNm
$N_{E,d}$	Strekk positiv	57	KN	50	KN
$V_{E,d}$	Nedover positiv	270	KN	275	KN
	Skjær for splittingsberegning	144,6	KN	145	KN
	Endeavstand	1000	mm	1000	mm
	Kantavstand (både øvre og nedre)	50	mm	50	mm
	Innbyrdes avstand mellom radene	48	mm	48	mm
	Innbyrdes avstand	80	mm	80	mm
	Antall forbindere (dybler)	20	stk	20	stk
	Resultantkraft gruppe	276	KN	280	KN
	Resultantkraft forbinder (mest påkjent forbinder)	18,85	KN	22	KN
	Stålplate tykkelse	8	mm	8	mm
	Tykkelse innerfelt	97	mm	105	mm
	Tykkelse ytterfelt	38,8	mm	34	mm
	Kontroll forbinder (utnyttelse)	73	%	78	%
	Kontroll skjærblokk brudd (utnyttelse)	7,5	%	5	%
	Kontroll splitting (utnyttelse)	75,3	%	75	%

I beregningsrapporten til Sweco er knutepunktene i de vertikale stavene kontrollert for både strekk og trykk, begge disse kontrollene er presentert i Tabell 7.34. Strekkkontrollen kommer fra den oppløftende vindlasten. Dette er ikke kontrollert i Grasshopper-modellen, der er kun presentert kontroll for trykk, siden det vare kontrolleres etter lastkombinasjon 4, 5 og 6.

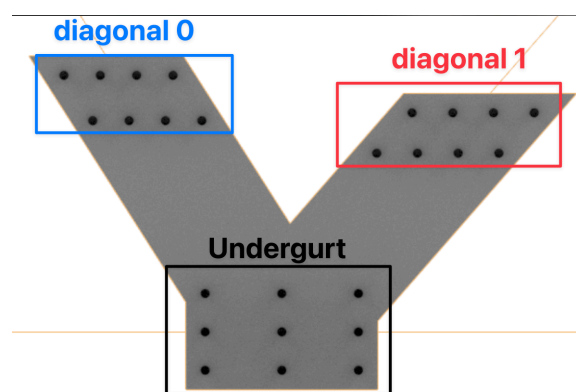
Ved å sammenligne resultatet kraften på mest påkjente forbinder fra Tabell 7.34, er resultatet relativt likt Sweco sitt resultat. Forskjellen er igjen tykkelsen på ytterfeltet, som gjør at utnyttelsesverdien er større i Grasshopper-modellen.

Tabell 7.34: Kontrollknutepunktene i de vertikale stavnene

KNUTEPUNKT	VERTIKAL STAV	GRASSHOPPER MODELL		SWECO			
		VERDI	ENHET	VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	-276,3	KN	-283	KN	50	KN
$V_{E,d}$	Nedover positiv	0	KN	0	KN	0	KN
	Skjær for splittingsberegning	0	KN	0	KN	0	KN
	Endeavstand	100	mm	100	mm	100	mm
	Kantavstand (både øvre og nedre)	45	mm	45	mm	45	mm
	Innbyrdes avstand mellom radene	45	mm	45	mm	45	mm
	Innbyrdes avstand	120	mm	120	mm	120	mm
	Antall forbindere (dybler)	10	stk	10	stk	10	stk
	Resultantkraft gruppe	276,3	KN	283	KN	50	KN
	Resultantkraft forbinder (mest påkjent forbinder)	31,62	KN	32,4	KN	5,7	KN
	Stålplate tykkelse	8	mm	8	mm	8	mm
	Tykkelse innerfelt	97	mm	105	mm	105	mm
	Tykkelse ytterfelt	38,8	mm	79	mm	79	mm
	Kontroll forbinder (utnyttelse)	90	%	81	%	12	%
	Kontroll skjærblokk brudd (utnyttelse)	0	%	0	%	6	%
	Kontroll splitting (utnyttelse)	0	%	0	%	0	%

7.5.5 Forbindelse undergurt diagonal 0 og 1

Figur 7.25 viser visualiseringen av forbindelsen med knutepunktene til undergurten, diagonal 0 og 1 i Rhino3D. Resultatet for knutepunktet til diagonal 0 og 1 ble presentert i Tabell 7.26 og Tabell 7.28. Tabell 7.35 oppsummerer resultatet for kontrollen av undergurten i denne forbindelsen.



Figur 7.25: Forbindelse undergurt, diagonal 0 og 1

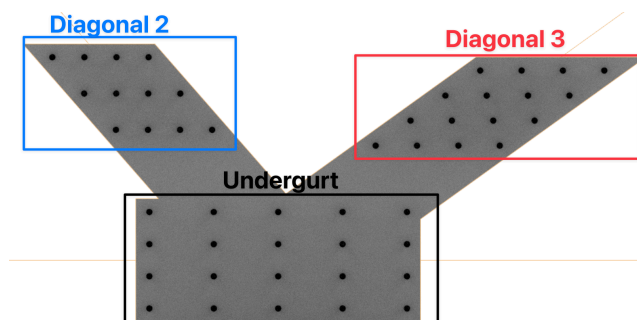
Resultatet fra Tabell 7.35 viser en stor forskjell på normalkraften som brukes i Grasshopper-modellen og rapporten til Sweco. I tillegg er det også en forskjell på skjærkreftene. Dette gjør at resultatkraften på den mest påkjent forbinder er rundt dobbelt så stor i Sweco sin beregning.

Tabell 7.35: Kontroll undergurt med diagonal 0 og 1

KNOTEPUNKT	UNDERGURT	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	76,35	KN	150	KN
$V_{E,d}$	Nedover positiv	2,62	KN	10	KN
Skjær for splittingsberegning		1,875	KN	10	KN
Endeavstand		1000	mm	1000	mm
Kantavstand (både øvre og nedre)		85	mm	85	mm
Innbyrdes avstand mellom radene		50	mm	50	mm
Innbyrdes avstand		100	mm	100	mm
Antall forbindere (dybler)		9	stk	9	stk
Resultantkraft gruppe		76,4	KN	150	KN
Resultantkraft forbinder (mest påkjent forbinder)		10,53	KN	20,8	KN
Stålplate tykkelse		8	mm	8	mm
Tykkelse innerfelt		97	mm	105	mm
Tykkelse ytterfelt		38,8	mm	79	mm
Kontroll forbinder (utnyttelse)		30	%	52	%
Kontroll skjærblokk brudd (utnyttelse)		10,9	%	14	%
Kontroll splitting (utnyttelse)		2,924	%	16	%

7.5.6 Forbindelse undergurt, diagonal 2 og 3

I Figur 7.26 vises forbindelsen som kobler sammen undergurt, diagonal 2 og 3. Resultatet for knutepunktene til diagonal 2 og 3 ble gitt i Tabell 7.29 og Tabell 7.31. I Tabell 7.36 presenteres resultatet for undergruten i denne forbindelsen.



Figur 7.26: Forbindelse som kobler sammen undergurt, diagonal 2 og 3

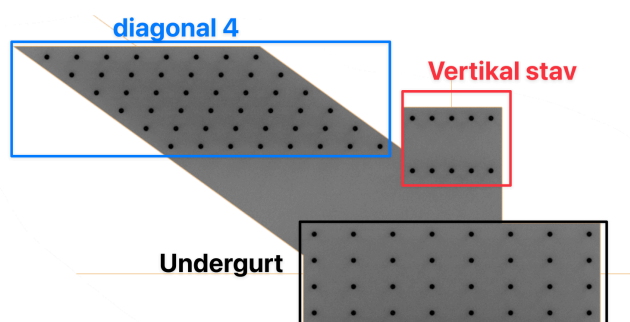
Resultatet som vises i Tabell 7.36 har store forskjeller på skjærblokk brudd og forbinder utnyttelsen. Dette skyldes forskjellene i tykkelsen på både innerfeltet og ytterfeltene som brukes i Grasshoppermodellen, og av skjærkreftene som er større i Sweco sin knutepunkts beregning.

Tabell 7.36: Kontroll undergurt med diagonal 2 og 3

KNOTEPUNKT	UNDERGURT	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	494.4	KN	500	KN
$V_{E,d}$	Nedover positiv	5,07	KN	10	KN
Skjær for splittingsberegning		4,68	KN	10	KN
Endeavstand		1000	mm	1000	mm
Kantavstand (både øvre og nedre)		45	mm	45	mm
Innbyrdes avstand mellom radene		60	mm	60	mm
Innbyrdes avstand		120	mm	120	mm
Antall forbindere (dybler)		20	stk	20	stk
Resultantkraft gruppe		494,4	KN	500	KN
Resultantkraft forbinder (mest påkjent forbinder)		30,95	KN	31,4	KN
Stålplate tykkelse		8	mm	8	mm
Tykkelse innerfelt		97	mm	105	mm
Tykkelse ytterfelt		38,8	mm	79	mm
Kontroll forbinder (utnyttelse)		88,4	%	78	%
Kontroll skjærblokk brudd (utnyttelse)		63,4	%	39	%
Kontroll splitting (utnyttelse)		4,81	%	10	%

7.5.7 Forbindelse undergurt diagonal og vertikal stav

I Figur 7.27 vises forbindelsen fra Rhino3D som knytter sammen knutepunktet i diagonal 4 med den vertikale staven og undergurten i limtrefagverket. Kontrollen for knutepunktet i diagonal 4 og den vertikale staven ble presentert i Tabell 7.32 og Tabell 7.33. I Tabell 7.37 oppsummeres resultatet for undergurten i denne forbindelsen.



Figur 7.27: Forbindelsen som sammenkobler undergrut, diagonal 4 og vertikal stav

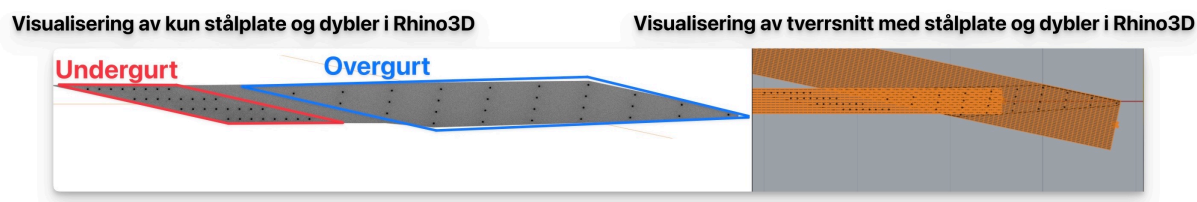
Resultatet fra Tabell 7.37 viser en forskjell på normalkraften i dette knutepunktet. Dersom normalkraften i Grasshopper-modellen hadde vært 700 KN, ville det vært 100% utnyttelse for skjærblokkbrudd. Dette kommer igjen av de reduserte ytterfeltene, og dersom innerfeltet hadde økt til 105 i Grasshopper-modellen ville det vært 91% utnyttelse.

Tabell 7.37: Kontroll undergrut i forbindelse med diagonal 4 og vertikal stav

KNOTEPUNKT	UNDERGURT	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$	Medurs = positiv	0	KNm	0	KNm
$N_{E,d}$	Strekk positiv	674,43	KN	700	KN
$V_{E,d}$	Nedover positiv	10,42	KN	10	KN
Skjær for splittingsberegning		6,17	KN	10	KN
Endeavstand		1000	mm	1000	mm
Kantavstand (både øvre og nedre)		45	mm	45	mm
Innbyrdes avstand mellom radene		60	mm	60	mm
Innbyrdes avstand		90	mm	90	mm
Antall forbindere (dybler)		32	stk	32	stk
Resultantkraft gruppe		674,5	KN	700	KN
Resultantkraft forbinder (mest påkjent forbinder)		29,65	KN	30,9	KN
Stålplate tykkelse		8	mm	8	mm
Tykkelse innerfelt		97	mm	105	mm
Tykkelse ytterfelt		38,8	mm	79	mm
Kontroll forbinder (utnyttelse)		84,67	%	77	%
Kontroll skjærblokk brudd (utnyttelse)		94,45	%	51	%
Kontroll splitting (utnyttelse)		6,34	%	10	%

7.5.8 Forbindelse under- og overgrut

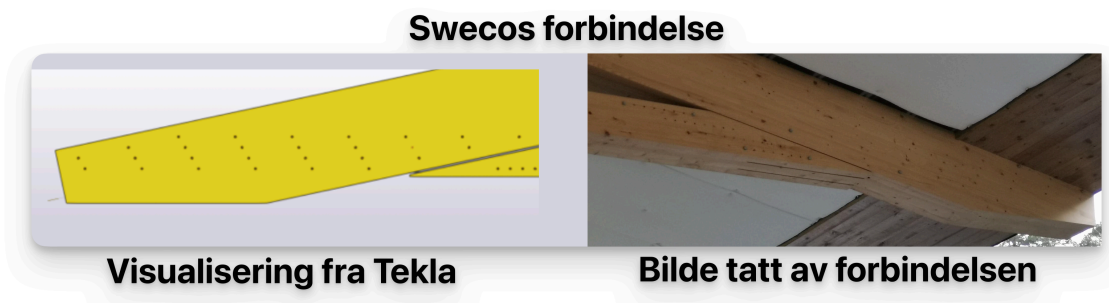
Knutepunktet i overgruten er utformet på en annen måte i Grasshopper, enn det som ble prosjektert av Sweco. Sweco har fjernet en dybel for å tilpasse arkitektens krav om kutting av ende av overgruten [1], dette er vist i Figur 7.29. Figur 7.28 illustrerer denne forbindelsen i Grasshopper-modellen, der tverrsnittet til overgruten ikke er kuttet. I Tabell 7.39 oppsummeres resultater for overgruten i denne forbindelsen. Det en del forskjeller mellom de valgte avstandene i knutepunktet fra Grasshopper modellen og Swecos knutepunkt. Dette er gjort for å få et forholdsvis likt antall dybler i knutepunktet som Sweco har. Avstandene som er valgt er relativt nære minimumskravene fra EK5, og er tilpasset for å få plass til knutepunktet i tverrsnittet til overgruten.



Figur 7.28: Forbindelse som sammenkobler undergrut og overgrut

I Tabell 7.38 presenteres resultatet for undergruten i denne forbindelsen, og Figur 7.28 viser hvordan denne forbindelsen er visualisert i Rhino3D. I beregningsrapporten til Sweco er det skrevet at knutepunktet i undergruten har en stålplate tykkelse på 12 mm, mens overgruten i denne

forbindelsen har en stålplatetykkelse på 8 mm. Det er også en forskjell på tykkelsen på innerfeltene i denne forbindelsen. I overgruten er det skrevet at innerfeltet har en tykkelse på 100 mm, mens undergurten har en tykkelse på 105 mm. Dette antas å være en skrivefeil i rapporten til Sweco, og det antas at stålplaten har en tykkelse på 12 mm og innerfeltet er 100 mm for knutepunktet i over- og undergruten.



Figur 7.29: Swecos prosjekterte forbindelse med over- og undergurt

Resultatet som er vist i Tabell 7.38 er relativt likt med resultatet til Sweco. Det er igjen større utnyttelse på forbindeerne, men siden yterfeltet er redusert i Grasshopper, kan det antas at beregningen som utføres er til sikker side. Basert på alle knutepunktene i undergurten, er det bemerket at Sweco har bruk samme skjærkraft i alle knutepunkts beregningene. Dette er ikke utført i Grasshopper, ettersom skjærkreftene fra det bestemte knutepunktet brukes direkte.

Tabell 7.38: Kontroll knutepunkt undergrut i forbindelse med overgurt

KNUTEPUNKT	UNDERGRUT	VERDI	GRASSHOPPER MODELL		SWECO	
			ENHET	VERDI	ENHET	
$M_{E,d}$ Medurs = positiv		0	KNm	0	KNm	
$N_{E,d}$ Strekk positiv		-640	KN	-650	KN	
$V_{E,d}$ Nedover positiv		2,433	KN	10	KN	
Skjær for splittingsberegning		2,433	KN	10	KN	
Endeavstand		340	mm	340	mm	
Kantavstand (både øvre og nedre)		45	mm	45	mm	
Innbyrdes avstand mellom radene		60	mm	60	mm	
Innbyrdes avstand		70	mm	70	mm	
Antall forbindere (dybler)		32	stk	32	stk	
Resultantkraft gruppe		640,54	KN	650	KN	
Resultantkraft forbinder (mest påkjent forbinder)		30,07	KN	30,6	KN	
Stålplate tykkelse		8	mm	12	mm	
Tykkelse innerfelt		97	mm	105	mm	
Tykkelse ytterfelt		38,8	mm	79	mm	
Kontroll forbinder (utnyttelse)		85,86	%	76	%	
Kontroll skjærblokk brudd (utnyttelse)		0	%	0	%	
Kontroll splitting (utnyttelse)		2,50	%	10	%	

Resultatet fra Tabell 7.39 viser store forskjeller i utnyttelse for skjærblokkbrudd og splitting. Forskjellen i skjærblokkbruddet kommer igjen av forskjellen på ytterfeltene, og innerfeltet.

Tabell 7.39: Kontroll knutepunkt overgrut i forbindelse med undergrut

KNUTEPUNKT	OVERGURT	GRASSHOPPER MODELL		SWECO	
		VERDI	ENHET	VERDI	ENHET
$M_{E,d}$ Medurs = positiv		0	KNm	0	KNm
$N_{E,d}$ Strekk positiv		638	KN	646	KN
$V_{E,d}$ Nedover positiv		135,45	KN	137	KN
Skjær for splittingsberegning		78,44	KN	80	KN
Endeavstand		150	mm	119	mm
Kantavstand (både øvre og nedre)		70	mm	71	mm
Innbyrdes avstand mellom radene		78	mm	67	mm
Innbyrdes avstand		269	mm	288	mm
Antall forbindere (dybler)		28	stk	27	stk
Resultantkraft gruppe		652,38	KN	660	KN
Resultantkraft forbinder (mest påkjent forbinder)		24,5	KN	24,5	KN
Stålplate tykkelse		8	mm	12	mm
Tykkelse innerfelt		97	mm	100	mm
Tykkelse ytterfelt		38,8	mm	78	mm
Kontroll forbinder (utnyttelse)		70,8	%	62	%
Kontroll skjærblokk brudd (utnyttelse)		91	%	31	%
Kontroll splitting (utnyttelse)		49	%	39	%

7.6 Utnyttelse av stålet i forbindelsene

I dette kapitelet presenteres kontrollen for stålplatene i forbindelsene. Denne kontrollen inneholder: hullkantrykk, strekkbrudd, trykkbrudd, skjærbrudd, bøyeburdd, kombinasjon av spenninger og knekking. Det er også gitt en verdi for slankheten, som skal indikere slankheten til stålplaten.

Sweco har ikke inkludert kontroll av stålplaten i beregningsrapporten til limtre-fagverket, men ettersom det er stålplatene som tar opp kreftene i forbindelsene, ble det programmert kontroller for dette i Grasshopper. I tillegg er det viktig å undersøke knekking av stålplatene, ettersom korrigeringer av antall dybler og avstandene mellom dem i forbindelsene, kan øke risikoen for knekking mellom stålplatene.

Det er i denne oppgaven kun utført forenklet kontroller av stålplatene basert på anbefalinger fra Limtreboka. Formelen som brukes er presiserte å være til sikker side [58]. Resultatene som presenteres i dette underkapittelet, er utnyttelsene for alle knutepunktene i Grasshopper-modellen.

7.6.1 Forbindelse i toppunkt

I Tabell 7.40 vises utnyttelsene fra kontrollen av stålplatene i overgurten og diagonal 0. Resultatet viser at det er generell lav utnyttelse av stålplatene, og verdien til slankheten er lav.

Tabell 7.40: Kontroll stålplate toppunkt

KONTROLLER	UTNYTTELSE FOR KNUTEPUNKT I OVERGURT	UTNYTTELSE FOR KNUTEPUNKT I DIAGONAL 0
Hullkantrykk	24,5%	8,47%
Skjærbrudd	7,2%	0,05%
Strekk/Trykkbrudd	26,6%	6,7%
Bøyebrudd	10,7%	0%
Kombinasjon spenninger	37,9%	0,64%
Knekking	13,2%	0
Slankhet	25,98	69,43

7.6.2 Forbindelse overgurt, diagonal 1 og 2

Resultatet som er presentert i Tabell 7.41, viser at det er en generell lav utnyttelse av stålplaten, og for knekkingen. Slankheten til stålplaten i diagonal 2 er relativt høy.

Tabell 7.41: Kontroll stålplate overgurt, diagonal 1 og 2

KONTROLLER	UTNYTTELSE (OVERGURT)	UTNYTTELSE (DIAGONAL 1)	UTNYTTELSE (DIAGONAL 2)
Hullkantrykk	18,41%	8,88%	25,93%
Skjærbrudd	16,6%	0,05%	0,075%
Strekk/Trykkbrudd	9,1%	7,1%	29,27%
Bøyebrudd	0%	0%	0%
Kombinasjon spenninger	74,65%	0,69%	8,8%
Knekking	4,45%	4,3%	24,96%
Slankhet	25,98	84,35	108,7

7.6.3 Forbindelse overgurt, diagonal 3 og 4

Utnyttelsen som er vist i Tabell 7.42 generelt lav innen alle knutepunktene. Det eneste er slankhetsverdien som er relativt stor.

Tabell 7.42: Kontroll stålplate overgurt, diagonal 3 og 4

KONTROLLER	UTNYTTELSE (OVERGURT)	UTNYTTELSE (DIAGONAL 3)	UTNYTTELSE (DIAGONAL 4)
Hullkantrykk	14,28%	26,12%	20,96%
Skjærbrudd	4,47%	0,07%	0,071%
Strekk/Trykkbrudd	37,34%	37,1%	46,55%
Bøyebrudd	0%	0%	0%
Kombinasjon spenninger	33,5%	14,05%	21,8%
Knekking	0	0%	66,19%
Slankhet	25,98	55,1	101,7

7.6.4 Forbindelse overgurt og vertikal stav

Resultatet som er vist i Tabell 7.43, viser at stålplaten til overgurten i denne forbindelsen ikke holder i kombinasjon av flere spenninger. Denne kontrollen er til sikker side, ettersom den ikke tar hensyn til gunstige effekter fra flytingen i stålet [58]. Dette limtre-fagverket er konstruert slik at alle kreftene blir sentret i området med de vertikale stavene. Dette gjør at skjærkraften i dette knutepunktet veldig stor. Ved å øke avstanden mellom dyblene eller antallet av dybler i den vertikale retningen, vil arealet til stålplaten i tverretningen økes, og dermed redusere utnyttelses verdien.

Tabell 7.43: Kontroll stålplate overgurt og vertikal stav

KONTROLLER	UTNYTTELSE FOR KNUTEPUNKT I OVERGURT	UTNYTTELSE FOR KNUTEPUNKT I VERTIKAL STAV
Hullkantrykk	15,6%	26,2%
Skjærbrudd	48,8%	0%
Strekk/Trykkbrudd	4,35%	22,2%
Bøyebrudd	9%	0%
Kombinasjon spenninger	186%	4,93%
Knekking	0%	8,44
Slankhet	20,78	38,99

7.6.5 Knutepunkt undergurt med diagonal 0 og 1

I Tabell 7.44 vises resultatet at det er generelt lav utnyttelse av innenfor alle kontrollene.

Tabell 7.44: Kontroll stålplate undergurt med diagonal 0 og 1

KONTROLLER	UTNYTTELSE FOR KNOTEPUNKT I UNDERGURT
Hullkantrykk	8,73%
Skjærbrudd	0,76%
Strek/Trykkbrudd	9,4%
Bøyebrudd	0%
Kombinasjon spenninger	3,79%
Knekking	0%
Slankhet	25,98

7.6.6 Knutepunkt undergurt med diagonal 2 og 3

I Tabell 7.45 vises det at utnyttelsen innenfor kontrollene er generelt lave.

Tabell 7.45: Kontroll stålplate undergurt med diagonal 2 og 3

KONTROLLER	UTNYTTELSE FOR KNOTEPUNKT I UNDERGURT
Hullkantrykk	25,6%
Skjærbrudd	0,91%
Strek/Trykkbrudd	39,7%
Bøyebrudd	0%
Kombinasjon spenninger	19,45%
Knekking	0%
Slankhet	31,177

7.6.7 Knutepunkt undergurt med diagonal 4 og vertikal stav

I Tabell 7.46 vises resultatet for undergruten i denne forbindelsen. Alle utnyttelsene er generelt lave.

Tabell 7.46: Kontroll stålplate undergurt med diagonal 4 og vertikal stav

KONTROLLER	UTNYTTELSE FOR KNOTEPUNKT I UNDERGURT
Hullkantrykk	24,6%
Skjærbrudd	1,9%
Strek/Trykkbrudd	54,2%
Bøyebrudd	0%
Kombinasjon spenninger	36,9%
Knekking	0%
Slankhet	23,38

7.6.8 Forbindelse over- og undergurt

I Tabell 7.47 vises utnyttelsen for forbindelsen mellom over- og undergurten. Det er en stor utnyttelsesverdi for kombinasjon av spenningen. Grunnen til dette er den store normalkraften og skjærkraften som er i forbindelsen. I Tabell 7.39 ble det vist at Sweco har økt tykkelsen på stålplaten til 12 mm. I Grasshopper modellen ble det brukt en tykkelse på 8 mm for alle stålplatene. Det bemerkes også at slankheten til stålplaten i undergurten er veldig høy.

Tabell 7.47: Kontroll stålplate under- og overgurt

KONTROLLER	UTNYTTELSE FOR KNUTEPUNKT I OVERGURT	UTNYTTELSE FOR KNUTEPUNKT I UNDERGURT
Hullkantrykk	18,7%	24,9%
Skjærbrudd	18,7%	0,44%
Strek/Trykkbrudd	41,5%	51,5%
Bøyebrudd	0%	0%
Kombinasjon spenninger	96,6%	28,26%
Knekking	0%	66,9%
Slankhet	69,88	133,35

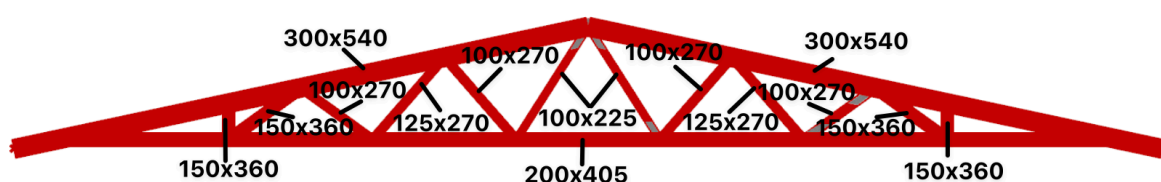
7.7 Optimalisering av limtrefagverket

I dette kapittelet presenteres resultatet fra størrelsesoptimaliseringen av limtre-fagverket. Formålet var å redusere totalvekten til strukturen. Dette resultatet inkluderer ikke vekten til stålplatene og dyblene i forbindelsene, siden det ikke er programmert et system for dette. Det er kun optimalisert for tverrsnittstørrelsen til limtre-elementene, og resultatet representerer dermed kun totalvekten til alle de optimaliserte elementene.

Resultatet for alle iterasjonene av hvert element er gitt i vedlegg 7. Til sammen ble det utført over 700 iterasjoner. Resultatet som var nærmest 100% utnyttelse for element analysen, og som i tillegg tilfredsstilte andre utnyttelseskontroller for elementet, ble dermed valgt. Videre presenteres tverrsnittstørrelsen for elementene i limtrefagverket.

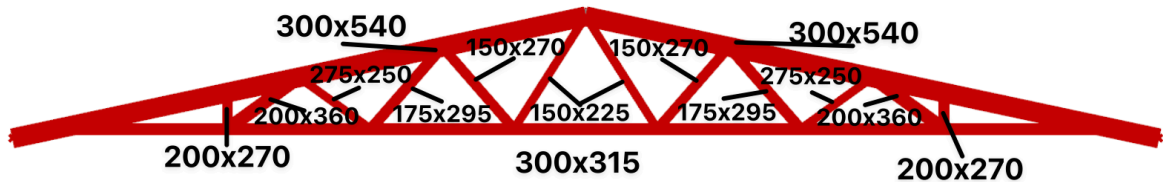
7.7.1 Optimaliserte tverrsnitt

I Figur 7.30 vises resultatet for optimalisering av tverrsnittstørrelsen til hvert element. Vekten til fagverket fra casen til Sweco, var ifølge Karamba3D 3717 kilogram, mens etter optimaliseringen ble vekten redusert til 3409 kilogram. Optimaliseringen reduserte dermed vekten med 308 kilogram, men i den optimaliserte modellen er det ikke tatt hensyn til utnyttelses kontrollene i forbindelsene. Flere av stavene måtte dermed korrigeres for å ta hensyn til utnyttelsene i forbindelsene.



Figur 7.30: Størrelsesoptimalisering fra Opossum

I Figur 7.31 vises den oppdaterte modellen, som tar hensyn til utnyttelses kontrollene til forbindelsene. I dette tilfellet ble det utført endringer av alle stavene, og undergurten. Breddene og høydene måtte økes, ettersom det oppsto brudd i forbindelsene ved bruk av tverrsnittene fra optimaliseringen. Bredden til undergurten ble økt for å ikke være mindre enn de nødvendige breddene for stavene. Tykkelsen mellom stålplatene ble også økt fra 97 mm til 105.



Figur 7.31: Korrigering av Opossum resultat

Totalvekten til den korrigerte modellen gikk dermed opp til 3786 kg. Dette er en forskjell på 377 kg fra den optimaliserte modellen, og 69 kg mer enn fagverket til Sweco. Grunnen for at totalvektene har økt, kommer av at resultatet er påvirket av knutepunkts kontrollene, som bruker den effektive tykkelsen på ytterfeltene, noe som påvirker de valgte størrelsene på tverrsnittene.

8 Diskusjon

I dette kapittelet vil først og fremst kreftene fra analysen i Kramba3D sammenlignes mot resultatene fra tradisjonell metode (hentet fra oppdragsgivers Robot-modell). Deretter diskuteres metoden og resultatene fra kontrollene som er utført i henhold til eurokodens bruddgrensetilstand i kapittel 8.2. I tillegg diskuteres metoden og resultatene fra knutepunksberegningene i kapittel 8.3. Videre diskuteres metode-delene som har skapt komplikasjoner underveis i kapittel 8.4, samt en vurdering av systemets brukbarhet i kapittel 8.5. Til slutt diskuteres resultatene fra optimaliseringsforsøkene i kapittel 8.6.

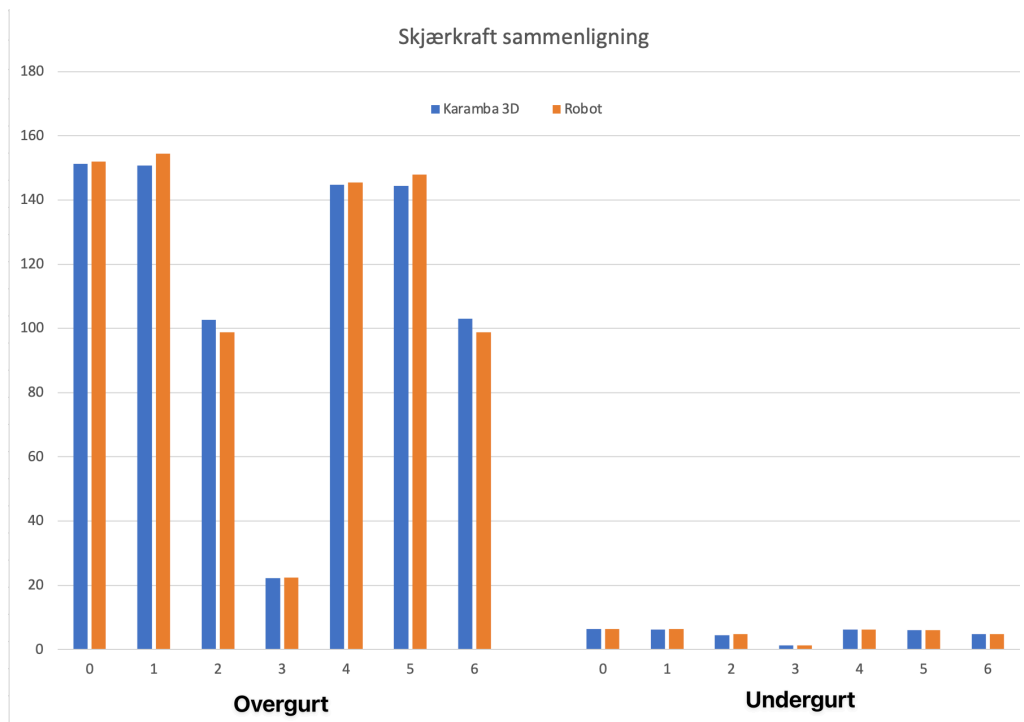
8.1 Sammenligning av krefter

Her sammenlignes resultatene fra lineær elastisk analyse i Kramba3D mot resultatene fra oppdragsgivers Robot-modell. Det var valgt å diskutere de største forskjellene, da det observeres at Kramba3D i noen tilfeller gir kraftresultater med store avvik fra Robot, mens den i andre tilfeller er nokså nøyaktig. Det forsøkes derfor å finne sammenhenger i hvordan Karamba3D håndterer de ulike analysesegmentene.

8.1.1 Skjærkraft

I Figur 8.1 sammenlignes forskjellene av skjærkreftene mellom Karamba3D og Robot. Det var kun forskjeller i over- og undergurten, og i de andre elementene var skjærkreftene de samme. Figuren viser at skjærkreftene er relativt like i de fleste lastkombinasjonene. De største forskjellene som ble bemerket i resultatet, var i lastkombinasjon 1, 5 og 6 for overgurten. Dette er lastkombinasjoner med fordelt snølast, og det observeres at Karamba3D er større enn Robot i lastkombinasjon 2 og 6. Her er snølasten sentrert mellom oppleggene til fagverket. Forskjellen på disse lastene er at lastkombinasjon 2 inneholder en vindlast som belaster fagverket i y-retning, mens lastkombinasjon 6 er uten denne vindlasten. Det skal dermed ikke være noen forskjell på skjærkraften i z-retning. Ved å sammenligne forskjellen på skjærkraften fra lastkombinasjon 2 og 6 (fra både Robot og Karamba3D), er skjærkraften den samme i Robot, mens den i Karamba3D er større i lastkombinasjon 6 enn lastkombinasjon 2. Dette skal ikke være mulig. Ved sammenligning av lastkombinasjonene 0 og 4, samt lastkombinasjonene 1 og 5, er forskjellene i skjærkraften 6 kN. Dette ser ut til å være riktig, ettersom det også er tilfellet for Robot. Her er last 0 større enn 4 og last 1 større enn 5, på grunn av at last 0 og 1 inneholder jevnt fordelt vindlast (i trykk) i samme retning som snølasten og egenlasten.

Ved å også sammenligne forskjellene på skjærkreftene i lastkombinasjon 1 og 5 (mellom Karamba3D og Robot) er det en forskjell på rundt 2%, der Robot har større krefter. Basert på dette kan det virke som at Karamba3D er dårlig på å håndtere skjevt fordelte laster, ettersom disse lastkombinasjonene har skjevt fordelt snølast som sin dominerende variabel. I tillegg øker den en last, som ikke skal være mulig å øke. Dette er noe som bergreser bruken av Karamba3D, og styrker bruken av Robot. I tillegg utfører Robot belastningsanalysene etter verifiserte standarder og vil dermed gi mer pålitelige resultater, enn Karamba3D.



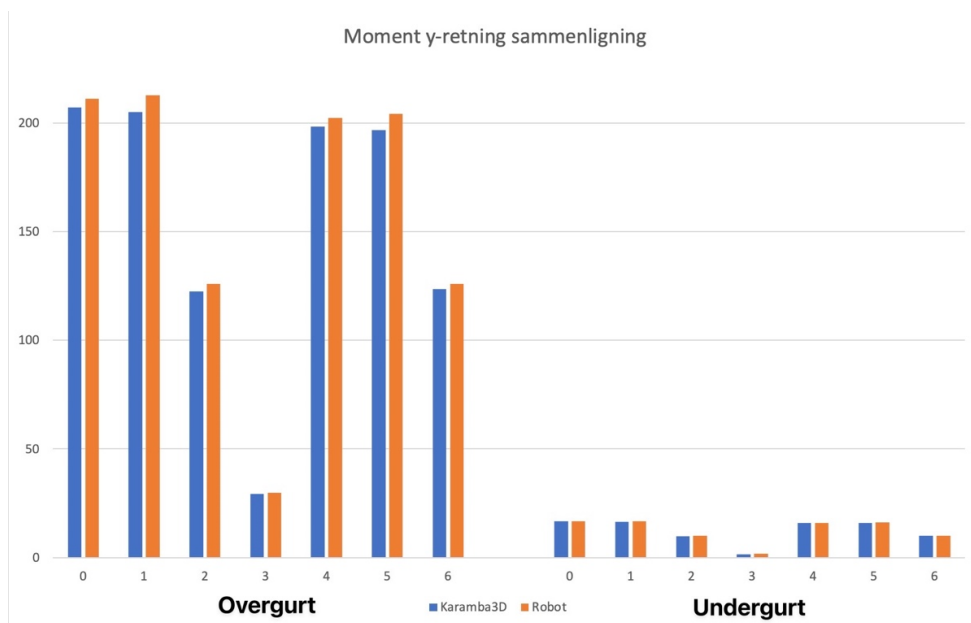
Figur 8.1: Sammenligning av skjærkraft, x-akse lastkombinasjoner, y-akse Kilonewton

8.1.2 Momentkraft y-retning

For momentkraften i y-retningen var resultatene identiske for alle stavene, ettersom det ikke oppstår moment i dem. I Figur 8.2 vises sammenligningen fra tabellene i kapittel 7.2. I undergurten er momentkreftene i y-retningen fra Karamba3D og Robot nesten identiske, men i overgurten viste resultatet bemerkelsesverdige forskjeller.

Alle momentkreftene i y-retning er størst i Robot, og de største forskjellene er i lastkombinasjon 1 og 5. Her var momentene ca. 4% større enn Karamba3D, noe som også vil utgjøre en forskjell på 4% i utnyttelseskontrollen av overgurten. Det bemerkes også at momentkraften fra lastkombinasjon 2 er mindre enn lastkombinasjon 6 for Karamba3D, noe som også var tilfellet i skjærkraften. Ved å sammenligne momentkraften fra Robot med de samme lastkombinasjonene (2 og 6), er momentet likt i begge lastene, noe som skal stemme.

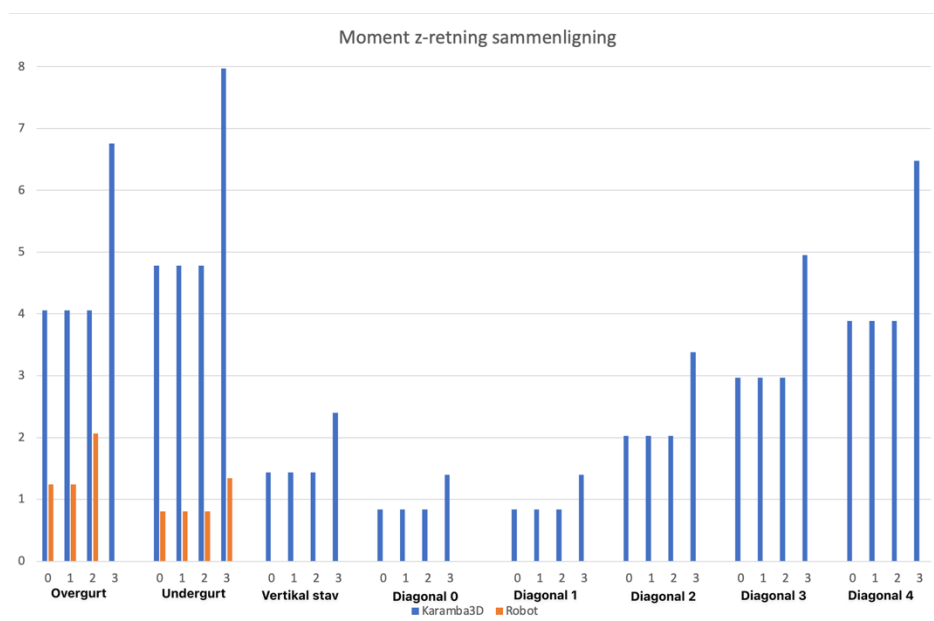
Ved å sammenligne momentkreftene i de andre lastkombinasjonene (0 og 4 / 1 og 5), er forskjellene på ca. 8 kN i både Karamba3D og Robot. Dette peker igjen mot at Karamba3D ikke kan håndtere skjevt fordelte laster, da det også her er lastkombinasjon 1 og 6 som ikke stemmer overens med Robot-modellen. Robot virker dermed mer pålitelig enn Karamba3D, på å håndtere momentkreftene i y-retningen.



Figur 8.2: Sammenligning av momentkraft y-retning, x-akse lastkombinasjoner, y-akse Kilonewton-meter

8.1.3 Momentkraft z-retning

I Figur 8.3 er alle resultatene for Momentkraften i z-retningen fra kapittel 7.2 sammenlignet i et diagram. Resultatet viser at momentkraften i z-retningen fra Karamba3D er større enn Robot i de fire første lastkombinasjonene. Dette gjelder alle elementene, og skyldes sannsynligvis fastholdningen av fagverket i z-retning. I Karamba3D ble overgurten fastholdt i alle punkter hvor stavene treffer gurten, ettersom det ikke ble laget et eget punktsystem for å kunne definere fastholdningen i Grasshopper. Vi har ikke oversikt over hvor tett fastholdningen er satt i Robot, men den er sannsynligvis satt til å være noe tettere enn dette. Det er imidlertid mulig å etablere et bedre system for dette i Grasshopper, og dermed ville resultatet sannsynligvis stemme bedre overens med Robot-modellen.



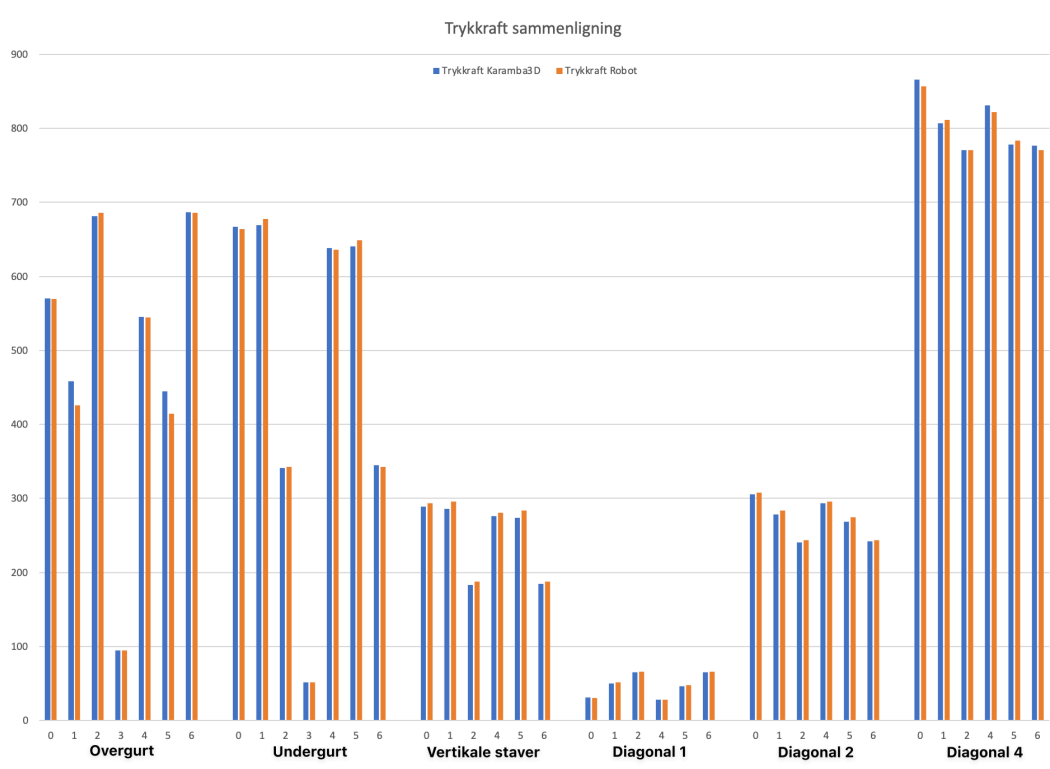
Figur 8.3: Sammenligning av momentkraft z-retning, x-akse lastkombinasjoner, y-akse Kilonewton-meter

8.1.4 Trykkraft

Resultatet for alle trykkreftene i fagverket er illustrert i et diagram i Figur 8.4. Her var det mye større variasjon mellom hvilke av programmene som hadde de største trykkreftene. I resultatene for både skjær- og momentkreftene (M_y) hadde Robot de største belastningskreftene. Det er viktig å påpeke at kreftene fra Robot-modellen antas å være det «riktige» resultatet, ettersom dette er en standardisert programvare. Så dersom kreftene fra Karamba3D er større enn kreftene fra Robot, kan man anta at resultatet er til sikker side. Derimot, vil kontrollere ikke være til sikker side, dersom man benytter krefter fra Karamba3D som er mindre enn kreftene fra Robot-modellen.

I de fleste tilfeller var det rundt 1 KN forskjell på trykkreftene mellom Karamba3D og Robot. De største forskjellene er i overgurten, og her var det 30 KN forskjell mellom trykkraften i lastkombinasjon 1 og 5 fra Karamba3D og Robot. Dette utgjør en forskjell på rundt 7%, noe som gjør at utnyttelseskontrollen også vil være rundt 7% større ved å bruke denne trykkraften fra Karamba3D. Dette resultatet var ikke tilfellet for skjær- og momentkreftene, der var kreftene fra Robot større. Det er vanskelig å forstå hvorfor dette er tilfellet for trykkraften, men igjen kan måten Karamba3D håndterer skjevt fordelte laster være grunne til dette.

Det bemerkes at trykkraften i lastkombinasjon 2 er større enn lastkombinasjon 6 i Karamba3D, mens i Robot er trykkraften den samme i begge disse lastkombinasjonene. Dette er tilfellet i alle elementene som er vist i Figur 8.4. Ved å sammenligne trykkreftene i lastkombinasjonene som er relaterte til hverandre, altså lastkombinasjon 0 og 4, 1 og 5, reduseres trykkraften på grunn av at vindlasten ikke er i last 4 og 5. Når man sammenligner reduksjonen på trykkreftene til Karamba3D og trykkreftene til Robot, er reduksjonen relativt like. Dette bygger opp under observasjonen om at Karamba3D er dårlig på å håndtere skjevt fordelte laster.



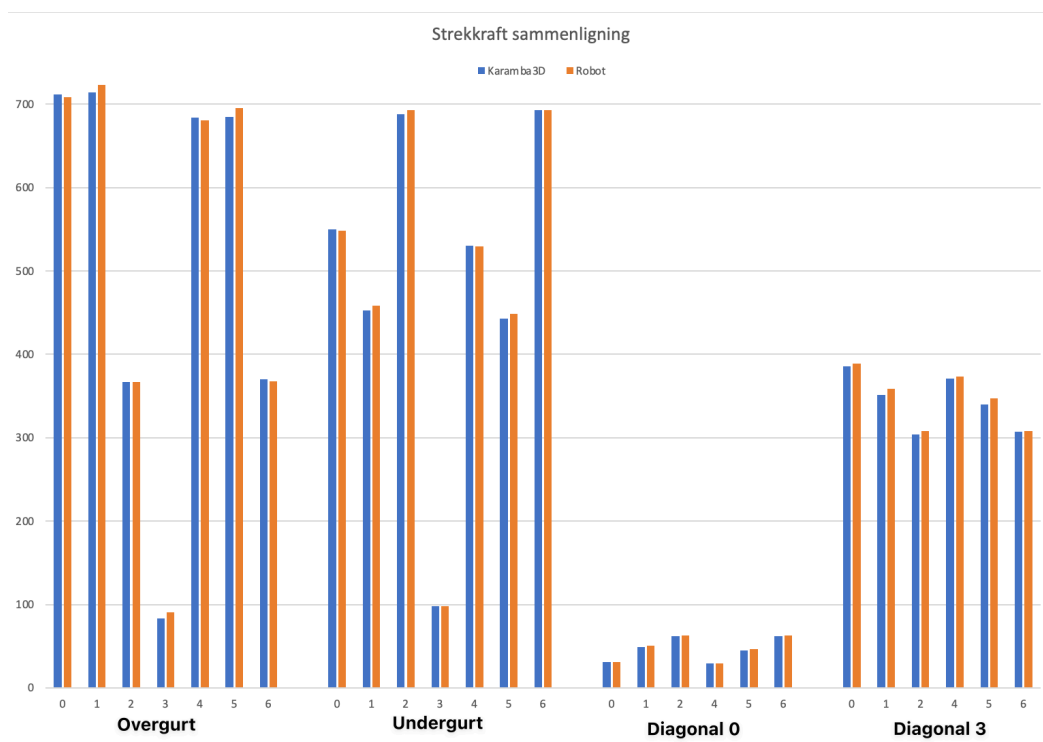
Figur 8.4: Sammenligning av trykkraft, x-akse lastkombinasjoner, y-akse Kilonewton

8.1.5 Strekkraft

Resultatet for strekkreftene mellom Karamba3D og Robot er sammenlignet i et diagram som er vist i Figur 8.5. Resultatet viste at Robot har større strekkrefter i de fleste av tilfellene i elementene som er utsatt for strekk, men forskjellene er relativt minimale. De største forskjellene er på rundt 7 KN, mens de minste er på rundt 0,3 KN. Det var igjen lastkombinasjon 1 og 5 som har de største forskjellene mellom Kramba3D og Robot. Her er det rundt 3% forskjell avhengig av størrelsen på kreftene.

Ved å undersøke lastkombinasjon 2 og 6, er resultatet for strekkreftene annerledes i forhold til skjærmoment- og trykkreftene resultatene. I diagonal 0 er strekkraften i lastkombinasjon 2 og 6 den samme i Kramba3D. Dette er derimot ikke tilfellet for diagonal 3 og begge gurtene, der lastkombinasjon 6 større enn 2. Det er vanskelig å forklare hvorfor det ikke var noen variasjon på strekkraften lastene 2 og 6 i diagonal 0, når belastningen i de andre elementene varierer.

Ved å sammenlignede strekkreftene i Kramba3D mellom de relativt like lastkombinasjonene 0 og 4, 1 og 5 i elementene fra Figur 8.5, er forskjellene mellom dem relativt like med forskjellen til Robot. Selv om Robot har litt større krefter enn Kramba3D i disse lastkombinasjonene, er differanse minimal og vil dermed ikke påvirke utnyttelsesresultatene i stor grad. Basert på resultatene må man derimot være oppmerksom på at dersom belastningskreftene som frembringes av Karamba3D benyttes, så vil resultatet være ca. 3% større.

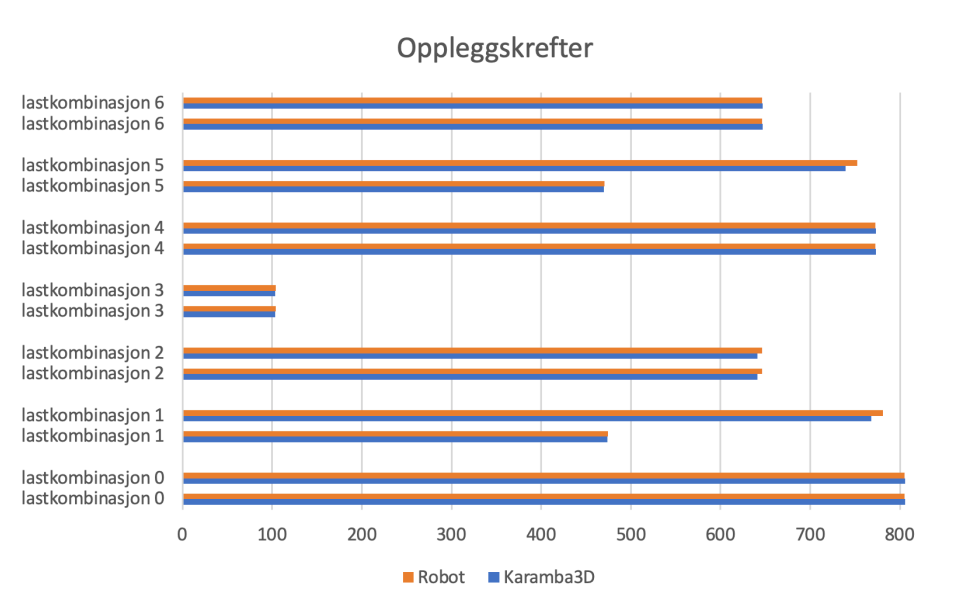


Figur 8.5: Sammenligning av strekkraft, x-akse lastkombinasjoner, y-akse Kilonewton

8.1.6 Sammenligningen av oppleggs-krefter

I Figur 8.6 er oppleggs-kreftene til lastkombinasjonene til Sweco som ble vist i Figur 5.12, og det som ble vist i resultatet i Figur 7.11 sammenlignet i et diagram. Her kommer det tydeligere frem at

lastkombinasjon 1 og 5 i Kramba3D har ca. 12 KN mindre oppleggs-kraft enn Robot, der den skjeve snølasten plasseres over overgurt 1. I tillegg ved å undersøke oppleggs-kreftene til lastkombinasjon 2 og 6 i Kramba3D, økes oppleggs-kraften i lastkombinasjon 6, mens disse er like for Robot. Dette indikerer hvorfor belastningskreftene som kommer fra disse lastene er mindre i Karamba3D enn Robot. Det er mulig at det er feil med måte lastene er programmert i C# skriptet i Grasshopper, men dette er undersøkt flere ganger. I tillegg er oppleggs-kreftene fra lastkombinasjon 0, 3 og 4 fra Karamba3D og Robot neste helt like. Basert på dette og sammenligningen mellom belastningskreftene fra Karamba3D og Robot, kan det antas at Karamba3D ikke er optimal for håndtering av skjevt fordelte laster på konstruksjonen.



Figur 8.6: Diagram for sammenligning av opplagerkrefter

8.1.7 Sammenligning med andre studier

Resultatet for sammenligningen mellom Karamba3D og Robot, viste at Robot hadde de største kreftene og var i tillegg mer pålitelige. Siden Robot er et verifisert program for belastningsanalyse, kan det basert på resultatet være bedre å bruke kreftene fra Robot. Dette stemmer med resultatet i forskningsartikkelen om Beaver, som ble presentert i kapittel 3.7.3 [6]. I denne artikkelen ble det nevnt at belastningsanalysen fra Kramba3D ikke samsvarte med belastningsanalysen fra RFEM. Det er vanskelig å vite om et belastningsresultat fra Robot og et fra RFEM ville samsvart, men ettersom disse programmene bruker verifiserte standarder for belastningsanalyser, er resultatene mer pålitelige enn resultatene fra Karamba3D.

Det kan også påpekes at det er vanskelig å få de samme svarene når det kommer til kvalitative data, ettersom dataen i Kramba3D og Robot behandles på forskjellige måter. Karamba3D er laget for å raskt utføre belastningsresultater for parametrisk design av strukturer med visuell programmering, og optimalisering av dem, mens Robot er laget for å utføre belastningsanalyse av modeller som er oppbygget i BIM. Det kan dermed være forskjeller i hvordan Karamba3D og Robot utfører FEA-analysen, noe som ble påpekt i Beaver artikkelen [6].

I resultatet til masteroppgaven som ble presentert i kapittel 3.7.5, viste resultatet at belastningskreftene fra Kramba3D og Robot stemte med hverandre [4]. Grunnen til dette kan være at det ble brukt stål og ikke tre som materiale. Det kan dermed virke som at Karamba3D ikke er optimal for belastningsanalyse når materielt er av tre, siden det behandler materialet som stål og ikke som tre.

8.2 Cluster-komponent for elementanalyse

Her diskuteres komponenten som ble programmert til å utføre de nødvendige analysene for hvert element i fagverket. Det diskuteres hvorfor disse kontrollene ble utført, hvorfor det er forskjeller i utnyttelses resultatet til beregningsrapporten til fagverket, og påvirkningen av k_{mod} verdien. Innen hver av disse delene blir gjenbrukbarheten av denne komponenten også diskutert. Dette er også noe som dikteres videre i kapittel 8.5.1, men i dette kapittel påpekes eventuelle begrensinger med denne komponenten.

8.2.1 Dimensjonerende last for beregning av utnyttelse

Når det kommer til beregning av utnyttelse for elementer med tre som materiale, vil ikke nødvendigvis den største lasten være den som er dimensjonerende i elementanalysen. Dette kommer av klimaklassen som avhenger av plasseringen og lastvarigheten til den korteste lasten, noe som har en betydelig innvirkning på trematerialet, og er derfor avgjørende i utnyttelseskontrollen av både elementene og knutepunktene. Dette gjør at utnyttelseskontrollen fra en av lastkombinasjonene 4, 5 og 6 vil alltid være dimensjonerende, på grunn av at k_{mod} er 0,9 i disse lastene og reduserer derfor motstanden til elementet mer enn de andre lastene som har k_{mod} på 1,1.

Dette er noe som ikke utføres i betong- og stålkonstruksjoner, men behøves i trekonstruksjoner på grunn av treets ortotropiske materialegenskaper. Til lengre lastvarigheten er, til mindre er k_{mod} verdien, og ved å benytte den laveste k_{mod} verdien med den største lasten, vil man utføre beregninger til sikker side. Problemet med dette er at det alltid vil overdimensjonere konstruksjonen, og i tillegg ikke være økonomisk lønnsom, ettersom materialet ikke blir fullt utnyttet.

Cluster-komponenten som ble programmert i Grasshopper for å utføre elementkontroller, velger den maksimale utnyttelsen fra alle lastkombinasjonene som output. Dette løser dermed problemet med hvilke av lastene som vil være dimensjonerende, men det skaper et annet problem. Dersom det i et annet tilfelle er flere lastkombinasjoner, må brukeren selv korrigere for dette i Clusteren. Dette er nødvendigvis ikke vanskelig for noen som kan visuell programmering og forstår programmeringen som er utført Clusteren. Likevel kan dette skape problemer, ettersom visuelle beregninger er nokså ressurskrevende. Dermed vil modellen bli mer og mer saktegående, etter hvert som flere lastkombinasjoner blir lagt til i modellen.

Det er kanskje ikke nødvendig å utføre kontroller av alle lastkombinasjonene, ettersom enkelte lastkombinasjoner ikke vil være dimensjonerende på grunn av k_{mod} verdien. Dette vil redusere antall beregninger som Grasshopper må utføre, og modellen vil kunne bli raskere. Brukeren må da selv finne de dimensjonerende lastene og deretter programmere dem inn i Karamba3D. Det vil kanskje derimot være mer effektivt å benytte systemet som det er. Da trenger brukeren kun å legge inn de

forskjellige lastkombinasjoner i Karamba3D og deretter legge inn de nødvendige k_{mod} verdier i Cluster-komponenten. Det bør antagelig gjøres noen korrigeringer i Cluteren for at systemet skal være mer effektivt. I tillegg bør ikke brukerne stole alt for mye på beregningene som utføres, og selv kontrollere hva som vil være dimensjonerende krefter for fagverket.

8.2.2 Nødvendige elementkontroller i fagverket

I den programmerte Cluster-komponenten blir det kun utført kontroll for bruddgrense, ettersom den skal bekrefte om elementet har en tilstrekkelig lastbærende kapasitet. Samtidig, er kontroll for nedbøyning også viktig ved dimensjonering av fagverk. Men i denne oppgaven er dette ikke prioritert. Grunnen til dette er at en ny komponent som beregner lastene i bruksgrense måtte blitt programmert, og siden de fleste kontrollene som utføres er i bruddgrense, ble ikke bruksgrensekontroll prioritert. Dersom bruksgrensen hadde blitt programmert inn i Grasshopper-modellen ville dette også gjort modellen enda tregere.

Fagverket er designet for å overføre kreftene til oppleggene, noe som vises i moment-, skjær, og normalkraftdiagrammet som ble vist i kapittel 7.3. Derfor oppstår det store bøyekrefter over de vertikale stavene. Overgurtene har trykkrefter i toppunktet frem til diagonal 4, som overfører trykkreftene til de vertikale stavene og undergurten ved oppleggene. Undergurten har derimot strekkrefter på midten, som overføres til diagonal 3 og deretter til overgurtene. Dette gjør at elementene som utgjør overgurtene og undergurten har både strekk og trykkrefter. Under- og overgurtene måtte derfor kontrolleres for strekk og trykk, og kombinasjonen med bøyemoment både i y- og z-retningen. Det oppstår moment i z-retningen på grunn vindlasten fra siden på fagverket, og det ble derfor kontrollert for dette. Denne kontrollen for moment i z-retningen kunne derimot blitt fjernet, ettersom kreftene er neglisjerbare ved tettere fastholdning i z-retningen (som Robot analysen viste).

Diagonalstavene 0 og 3 er strekkstaver, mens diagonalstavene 1, 2, 4 og de vertikale stavene er trykkstaver. Disse måtte dermed kontrolleres for trykk og strekk. I kapittel 3.4.7 ble det presisert at diagonalene som er utsatt for strekk skal bruke nettotverrsnitt som er 70% av bruttotverrsnittet for kontroll av strekkraften [58]. Dette er noe som ikke ble utført i den programmerte Clusteren, ettersom det ikke ble utført i beregningsrapporten fra oppdragsgiver. Sweco kontrollerer heller at utnyttelsesverdien ikke overstiger 70% til slutt. Uansett, er utnyttelsen i strekk for de diagonale stavene være veldig lav, og selv om nettotverrsnittet hadde blitt brukt, ville ikke utnyttelsen blitt overskredet. For de diagonale stavene er det nemlig utnyttelsen i knutepunkts-kontrollene som vil bestemme tverrsnittstørrelsen. Nettotverrsnitt skulle nok likevel blitt programmert inn i elementanalysen, ettersom nettotverrsnittet tar hensyn til hullene for dyblene og dimensjoneringen blir dermed utført til sikker side [58]. For å programmere dette inn i Clustern kan et Python-skript benyttes, hvor skriptet programmeres slik at det reduserer bruttotverrsnittet til stavene når det oppstår strekk. Det må derimot kanskje lages en egen Cluster-komponent som kun utfører element-analyse av stavene, ettersom det både er strekk og trykkrefter i begge gurtene, og dersom samme cluster-komponent brukes, vil bruttotverrsnittet til begge gurtene reduseres i strekkkontrollen. Det kan antagelig programmeres et skript som kan løse dette problemet, men det trengs en større undersøkelse for å løse dette. Dette er også noe som begrenser komponenten som er laget for

elementanalysen i Grasshopper. I henhold til casen fungerte den, men i et annet fagverk kan dette være ønskelig å korrigere for.

Overgurten vil ikke nødvendigvis ha knekking i z-asken på grunn av takflaten forhindrer dette, men knekking i y-retningen må kontrolleres. I undergurten må sideveis knekking kontrolleres på grunn av den vindlasten i lastkombinasjon løfter fagverket oppover. I de diagonale og vertikale stavene må det kontrolleres for knekking i begge retninger. Her kan knekk lengden til trykkstavene reduseres med 10%, på grunn av innspenningen som oppstår når flere staver er tilknyttet knutepunktet [58]. For knekking ut av planet skal derimot den fulle systemlengden brukes. I Cluster som utfører elementanalysen reduseres ikke knekk lengden. Her brukes full systemlengde for begge knekkings kontrollene, ettersom dette ble utført i beregningsrapporten fra oppdragsgiver. I dette tilfellet er utnyttelsen for knekking i alle stavene lave uansett, så det ville ikke vært behov for reduksjon. Her er det derfor muligheter for forbedring i Cluster-komponenten som utfører kontrollene, da bruk av redusert knekk lengde kan gi mer realistiske verdier. Det er imidlertid noe som «kan» benyttes ifølge limtreboka, og det å overse dette vil gi beregninger til sikker side [58]. For å programmere dette inn i clusteren må knekk lengden i y-retningen multipliseres med 0.9, og et Python-skript kan brukes for at dette skal utføres når det er trykk i staven. Problemet er igjen at dette vil også påvirke kontrollene for knekking i begge gurtene. Ved å programmere inn at knekk lengden i trykkstavene skal multipliseres med 0.9, i de tilfellene når det er flere enn en stav i forbindelsen, medfører dette at knekk lengden i gurene også reduseres. Det vil gi kontroll som ikke er til sikker side for overgurten. Det måtte derfor ha blitt programmert egne komponenter som utfører kontroll av stavene, og egne komponenter for gurtene.

Basert på det som er diskutert, virker det som at det burde blitt programmert egne komponenter for stavene og begge gurtene, siden ikke alle kontrollene vil være relevante å utføre og på grunn av enkelte elementer trenger å kontrolleres med bestemte hensyn. I stavene oppstår det ikke bøyemomenter og skjærkrefter, noe som gjør at det ikke er nødvendig å utføre kontroll av dette, men siden det er skjærkrefter i overgurtene og bøyemomenter i begge gurtene ble det utført kontroll for dette. For oss var det enklere å programmere en komponent som utfører alle de nødvendige kontrollene som måtte gjøres i alle elementene, i motsett til å programmere egne komponenter for stavene og begge gurtene. Problemet med denne metoden, er at Grasshopper utfører unødvendige kontroller og blir dermed tregere. Dette gjorde det også vanskeligere for oss å både programmere nye deler og redigere dem i modellen. Dette er noe som ble oppdaget etter hvert som flere deler ble programmert inn i Grasshopper-modellen.

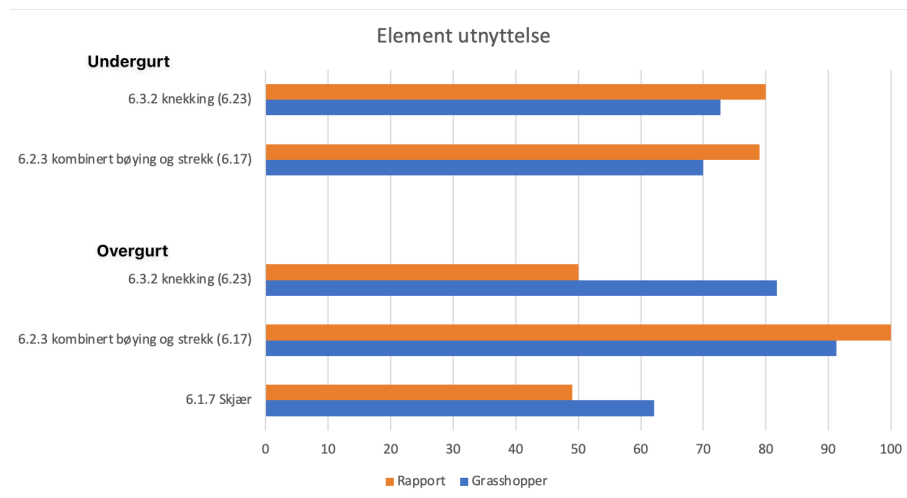
I dette fagverket utsettes elementene for spenninger i fiberretningen, og det er ingen krefter som er vinkelrett på elementene. Det er derfor ikke utført kontroll for torsjon, og kontrollen for kombinasjon av spenninger, følger dermed kontrollen for søyler og ikke bjelker som er gitt i EK5 [63]. Dersom torsjon eller at krefter påføres vinkelrett på fiberretningen, må det programmeres enten nye komponenter, eller å omprogrammere de programmerte Clusterene som gjør at dette blir tatt hensyn til. Dette er noe som begrenser denne programmerte Clusterene som utfører kontrollen, den utfører kun bestemte kontrollene basert på kreftene som er i fagverket til casen. Noe som gjør at den kan brukes i bestemte forhold. Det er selvsagt veldig vanskelig å programmere en komponent om utfører alle nødvendige kontrollen for alle strukturer, noe som igjen viser at det burde blitt programmert flere clusterer i henhold til hvilket element det er og hva som påvirker dette elementet. Dette gjør

også at brukeres selv må vurdere hva som vil påvirke elementene og ikke stole kun på resultatene fra komponentene.

8.2.3 Sammenligning av utnyttelse

Over- og undergurt

Sammenligningen mellom utnyttelseskontrollene for over- og undergurten fra kapittel 7.4 er presentert i et diagram i Figur 8.7. Diagrammet viser at det forskjeller i alle utnyttelseskontrollene, det er flere grunn for dette, men hovedgrunnen er forenklingen som ble utført i Grasshopper med analysen av over-, og undergurten. Hvor dette har gjort de største utslagene, spesifiseres og diskuteres videre. I kontrollen for knekking i overgruten etter 6.3.2 i EK5 [63], er det en forskjell på over 30%. Det er forskjellige grunner for dette. Den første grunnen er forskjellen mellom moment- og strekkraften som er brukt i beregningsrapporten til fagverket. I rapporten er det brukt et moment på 75 KNm og en trykkraft på 666 KN, mens i Grasshopper-modellen er momentet 198 KNm og trykkraften er på 545 KN. Den andre grunnen er forskjellen på knekk lengden. I dette tilfellet kommer forenklingen av element analysen fra Grasshopper-modellen av overgurten tydelig fram. I Grasshopper-modellen brukes de største belastningskreftene fra alle elementene som utgjør overgurten. Dette vil si at moment- og trykkraften som brukes i knekkingskontrollen er fra to forskjellige elementer som utgjør overgurten. I tillegg brukes den lengste elementlengden i overgurten. Dette gjør at kontrollen av knekking er veldig konservativ og kontrollerer ikke knekkingen for elementet som er i tykk. Dette overdimensjonerer begge gurtene, og begrenser påliteligheten av forenklingen som er utført i Cluster-komponenten.



Figur 8.7: Utnyttelse under- og overgurt vist i prosent %

Kontrollen for knekking i undergurten er det en forskjell på ca. 7%. Denne viser begrensingen med belastningskreftene fra Karamba3D. I denne kontrollen er kreftene fra lastkombinasjon 5 dimensjonerende for knekking, og momentet fra Karamba3D var 15,88 KMm, og trykkraften var 640,6 KN. I rapporten til fagverket er momentet 16,18 KNm og trykkraften 650 KN. Dette gjør at utnyttelsen som beregnes i Grasshopper blir mindre, noe fremhever begrensingen med å bruke kreftene fra Karamba3D.

I kontrollen for bøyning og strekk i undergurten er det en forskjell på 9% mellom rapporten til fagverket og det som er beregnet i Grasshopper-modellen. I denne kontrollen virker det som at

rapporten har brukt det største moment- og strekkraften fra alle lastkombinasjonene. Momentkraften som er brukt er på 16,18 KNm som er fra lastkombinasjon 5, mens strekkraften som er brukt er 695 KN som kommer fra lastkombinasjon 6. Denne strekkraften er litt større enn strekkraften som er gitt i tabell 7.2, og antas å være forstørret. I Grasshopper modellen kommer den dimensjonerende utnyttelsen fra lastkombinasjon 6, der bøyemomentet er 10,16 KNm og strekkraften er 693 KN. Forskjellene i bøyemomentet er dermed på 37% og utgjør en forskjell på 7% i utnyttelse for bøyemomentet. I tillegg siden det er en forskjell på strekkraftene, gjør dette at utnyttelsen for kombinasjon av bøyning og strekk får en forskjell på 9%.

Når det kommer til utnyttelsen for kombinert bøyning og strekk i overgurten, er denne 100% i rapporten til fagverket, mens den er 91% i Grasshoppermodellen. Her kommer svakheten til Karamba3D godt frem, ettersom denne kontrollen kombinerer bøyemomentet og strekkraften fra lastkombinasjon 5, der Robot hadde ca. 4% større moment, og 1,5 % større strekkraft. Selv om det ikke var store forskjell på belastningskreftene, viser dette at det blir store utslag når de kombineres.

For utnyttelse av skjærkraften som er i overgurten, er det en forskjell på 13%. Det er flere grunner til denne forskjellen. I Grasshopper modellen kommer den dimensjonerende skjærkraften fra lastkombinasjon 4, som er på 146 KN og gir dermed en utnyttelse på 63%. I beregningsrapporten til fagverket er skjærkraften fra lastkombinasjon 2 brukt, som er på 102 KN. Dette kan dermed antas som en skrivefeil i rapporten, siden lastkombinasjon 2 ikke vil være dimensjonerende for overgurten. Det kan derimot være at Sweco har undersøkt hvordan fagverket håndterer lastkombinasjon 2, men siden det er presisert at de maksimale utnyttelsene vises, kan det likevel antas å være en skrivefeil.

Som nevnt i begynnelsen av dette delkapittelet, kommer flere av forskjellene på grunn av forenklingen som ble utført for kontroll analysene av begge grutene. Dette gjør at resultatene som er presentert i Tabell 7.10 og Tabell 7.12 er veldig konservative. Det skulle blitt utført kontrollanalyse for hvert element som utgjør over- og undergurten, siden dette ville gitt de reelle utnyttelsene for hvert element, noe som også ville gjort sammenligningen med beregningsrapporten bedre. Problemet er igjen at, flere Cluster-komponenter som utfører analysene må legges inn, noe som vil gjøre Grasshopper modellen tregere enn det den allerede var, noe som viser begrensningen med å programmere kontroll analyser i Grasshopper.

Vertikale- og diagonalstavene 0 og 1

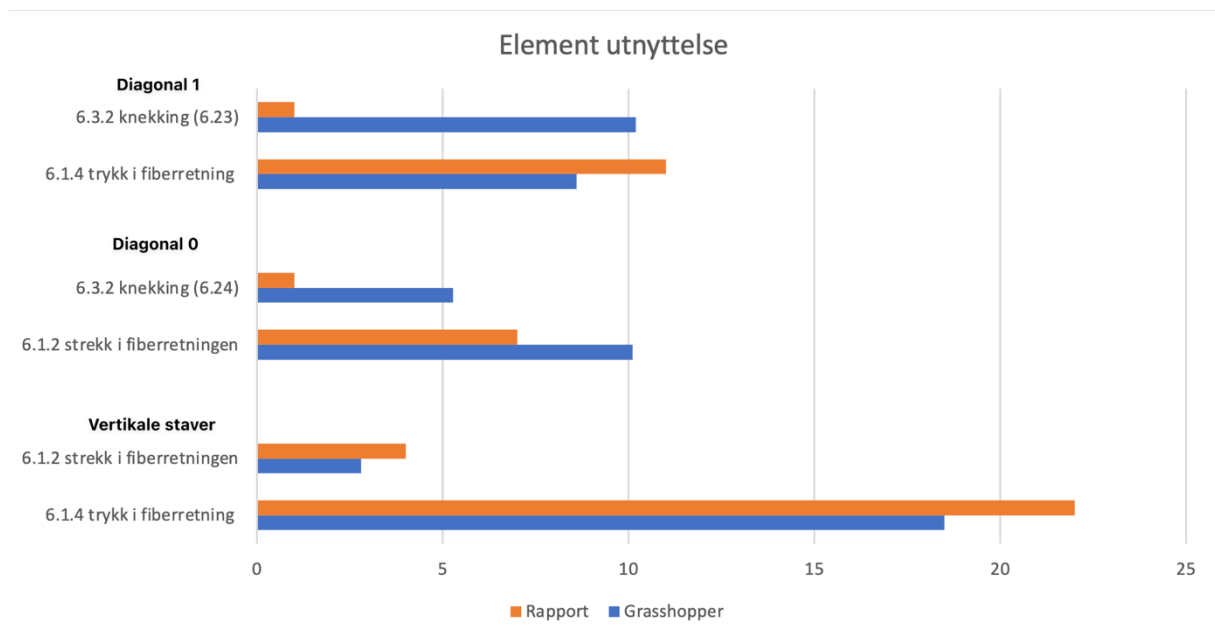
I Figur 8.8 er det laget et diagram som sammenligner utnyttelsesk kontrollene som er presentert i kapittel 7.4 for de vertikale og diagonalstavene 0 og 1. Diagrammet viser at det er relativt store forskjeller mellom utnyttelsesk kontrollene som kommer fra Grasshopper og rapporten til fagverket.

For kontrollen av knekking er det en forskjell på 9% mellom Grasshopper og beregningsrapporten til fagverket. Dette antas å komme av skrivefeil i rapporten til Sweco, ettersom kontrollen for knekking kontrollerer med strekkraften fra lastkombinasjon 3 og ikke trykkraften fra Lastkombinasjon 6. Det er også en skrivefeil i kontrollen for trykk i fiberretningen, noe som ble påpekt i kapittel 7.4.5, og er korrigert i diagrammet i Figur 8.8. Dette gjør at kontrollen i rapporten bruker trykkraften i strekkkontrollen, der beregningen bruker $f_{t,0k}$ og ikke $f_{c,0k}$ som dermed øker utnyttelsen.

For kontrollen av strekk i fiberretningen for diagonal 0, er utnyttelsen fra Grasshopper ca. 3% større enn utnyttelsen fra rapporten. Dette kommer av at beregningsrapporten har utført kontrollen med et større tverrsnitt i forhold til det som var beskrevet i Figur 5.3. I Grasshopper modellen ble geometrien fra Figur 5.3, fulgt. Forskjellen på tverrsnittet er 17500 mm^2 og vil dermed redusere utnyttelsen. Dette påvirker også kontrollen for knekking, der utnyttelsen fra Grasshopper er ca. 3% større.

I kontrollen for strekk i fiberretningen til den vertikale staven er utnyttelsen 4% i rapporten, mens den er 3 i Grasshopper modellen. Dette kommer av at strekkraften fra lastkombinasjon 3 er 1 KN større i Roboten, enn den er i Kramba3D. I dette tilfellet er forskjellen veldig lite, men igjen viser det begrensingen med Kramba3D.

Kontrollen for trykk i fiberretningen for den vertikale staven, viser at det er en forskjell på 3% i utnyttelseskontrollen. Grunnen til dette kommer av at rapporten har kontrollert med en trykkraft på 295 KN som kommer fra lastkombinasjon 1, mens Grasshopper modellen har kontrollert etter lastkombinasjon 4, som har en strekkraft på 277 KN. Rapporten har mest sannsynlig brukt k_{mod} verdien til kortidslasten og ikke øyeblikklasten som egentlig skal brukes for lastkombinasjon 1. Dette gjør at denne staven er overdimensjonert, men ettersom utnyttelsen er generelt lav kan dette anses som greit.



Figur 8.8: Utnyttelse vertikal og diagonalstaver 0 og 1 vist i prosent %

Diagonal 2, 3 og 4

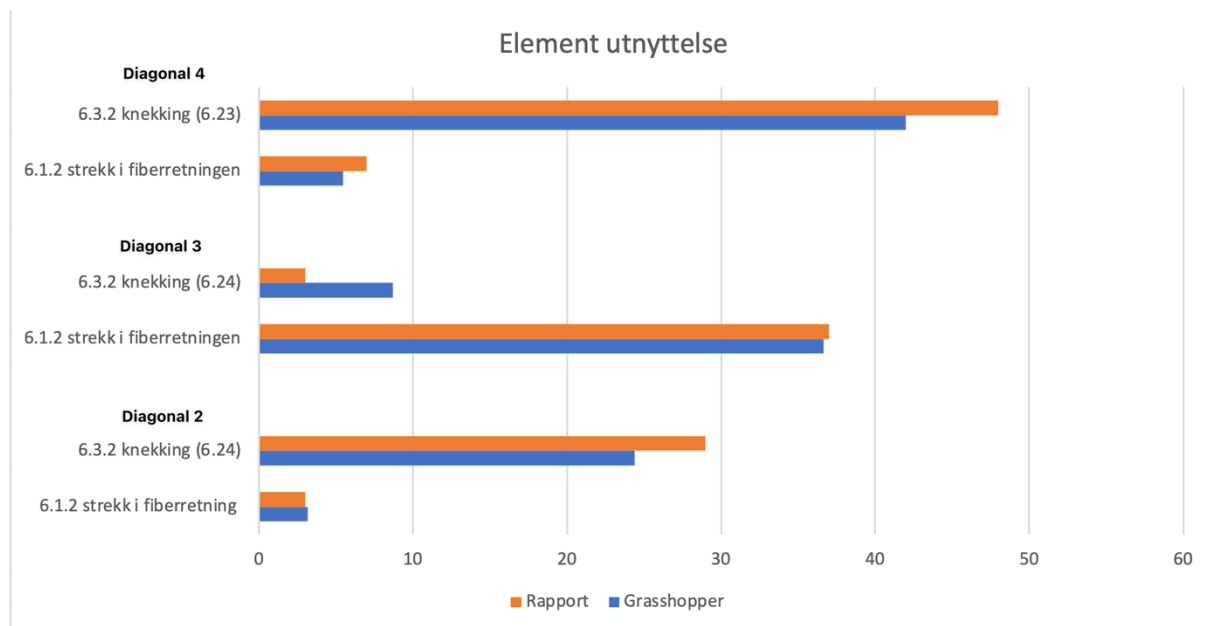
I Figur 8.9 vises sammenligningen av utnyttelsene fra kapittel 7.4 for diagonal 2, 3 og 4 i et diagram. Utnyttelsen for knekking i diagonal 4 er ca. 6% større i rapporten til fagverket enn det som er beregnet i Grasshopper. Grunnen til dette kommer av at i rapporten er det kontrollert med den største trykkraften, som kommer fra lastkombinasjon 1 og har dermed vindlast. Denne er dokumentert som kortidslast i rapporten, når vindlast er ifølge EK5 en øyeblikklast [63]. k_{mod} verdien til kortidslasten

kan dermed antydes å være bruk i denne kontrollen. I tillegg er utnyttelsen til Grasshopper fra lastkombinasjon 4, som har en trykkraft på 831 KN. I lastkombinasjon 1 er trykkraften 856 KN, noe som også gjør at det er en forskjell mellom resultatene.

Kontrollen for strekk i fiberretningen i diagonal 4 er det en forskjell på ca. 1,5 %. Selv om denne forskjellen er liten, er det likevel viktig å påpeke at k_{mod} for kortidslasten er brukt i rapporten til fagverket for denne kontrollen. Strekkraften kommer fra lastkombinasjon 3, som inkluderer en vindlast, og dermed skal k_{mod} verdien for øyeblikklast brukes. Selv om strekkraften var større i Karamba3D, utgjorde k_{mod} verdien 1,5% forskjell på utnyttelsen.

Kontrollen for strekk i fiberretningen for diagonal 3 er nærmest identiske, ettersom strekkraftene fra Robot og Karamba3D har en liten forskjell. Når det kommer til kontroll av knekking som utføres på grunn av lastkombinasjon 3, er det derimot en forskjell på ca. 6%. Dette kommer av at fagverket i Grasshopper-modellen ikke helt fastholdt mot kreftene i z-retningen, noe som gjør at momentkreftene i z-retningen blir tatt med i kontrollen for knekkingen.

Sammenligningen i Figur 8.9 viser at utnyttelsen for diagonal 2 at knekking er ca. 4% større i beregningsrapporten enn det som er beregnet i Grasshopper-modellen. Grunnen til dette er forskjellen mellom trykkraftene i lastkombinasjon 4 fra Karamba3D og Robot. Noe som igjen viser begrensningen med Karamba3D. Kontrollen for strekkraften fra lastkombinasjon 3, er derimot er nesten identiske.



Figur 8.9: Utnyttelse diagonal 2, 3 og 4 vist i prosent %

8.2.4 Utnyttelse av fagverk

Bestemmelsene som angis i punkt 9.2.1(1) og 9.2.1(5) i Eurokode 5 [63], forutsetter at det brukes forenklet konstruksjonsanalyse i henhold til prinsippene for fagverksmodellen som beskrevet i kapittel 3.4.3 [63]. Prinsippene for fagverksmodellen baseres blant annet på at alle elementer kun overfører aksialkrefter, og at alle lastene angriper i knutepunktene. På den andre siden utføres konstruksjonsanalysen for denne oppgaven i henhold til prinsippene for «blandet modell», hvor

gurtene regnes som bjelkeelementer med bøyingsstivhet og lastene kan angripe hvor som helst [22, 64]. Dermed kan ikke konstruksjonsanalysen lenger regnes som «forenklet», så det å overholde kravene for maks 70% utnyttelse i alle kontroller for både bruddgrensetilstand og forbindelser, vil være svært konservativt. Derfor tar Sweco utgangspunkt i å bare overholde disse kravene for kontroll av stav-elementene (som hovedsakelig overfører aksialkrefter), samt for kontroll av forbindelser. Derimot, er kontrollene for over- og undergurt dimensjonert for å overholde 100% utnyttelse [116]. Vi ser imidlertid at kravet fra punkt 9.2.1(1) i Eurokode 5, om at trykkspenninger fra kombinert bøyning og aksiallast begrenses til 90%, også er overholdt i gurtene.

8.3 Cluster-komponent for knutepunkts beregninger

Her diskuteres resultatene fra komponenten som ble programmert til å kontrollere knutepunktene i forbindelsene til fagverket.

8.3.1 Belastningskrefter i knutepunkt

Siden det var forskjeller på belastningskreftene mellom Karamba3D og Robot, gjør dette at det også blir forskjeller på belastningskreftene som brukes i kontrollen av knutepunktene. Kraftene som er vist i kapittel 7.5 for Sweco sine knutepunkt er hentet direkte fra beregnings rapporten og ikke fra Robot modellen. Enkelte av kreftene til Sweco, er forstørret for å være til sikker side i kontrollen av knutepunktene. Spørsmålet blir dermed om kreftene fra Karamba3D burde økes, ettersom resultatet for belastningskreftene viser at Karamba3D har ca. 5% mindre avhending av størrelsen og hvilke lastkombinasjoner det kommer fra. Dette er ikke vanskelig å programmere inn økning av krefter i Cluster-komponenten som utfører analysen av knutepunktene. Dette kan også være hensiktsmessig for å øke påliteligheten til Karamba3D. Problemet er usikkerheten med Karamba3D, i enkelte tilfeller ble laster som ikke skal økes økt, og i andre tilfeller var lastene større enn Robot. Dette gjør at usikkerheten fortsatt er der, selv om belastningskreftene økes.

I den programmert knutepunkts beregningen i Grasshopper, utføres analysen basert på belastningskreftene fra Lastkombinasjon 4, 5 og 6. Dette er på grunn av at disse vil gi de dimensjonerende lastene for knutepunktene i henhold til k_{mod} verdien, og var hensiktsmessig, ettersom det redusert programmeringen for beregning av utnyttelsen. Problemet er at man ikke kan undersøke hvordan knutepunktene blir påvirket av lastkombinasjon 3, eller de andre lastene. I rapporten til Sweco [116], ble det utført en analyse av knutepunktet i diagonal 4 med lastkombinasjon 3. Dette er ikke mulig å undersøke i Grasshopper modellen, noe som begrenser bruken av den. Det er mulig å programmere for dette, men da må store deler av clusteren omprogrammeres.

Skjærkraft splitting

I for beregning av denne kraften brukes den største skjærekraften som er på en side av dybelgruppen, i motsetning til differansen mellom skjærkreftene som opptrer i gruppen. Spørsmålet her, er om differansen mellom skjærkreftene heller burde brukes, ettersom denne er både større og brukes i beregningen av resultantkraften på dybelgruppen. Dette vil også gjøre kontrollen mer konservativ og man vil være på den sikre siden. I en dybel-forbindelsen kan splitting oppstå i hullkantfastheten til tredelen, og ved å bruke den største skjærekraften som er på gruppen, kan man potensielt identifisere de svake hullkantfasthetene rundt dybelene. Det kan dermed antas at den

største skjærkraften viser hvor den kritiske delen av dybel-gruppen vil være. Ved å bruke skjærkraften fra differansen reflekteres ikke det kritiske tilfellet for splitting i forbindelsen. Det antas dermed at det kan være mer pålitelig å bruke den største skjærkraften, ettersom muligheten for splitting vurderes mer nøyaktig.

8.3.2 Sammenligning av utnyttelser

Effektiv tykkelse på ytterfelt

Selv om beregningsrapporten til fagverket kontrollerer knutepunktene med større belastningskrefter, har Grasshopper ofte større utnyttelsesverdier for både kontroll av forbinder og skjærblokkbrudd. Dette kommer av at Grasshopper bruker den effektive lengden til dybelen i de ytterste feltene i tredelen. Denne brukes når den reelle tykkelsen til ytterfeltene er større enn 40% av innerfeltet mellom stålplatene. I beregningsrapporten til fagverket er det brukt full lengde av dybelen, noe som gir en bygningsmessig større kapasitet. Resultat fra denne oppgaven, kan ikke bekrefte at å bruke fulle lengden til dybelen i det ytterste tredelen er nødvendigvis feil, ettersom det ikke nevnes noe i EK5 om at den effektive tykkelse skal brukes når den ytterste delen er tykk [63]. Men basert det som nevnes i på rapporten til Vegvesenet [92], om at dybelenes innspenningsmoment i stålplaten er den samme som momentet til dybelen i innerfeltet med samme stålplaten, påpekes dette at kapasiteten til dybelen ikke er helt lik i hele ytterdelen. Dette gjør at kontrollen som er utført i Grasshopper er til sikker side.

Den effektive tykkelsen til ytterdelene ble også bruk i kontrollen for skjærblokk brudd. Denne er ikke helt riktig, ettersom rapporten til Vegvesenet påpeker at den effektive tykkelsen til ytterfeltet skal brukes for kontroll av pluggskjærbrudd, og full lengde skal brukes i skjærblokkbrudd [92]. I Grasshopper er det dermed kontrollert for pluggskjærbrudd og ikke for skjærblokkbrudd. Dette blir nødvendigvis ikke feil for alle knutepunktene, ettersom hvilke av skjærbruddene som oppstår avhenger av bruddformen. Grunnen for at den effektive tykkelsen ble bruk for skjærblokkbrudd i Grasshopper, er fordi høringen om NS-EN 1995-1-1/prNA:2023 [170] påpekte å fjerne t_1 tykkelsen i skjærblokk kontrollen, samt å endre beregningsformelen A.1, slik at den karakteristiske kapasitet til blokk- og pluggskjær reduseres i forhold til hva som er gitt i dagens EK5 (NS-EN 1995-1-1:2004+A1:2008+NA:2010) [63]. Det ble dermed bestemt å bruke den effektive lengden, ettersom dette tar hensyn til eventuelle endringer som kommer i en fremtidig EK5. I tillegg var dette enklere i henhold til programmeringen som måtte blitt utført. Komponenten måtte blitt omprogrammert slik at den beregner blokk- og pluggbrudd, basert på bruddformen i knutepunktet. Denne forenklingen begrenser derimot gjenbrukbarheten til den programmert komponenten for kontroll av knutepunkt.

Overgurt

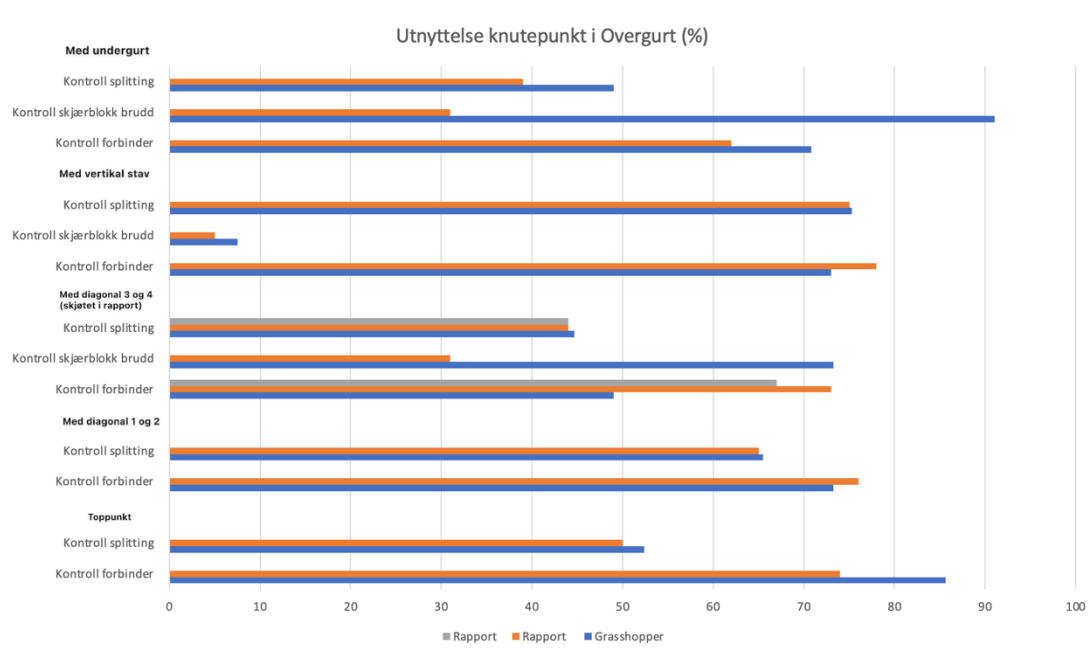
I Figur 8.10 vises utnyttelsene fra knutepunkts beregningene for overgurten fra Grasshopper og rapporten i et diagram. Utnyttelsene som ble null, er ikke tatt med i diagrammet.

Diagrammet i Figur 8.10 viser at det er store forskjeller i enkelte tilfeller, spesielt i forbindelsen med undergurten er det en forskjell på 60% i skjærblokk brudd kontrollen. Dette kommer at av Sweco har en større stålplate, og et større innerfelt i forbindelsen. I Grasshopper bruker alle forbindelsene samme tykkelse på stålplatene og innerfeltet mellom stålplatene. Grunnen for at Grasshopper ikke har samme innerfelt er at dette ikke ble oppdaget før resultere ble skrevet inn i tabellene i kapittel

7.5, noe som dermed påvirker resultatet og sammenligningen. Dette er noe som enkelt kan forandres i Grasshopper-modellen, men resultatet må skrives inn på nytt i tabellene, noe som dessverre ikke var tid til. Forskjellen på innerfeltet, stålplatene, avstandene mellom dybelene og bruken av effektiv dybel lengde er grunnen for at forskjellen på utnyttelsen er så stor. I tillegg var denne forbindelsen vanskelig å utforme, på grunn av at antallet av dybeler bestemmes av to parameter. Dette diskuteres mer i kapittel 8.5.2.

I forbindelsen med diagonal 3 og 4 hadde Sweco skjøtet overgurten i punktet hvor det oppstår tykk- og strekkrefter i overgurten. Dette er ikke utført i Grasshopper modellenes, der brukes heller differansen mellom trykk- og strekkreftene i kontrollen. Dette gjør at det ble bruk flere dybler i forbindelsen, noe som ikke er fordelaktig både for miljøet og det økonomiske. I dette tilfellet er utnyttelsen på forbinde ren lav, men i kontrollen for skjærblokk brudd er den derimot høy. Spørsmålet er om kontrollen for skjærblokkbruddet er for konservativt i Grasshopper basert på det som ble diskutert tidligere i kapittelet.

I forbindelsen med den vertikale staven, bemerkes det at Sweco har brukt en lavere tykkelsen på ytterfeltene i beregningen av kapasiteten. Denne er mindre enn den effektive tykkelsen som brukes i Grasshopper, og det er uklart hvorfor dette er tilfellet for dette knutepunktet. Dette gjør at resultatene blir ganske like.



Figur 8.10: Sammenligning kontroller i overgurt

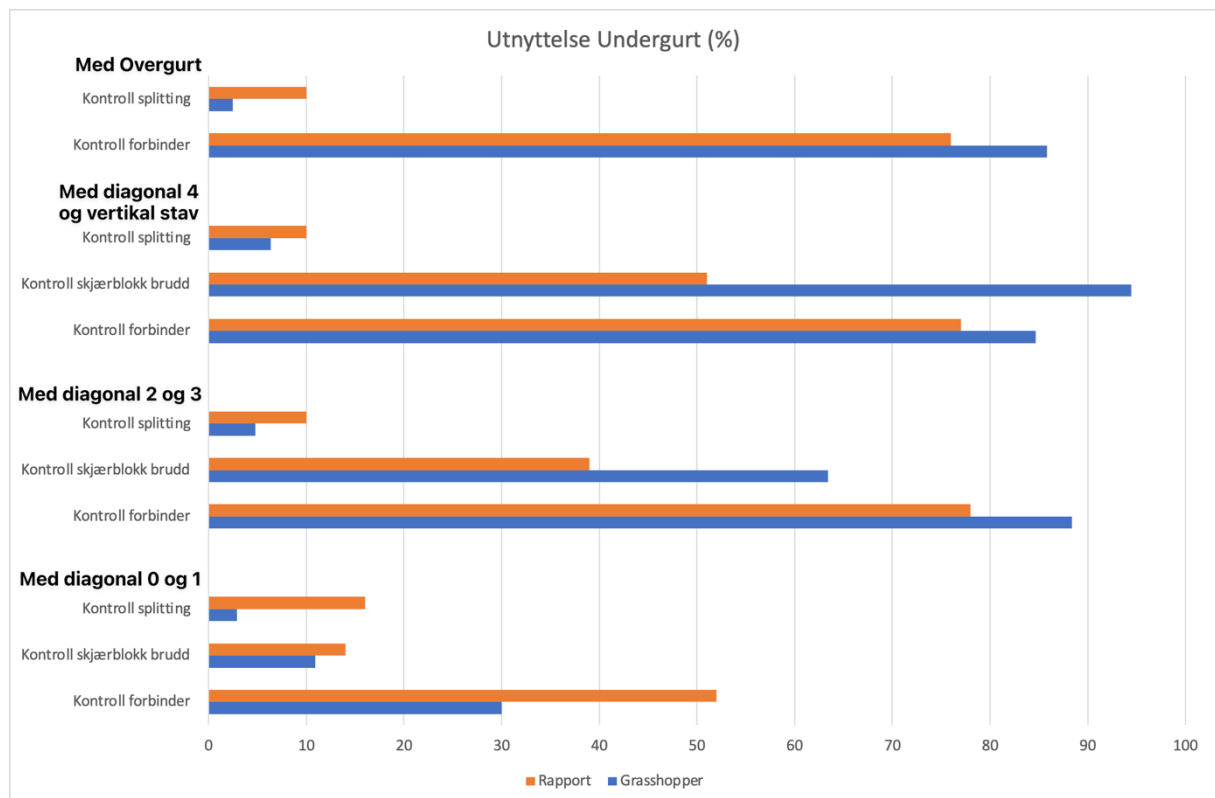
Undergurt

I Figur 8.11 vises sammenligningen av utnyttelsen for knutepunktene i undergurten fra Grasshopper og rapporten til fagverket i et diagram.

I alle knutepunktene i undergurten har Sweco lagt til ekstra skjærkrefter i kontrollen, noe som dermed påvirker resultatet. I forbindelsen med diagonal 0 og 1 er det en forskjell på over 20% i

utnyttelses verdien, som nevnt i kapittel 7.5.5 kommer dette av at normalkraften som er dobbelt så stor som kraften i Grasshopper. Det er vanskelig å forstå hvorfor denne er dobbelt så stor, når differansen mellom strekkraftene som oppstår i undergurtene ikke blir over 77 KN i noen av lasttilfellene.

I de andre knutepunktene er kommer forskjellen igjen av de redusere ytterfeltene i knutepunktene, og siden flere av kontrollene i undergurtene er over 70% utnyttelsen kan igjen diskusjonen om å følge punkt 9.2.1 (1) og 9.2.1 (5) fra Eurokode 5 [63]. Flere av knutepunktene i Grasshopper er over 70% i utnyttelse, og siden det brukes en mer konservativ beregning burde kravet fra Eurokode 5 følges i Grasshopper modellen, selv om sammenligningene ville vært forskjellige. Grunnen for at det ikke ble fulgt var for å gjør en sammenligning med Sweco sine dimensjoner og utforming av fagverket.



Figur 8.11: Sammenligning kontroll i undergurt

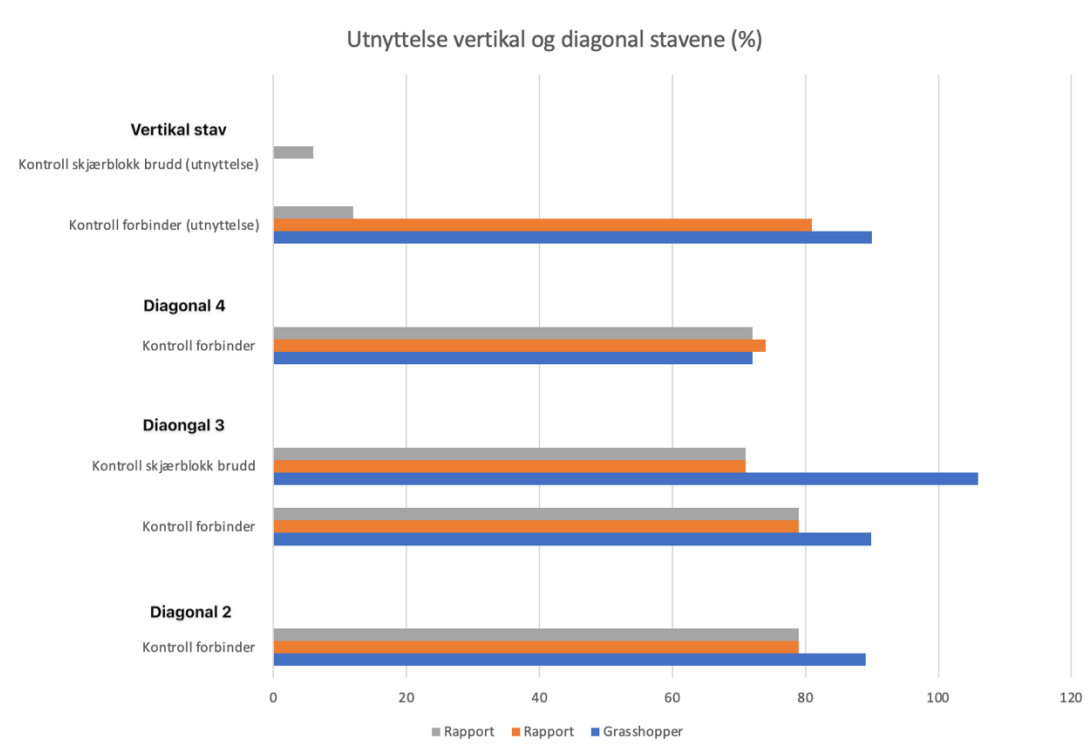
Vertikale og diagonale staver

I Figur 8.12 er resultatet for utnyttelsen til stavene fra kapittel 7.5 samlet i et diagram. Det vises kun de tilfellene hvor den effektive tykkelsen til ytterfeltene ble brukt. I diagonal 0 og 1 er ytterfeltene like og dermed er resultatene relativt like, bortsett fra forskjellene i belastningskraftene.

Som nevnt i kapittel 7.5 har Sweco utformet knutepunktene som er i samme stav med forskjellige avstander til kant, ender og mellom dyblene. Det er ikke store variasjoner, men dette er noe som ikke kan utføres i Grasshopper-modellen. Det kan være spesifikke forskjeller mellom knutepunktene som er i hver ende av stavene, og dette burde kunne korrigeres i Grasshopper modellen. Problemet blir derimot at flere beregningskomponenter må brukes, noe som igjen gjør Grasshopper modellen treg. Dette indikerer igjen begrensingen med å utføre beregninger med Grasshopper.

Diagonal 3 holder ikke i for skjærblokk brudd på grunn av den store strekkraften i forbindelsen, denne burde blitt justert i henhold til dette. Som diskutert i starten av dette kapitlet er skjærblokkbrudd kontrollen veldig konservativ, og er dermed en stor faktor for at dette ikke holdt.

Knutepunktet i den vertikale staven som ble vist øverst i Figur 8.12, har Sweco gjort en kontroll for lastkombinasjon 3. Det kan igjen påpekes at dette er ikke mulig å undersøke i Grasshopper, noe som bergenser brukbarheten med komponenten som ble programmet.



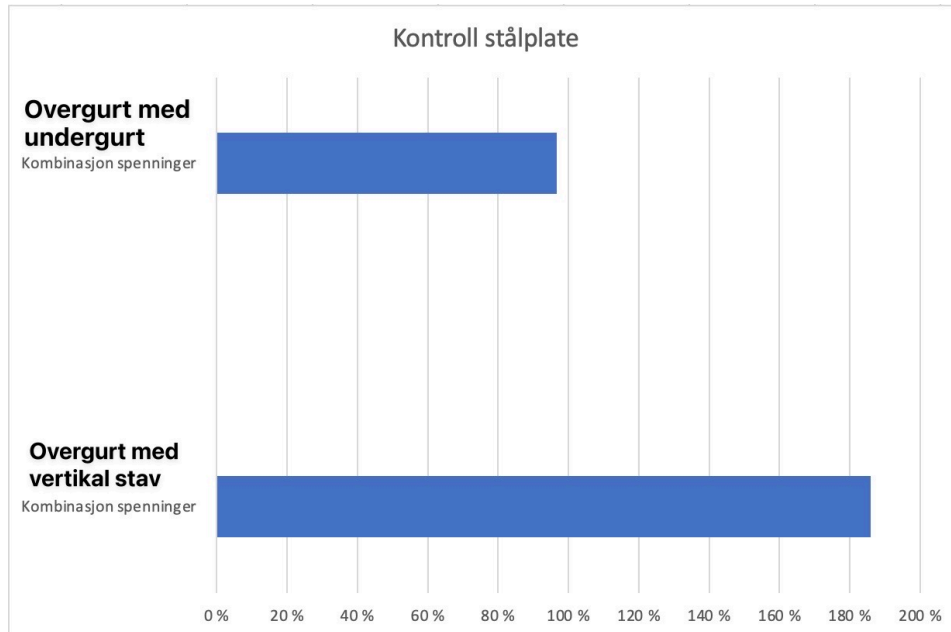
Figur 8.12: Sammenligning kontroll staver

8.3.3 Kontroll av stålplate

I denne oppgaven har ikke kontroll av stålplaten hvert hovedfokus, men det ble bestemt å utføre kontrollen for hullkanttrykk etter EK3 [115] og kontrollene som ble gitt i Limtreboka [58]. Dette kommer av at punkt 8.2.3 (2) i EK5 presiserer at stålplaten skal kontrolleres [63], samtidig vil det være fordelaktig, å kunne direkte se utnyttelsene for stålplatene i samme program som dybel kontrolleres i. Dette gjør at man kan utforske flere utforminger på knutepunktene og direkte se både utnyttelsen til dybelene og stålplaten i henhold til utformingen. Dette vil være fordelaktig for å redusere mengden på stålet i forbindelsene, og til å finne en optimal løsning for forbindelsene.

Resultatet fra kapittel 7.6 viser generelt lav utnyttelse innen de fleste kontrollene for stålplaten. De største utnyttelsene er i kombinasjon av spenninger, der forbindelsen mellom overgurten og undergurten, og forbindelsen mellom overgurten og den vertikale staven har store utnyttelses verdier. Dette er vist i diagrammet fra Figur 8.13, der stålplaten i overgurten med vertikal stav har en utnyttelse på 186%. Formelen for denne kontrollen er en Von Mises spenningsfordelings kriterium, og påpekes å være til sikker side [58]. Det er vanskelig å få en antydning om disse beregningene er

utført på en riktig måte, ettersom de ikke kan sammenlignes med noe. Det kan derimot antydes at det er beregnet feil, ettersom fagverket er bygget med samme utforming i dette knutepunktet. Dette begrenser dermed bruken av denne komponenten og den må sannsynligvis omprogrammeres for å løse dette.



Figur 8.13: De Kritiske utnyttelsene fra stålplate kontrollene i knutepunkter

I limtreboka ble det presisert at knekk lengden kunne kontrolleres etter avstanden mellom dybelene i fiberretningen. Dette utføres i knutepunktene som er i gurtene ettersom dette er den lengste avstanden i disse stålplatene. De forbindelsene med stavene og forbindelsen med over- og undergurten blir derimot den største avstanden i stålplaten mellom knutepunktene. Denne vil kunne knekke før stålet avstanden mellom dyblene knekkes. Dette er noe som ikke er presisert å gjøre i limtreboka, men siden risikoen for knekking i disse forbindelsene vil være større, ble det valgt å utføre dette slik.

Det var også presisert i Limtreboka [58] utføre knekkings kontrollen som en Euler søyle. Dette er noe som virker litt merkelig, ettersom det er en stålplate og ikke en søyle. Stålplatene burde kanskje heller kontrolleres etter plateknekkingskontrollen fra NS-EN 1993 1-5 [171]. Dette er noe som denne oppgaven ikke kan besvare, men basert på at i enkelte tilfeller er slankhetsverdiene ganske store, kan det antydes at det bør utføres dypere FEM-analyser på enkelte stålplater, for å få et innblikk i hvordan stålplaten håndterer belastningskreftene.

8.4 Valg av metode

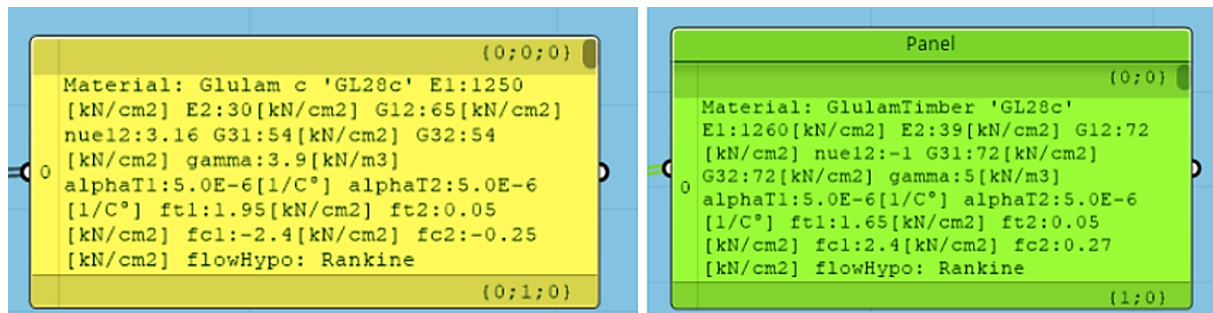
Det kan diskuteres hvorvidt metoden som er valgt er den mest fordelaktige. Metodevalget bygges i stor grad på prøving og feiling, og derfor er det gjort en del observasjoner ved ulike valgmuligheter. I tillegg ville oppbyggingen av Grasshopper-systemet sannsynligvis vært mer effektiv, dersom det ble bygget av en bruker med mer erfaring med programvaren. Det kan også tenkes at metodevalget påvirker kapasitetsmengden som trengs fra datamaskinen.

8.4.1 Materialinnhenting

Beaver (plug-in) ble benyttet for innhenting av materialeegenskaper, hvor materialoverføringen utføres som vist i Tabell 6.1 fra metodekapittelet. For eksempel oversettes verdien for $f_{t,90,k}$ direkte til f_{t2} | *Tensile strength dir 2*. Dette egner seg ikke for ortotrope materialer. Med denne overføringen ville tradisjonelt konstruksjonstrevirke kunne benyttes som større flater, noe som i realiteten ikke er mulig. Dette er også grunnen til at utnyttelsesverktøyet i Karamba ikke fungerer ved Beavers materialinnhenting. Verktøyet vil oppgi at utnyttelsen overskrides, da den sannsynligvis kontrollerer materialet i henhold til Von-Mises. Utnyttelsen skal ikke kontrolleres på denne måten for trekonstruksjoner, og derfor har vi programmert et eget system for å vise utnyttelsen til trematerialet i henhold til EC5 [1].

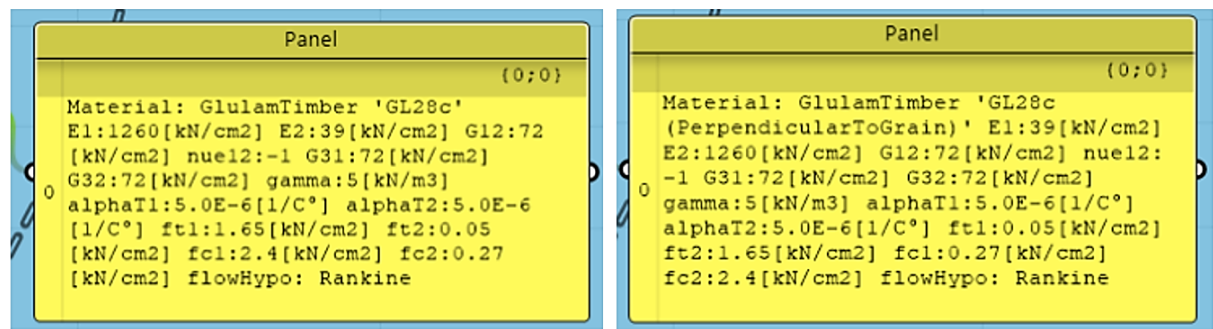
Det er også spesifisert i Karamba3D-manualen, at ortotrope materialeegenskaper ikke skal benyttes på bjelke-elementer. Når ortotrope materialeegenskaper påføres et bjelke-element, vil bare egenskapene i første retning benyttes i analysen. Derimot, vil egenskapene i begge retninger benyttes, dersom strukturen bygges med overflater (shells) [49]. Dette er imidlertid en helt annen analyse-metode, enn den som skal benyttes i dette prosjektet. Likevel kan analysen gjennomføres med materialinnhenting som er benyttet, ettersom FEM-analysen baseres på lineær-elastisk analyse. EK5 presiserer at en kan bruke lineær-elastisk fagverksteori, med de forenklingene det medfører [63]. For beregning av moment-, skjær og aksialkrefter som opptrer på bjelkeelementene, behøves bare verdier for trevirkets E-modul og densitet. Lineær-elastiske analyser tar nemlig utgangspunkt i Hooks lov, og behøver kun materialeegenskaper som påvirker tøyningen i materialet [1].

Etter Karambas siste oppdatering (2.2.0 versjonen), har det også kommet flere metoder for å hente inn materialeegenskapene til limtre. Materialene kan hentes inn via en Karambas «Material Selection»-komponent, og gir mulighet for å velge limtre-materialer fra Karambas bibliotek. Denne metoden er enklere, men ikke pålitelig. Figur 8.14 viser utklipp fra materialinnhenting ved bruk av Beaver i den gule rammen til venstre, samt materialinnhenting ved bruk av Karambas bibliotek i den grønne rammen til høyre. Ved sammenligning, ser man at verdiene ikke stemmer overens med hverandre. Materialeegenskapene som hentes ved bruk av Karambas bibliotek samsvarer ikke med standardiserte materialeegenskaper for limtre. Derfor kan ikke denne metoden benyttes.



Figur 8.14 Materialinnhenting ved bruk av Beaver versus materialinnhenting ved bruk av Karambas bibliotek

Materialinnhenting ved bruk av Karambas bibliotek gir i tillegg mulighet for å velge enten egenskapene parallelt med fiberretning eller vinkelrett på fiberretning. Med dette vendes materialmatrisen, som vist i Figur 8.15. I utklippet til venstre blir egenskapene parallelt med fiberretning valgt som første retning, mens det i høyre utklipp er egenskapene vinkelrett på fiberretning som velges som første retning. Likevel kan Karamba fremdeles ikke påføre egenskaper i flere retninger på bjelke-elementer, og vil bare påføre materialegenskapene i første retning. Ved bruk av egenskapene til høyre, vil derfor egenskapene vinkelrett på fiberretning benyttes i elementets lengdeakse. Så dette kan heller ikke benyttes.



Figur 8.15 Egenskaper parallelt med fiberretning og vinkelrett på fiberretning (fra Karambas bibliotek)

Det er også mulig at det finnes en mer passende metode for materialinnhenting som vi ikke har blitt introdusert for. Men med observasjonene som vi har gjort oss, kan materialinnhenting ved bruk av Beaver anses som et tilstrekkelig alternativ for lineær-elastisk analyse. Selv om denne metoden også har sine begrensinger.

8.4.2 Kontroll i henhold til Eurokode

Som forklart i metode-kapittelet, ble det også forsøkt å benytte Beaver for å utføre kontroller i henhold til eurokode. Plug-in verktøyet har komponenter for utnyttelseskontroller, samt modellering- og beregning av knutepunkter. Disse har imidlertid mange begrensninger, som tidligere forklart. Ved forsøket som ble utført med Beavers utnyttelses-kontroll, ble modellen i tillegg utrolig ressurskrevende for datamaskinen. Selv en modell uten beregning- og modellering av forbindelser ble uhåndterlig for maskinen.

I løpet av arbeidsperioden, er det bare Beaver som er observert å ha funksjoner for eurokodekontroller av trekonstruksjoner. Karamba er også langt bedre tilpasset stålkonstruksjoner, med tanke på materialinnhenting som ble beskrevet ovenfor. Grasshopper har derfor generelt store forbedringspotensialer når det kommer til bruk av tre som byggemateriale.

8.4.3 Ressurstung programvare

Det fullstendige systemet som er etablert, er nokså ressurstungt for datamaskinen. Dette setter store begrensninger for effektiviteten ved bruk av systemet. For å f.eks. få tilgang til å endre ett av navnene til en komponent, vil dette kunne ta opp imot 10 sekunder. Dersom det skal gjøres endringer i de mest ressurskrevende clusterene, vil det å lagre endringen kunne overskride 5 minutter. Dette oppleves som særlig ineffektivt, da det ikke kan utføres andre oppgaver i modellen samtidig som clusterene lagres. Heldigvis, er det mulighet for å utsette lagring av clusterene så lenge modellen er åpen. Maskinene som er benyttet i prosjektet har begge i7 prosessorer. I tillegg har programmet også blitt testet på en av universitetets maskiner med i9 prosessor. Likevel oppleves systemet som like saktegående. I tillegg kan det se ut til at programmet stiller store krav til minne (RAM) og skjermkort (GPU), ved analysering av maskinens ytelse ved bruk av programmet.

Det er mest krevende for maskinen å justere og tilpasse clusterene med knutepunkts beregninger. Dernest kommer clusteren med utnyttelseskontroller i bruddgrensetilstand. Begge disse clusterene er fylt med «Expression»-komponenter med matematiske funksjoner. Det kan tenkes at det ville krevd mindre fra datamaskinen, dersom disse beregningene heller ble programmert i script-komponenter. Selv om visuell programmering er en enklere metode for de som ikke innehar avanserte programmeringsferdigheter, er det sannsynligvis relativt enkelt å programmere rene matematiske funksjoner i script. Antagelig ville metoden som ble benyttet for lastkombinasjonene i modellen, også kunne fungere for knutepunkts beregninger. Det er også observert, at dette har blitt utført i tidligere studier [5]. På den andre siden er det enklere å kunne holde oversikt over beregninger som utføres med visuell programmering. Det blir lettere å selv kunne oppdage feil, samt for kollegaer å sette seg inn i systemet og kontrollere det.

Det kan også diskuteres om «Telepathy»-komponentene krever ekstra datamaskin-kapasitet. Disse benyttes imidlertid for å lage et ryddigere system, og modellen ville ikke være særlig brukervennlig uten disse.

8.5 Brukbarhet

Forskningen viser at visuell programmering er enkelt og lett forståelig for de fleste, sett i forhold til tradisjonell programmering [36, 37]. Også egne erfaringer fra bruk av C#, Python og Grasshoppers visuelle programmering, kan bygge opp under denne beskrivelsen. Utførelse av parametrisert design vil imidlertid kreve en tankeomstilling for å bygge gjensidige avhengigheter og andre sammenhenger, som artikkelen i kapittel 3.7.1 diskutert [8]. Derfor vil det sannsynligvis kreves litt opplæring for å kunne utvikle og bygge systemet videre. Men siden alt programmeres visuelt, vil det ferdige systemet være relativt enkelt å forstå og gjenbruke. Særlig med god beskrivelse av hva som er gjort, og hvorfor. Derfor vil de aller fleste konstruktører sannsynligvis kunne bruke Grasshopper-systemet fra denne oppgaven, etter relativt kort opplæring.

8.5.1 Gjenbruk

Slik systemet er bygget per i dag, kan man enkelt justere antall staver, høyde, lengde, plassering av opplegg, tverrsnitt og materialkvalitet. Lastkombinasjonene kan også justeres, men krever litt ekstra programmering ved fjerning eller tilføyning av lastkombinasjoner. Likevel, kan det plutselig dukke opp

defekter ved justering av parametere, som man ikke har lagt merke til tidligere. Disse kan stort sett repareres effektivt, da de er enkle å se årsaken til. For eksempel, kan en linje i knutepunktsmodelleringen være lang nok i et tilfelle, men for kort i et annet. Da må linjen bare korrigeres så den blir lang nok for alle tilfeller. Slik utvikles systemet til det bedre, desto mer det brukes. I tillegg bygger man seg raskt erfaringer for hvordan systemet kan bygges mer funksjonelt og effektivt. Ved analysing av tidligere arbeid, vil man (med litt erfaring) raskt legge merke til mulige korrigeringer som kan forbedre funksjonaliteten.

En del av systemet som vi i ettertid ville utført annerledes, er scriptet for lastkombinasjoner. Det er noe komplisert å justere lastkombinasjonene i scriptet. En bedre metode, ville vært å hente verdiene direkte fra Excel, på samme måte som materialeegenskapene. I Excel kunne man bygget et langt mer oversiktlig system, som er enkelt for de fleste å gjøre endringer i. Variasjon i antall lastkombinasjoner må i tillegg korrigeres for i systemene for både kontroll av bruksgrensetilstand og ved overføring av krefter til knutepunkts-beregninger. Dette burde være automatisert. En bør alltid etterstrebe reduksjon av denne typen kompleksitet, da dette reduserer potensiale for andre kollegaer å bruke systemet [1].

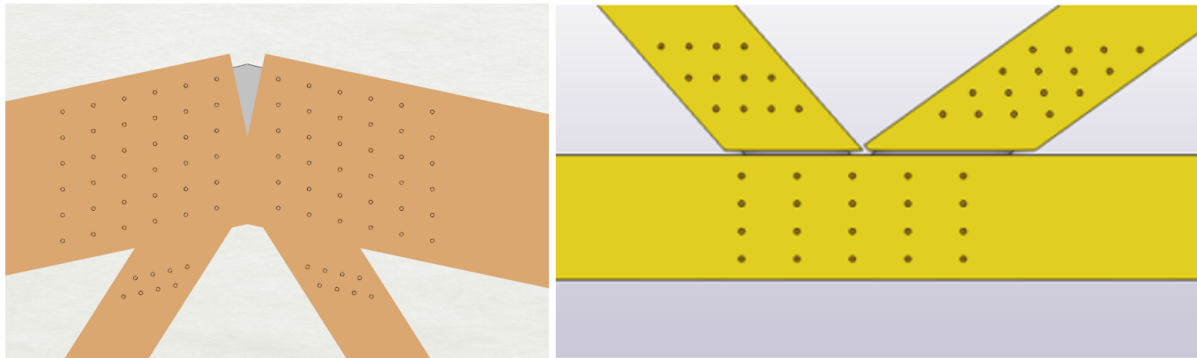
Videre bør en RIB-bedrift benytte standardiserte FEM-analyse programmer som er pålitelige [1]. Et forslag som kan gjøre systemet mer gjenbrukbart, er derfor å hente moment-, skjær-, og aksialkrefter fra en annen programvare. Grasshopper har mange muligheter for overføring til andre programvarer, og kan muligens samarbeide med Robot, Ansys, RFEM og lignende. Også innspill fra Sweco tilsier at det er systemene for modellering- og beregning av knutepunkter som ville vært mest hensiktsmessig for de å benytte seg av. Ved å bygge en enkel pinnemodell (som ikke er parametrisk), samt innhenting av kreftene fra en annen programvare, vil en kunne benytte seg av knutepunktsystemene uten hele systemet utenom. Parametrisk utforming av selve pinnemodellen, er muligens ikke det mest hensiktsmessige for RIB-bedrifter uansett. Utformingen til strukturen er såpass forskjellig fra prosjekt til prosjekt, at det ikke lønner seg økonomisk å bygge parametrisk fra gang til gang. Dette kommer blant annet av kravene som stilles fra arkitekter. Dette underbygger også diskusjonen fra Edmonds fagartikkel, om at det parametriske designet i noen tilfeller vil ha unødvendig kompleksitet, og dermed også øker kostnadene i prosjektet [8].

Parametrisk design for oppbygging av pinnemodellen, passer for tilfeller hvor strukturen benyttes igjen og igjen, med noen få justeringer fra gang til gang. Derfor kan det tenkes at systemet er bedre egnet for takstolprodusenter, fabrikker og lignende.

8.5.2 Forbedringspotensialer

Enkelte detaljer i systemet har forbedringspotensialer som er mulig å korrigere for, men som på grunn av tidsbruk og begrenset datakapasitet ikke har latt seg gjøre. For det første benyttes Karambas automatiske fremstilling av tredelene. Denne vil ikke kappe elementene riktig i forhold til hverandre, da den bare tar utgangspunkt i elementets systemlinje og bygger rektangulære volumer basert på denne. Et eksempel på denne problemstillingen er vist til venstre i Figur 8.16. For det andre er det ønskelig å se toleransen mellom tredelene, som vist i oppdragivers modell til høyre i Figur 8.16. For å oppnå et bedre resultat her, må man antagelig etablere fremstillingen av tredelene selv, ved hjelp av «surface» komponenter og lignende. Det kan også muligens eksistere plugin-verktøyer

som har egne komponenter for dette. På tilsvarende vis, kan det også hende at stålplatene innehar detaljer som ikke nødvendigvis er byggbare.



Figur 8.16 Problematikk knyttet til hvordan tredelene møtes

I tillegg er det valgt å kun utføre én knutepunkts-beregning for hver stav. Dette medfører at både dybelgruppe, stålplatetykkelse og tykkelse på tredelens innerfelt må være likt i begge ender av staven. Beregningene vil likevel være riktige, da kraften som påføres staven vil være lik i begge ender. Det er heller ingen eksentrisitet i disse gruppene, og de vil dermed ikke påvirkes av forskjellen på over- og undergurtens vinkel. Likevel, er det i enkelte tilfeller ønskelig å kunne ha forskjellig geometri i de to dybelgruppene. Det er blant annet observert ulike endeavstander i løsningen fra Sweco. Videre vil det også medføre at alle stålplatetykkelser og innerfelt-tykkelser må være like for alle forbindelser i hele fagverket (med unntak av kryssningspunktet mellom over- og undergurt). Dette er fordi stavene har et gjennomgående siksakk mønster, hvor hver av stavens ender alltid vil være i kontakt med neste forbindelse. Clusterene kan imidlertid kopieres, og vil dermed kunne utføre både beregninger og modellering for dybelgruppene i begge endene hver for seg. Grunnen til at dette ikke ble utført, er fordi bergengings-clusterne er svært ressurskrevende.

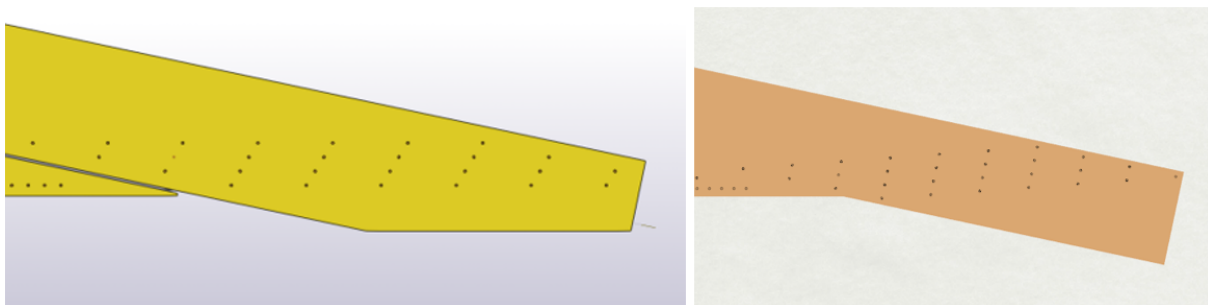
Videre er dybelberegningene bare tilrettelagt for to stålplater. Det må korrigeres i både beregning- og modelleringscluster dersom systemet skal kunne benyttes for flere eller færre stålplater. Heller ikke gurtens skjøter er tatt hensyn til i systemet. Som vist i Figur 8.17, er fagverket i realiteten skjøtet midt på undergurt, og ved overgurtens tilknytning til 3.- og 4. diagonal. For å tilføye dette til systemet, måtte man satt inn én ny cluster med knutepunkts-beregninger for hver av skjøtene. Skjøten i overgurt vil ha strekk på den ene siden og trykk på den andre. Derfor måtte det blitt utført én knutepunkts-beregning for hver av skjøtens sider. Når det gjelder modellering, måtte man laget en ny cluster for skjøten i undergurt, da ingen andre forbindelser har samme utforming. Her kunne man imidlertid tatt utgangspunkt i clusteren for kontinuerlig undergurt (blå cluster), og fjernet modelleringen av stavene. Dette vil ikke være særlig krevende, dersom man er godt kjent med oppsettet. I overgurtsskjøten, kunne man sannsynligvis ha brukt clusteren for toppunktforbindelsen (grønn cluster). Også her måtte det blitt gjort et par tilpasninger, ettersom toppunkts-clusteren tar utgangspunkt i to like staver.



Figur 8.17 Fagverkets skjøter

Ellers kan systemet bare etablere dybelgrupper med likt antall dybler i hver rekke. Som vist til venstre i Figur 8.18, har Sweco benyttet kun 3 dybler i øverste rekke i forbindelsen mellom over- og undergurt. Dette er utført som følger av arkitektkrav, da gurten skal kappes ved enden. Ettersom dette ikke er mulig i Grasshopper-systemet, ble det utarbeidet en annen løsning hvor innbyrdes avstand mellom dyblene reduseres. Denne løsningen er presentert til høyre i Figur 8.18. Resultatet overholder akkurat kant- og endeavstandene, men det vil ikke være mulig å kappe ytterste del av gurten. Dette er derfor en klar begrensning i systemet som krever videre arbeid. Det å gi en løsning som er akkurat innenfor minimumskravene, er heller ikke en ønsket løsning. Vanlig praksis er å overholde minimumskravene med god margin, samt å ikke presisere løsninger på minimumsmarginer [172].

I tillegg er modelleringen av denne dybelgruppen annerledes enn de resterende dybelgruppene. Her foreligger det noen begrensninger knyttet til avstandsstyring. Justering av kant- og endeavstand kan ikke utføres fra utsiden av clusteren, på samme måte som for de resterende modellerings-clusterne. I tillegg gir ikke clusteren for kontroll av avstandskrav riktige resultater, da beregningene ikke har samme forutsetninger som ved de andre dybelgruppene. Systemet bør derfor utvikles videre, og man må være ekstra oppmerksom på at avstandskravene overholdes.



Figur 8.18 Problematikk i dybelgruppe (forbindelse mellom over- og undergurt)

Det at systemet bare kan etablere dybelgrupper med likt antall dybler i hver rekke, skaper også komplikasjoner for dybelgruppen i 4.diagonal. Også her har vi måttet utarbeide en annen løsning, med likt antall dybler i hver rekke. Figur 8.19 viser Sweco sin løsning til venstre, samt løsningen i Grasshopper-systemet til høyre. Løsningen fra Sweco er trukket lengre nedover mot undergurten, for å redusere faren for knekking i stålplaten [1]. Ettersom antall dybler i de nederste rekkene ikke kan reduseres i Grasshopper-systemet, må dybelgruppen starte høyre oppe i elementet for å overholde

kravet til endeavstand. Løsningen er satt til minstekravet for belastet ende, og vil ikke gi fare for knekking av stålplaten ifølge utførte beregninger. Løsningen fra Grasshopper-systemet vil imidlertid gi en større momentarm inn til forbindelsens midtpunkt, noe som skaper mindre momentstivhet i forbindelsen. Samtidig vil løsningen gi en veldig kort avstand mellom de to dybelgruppene i staven, og dette kan øke risikoen for knekking av tre-elementet ut av planet. Dersom tverrsnittet nesten blir slisset hele veien gjennom, blir det muligens feil å betrakte tverrsnittet som et fullt tverrsnitt i knekkingskontrollen. Da er det mer rimelig å betrakte staven som 3 elementer som hver for seg har mulighet for å knekke ut (ett element på hver ytterside av slisseplatene og ett element imellom). Med dette vil det oppstå en risiko for at elementene knekker ut i motsatt retning av hverandre [172]. Denne løsningen underbygger derfor diskusjonen fra Edmonds fagartikkel, om at det parametriske designet i noen tilfeller kan foreslå løsninger som ikke er ideell for byggarbeidet [8].



Figur 8.19 Problematikk i dybelgruppen i 4. diagonal

8.5.3 Dokumentasjon

Enhver RIB-bedrift må kunne dokumentere beregninger og kontroller [3]. Dette er grunnen til at Sweco i utgangspunktet ønsket at alle beregninger ble utført i Excel, for så å føre resultatet tilbake til Grasshopper for optimalisering. Dette ble som nevnt ikke mulig for vårt tilfelle, da denne overføringen satte begrensninger for optimaliseringsverktøyene. Likevel er forsøket som ble utført for overføringer på tvers av programmene, en god indikator på at Grasshopper ikke har begrensende dokumentasjonsmuligheter. Dataoverføringen mellom programmene fungerer godt, ved god kontroll på navnsetting, celle- og filplassering. En bør imidlertid være oppmerksom på å sette verdiene som hentes fra Grasshopper i ett fraskilt ark i Excel-filen. Dermed bør disse inputverdiene overføres til et annet ark innad i Excel-filen, for å unngå overskriving eller annullering av funksjoner som skal være fast. Med denne fremgangsmåten har en ubegrensede dokumentasjonsmuligheter i Excel.

All dokumentasjon kan derfor hentes fra systemet som er bygget, men det må likevel programmeres et system som henter ut dokumentasjonen som ønskes. Dette vil kreve litt tid og arbeidsressurser, og derfor benyttes ofte parametriske design sammen med BIM-verktøyer som har innebygde informasjonssystemer (som beskrevet i kapittel 3.1.2). Begrensingen med dette er at oppdateringer som gjøres i den parametriske modellen, ikke vil oppdateres automatisk i BIM-verktøyet. Derfor må designet være helt fullstendig før det overføres til et BIM-verktøy [3].

8.5.4 Kan man stole på resultatene?

I Edmonds fagartikkel diskuteres risikoen for «Black box»-design, hvor brukeren stoler for mye på systemets resultater [8]. I denne oppgaven er resultatene sammenlignet med Sweco sine resultater, og har dermed blitt grundig sammenlignet og gransket opp mot disse. Dette gjelder imidlertid bare

løsningen med 5 diagonale staver, og med forbindelsesdesignene som Sweco har foreslått. Det kan ikke verifiseres at de andre løsningene som systemet foreslår er pålitelige. Derfor bør kreftene kontrolleres med standardiserte FEM-programvarer, og Eurokode-kontrollene bør undersøkes med andre beregningsprogrammer.

Artikkelen fremhever også at parametrisk design i noen tilfeller vil begrense antall løsninger, og ikke alltid foreslå den beste løsningen. Man bør derfor være kritisk og utfordre resultatene [8]. Som et eksempel, vil denne oppgavens Grasshopper-system kun foreslå W-takstoler, selv om en A-takstol gjerne kunne passet bedre i enkelte situasjoner. Systemet kan selvfølgelig bygges ut til å også gjelde A-takstoler, men poenget er at mulige løsninger alltid vil være begrenset. Derfor bør en være kritisk, og vurdere andre løsninger også. I tillegg vil systemet kreve mer kapasitet av datamaskinen desto mer det bygges ut, noe som begrenser antall løsninger ytterligere.

8.6 Optimalisering av fagverket.

I dette kapitlet vurderes og diskuteres resultatet for optimaliseringen av størrelsen til fagverket. Dette er noe som burde blitt undersøkt mer i oppgaven, men på grunn av at de ikke var til tid å utforske denne delen, blir eventuelle løsninger på problemet med optimaliserings resultatet diskutert.

8.6.1 Optimalisering for størrelse, form og topologi

Resultatet som er presentert i kapittel 7.7.1 er kun en størrelsesoptimalisering av alle tverrsnittene som er i limtrefagverket. Grunnen til dette var at geometrien til modellen i Grasshopper var parametrisk basert på fagverket til Sweco, noe som gjorde det utfordrerne å utforske optimalisering av både formen og topologien til fagverket.

Det hadde vært mulig å utføre formoptimalisering av fagverket, siden høyden, lengden og antall staver kan endres av parametere. Problemet er at det mange krav som må tilfredstilles i limtrefagverk, noe som gjør denne prosessen veldig vanskelig å optimalisere, siden det er så mange krav å ta hensyn til. Det må i tillegg legges til nye element- og knutepunkts kontroller, dersom flere elementer og forbindelser genereres inn i modellen. Dette gjør at det blir flere krav som må tas hensyn til. I tillegg, dersom andre former på elementene skal utforskes, som for eksempel en buet bjelke, må nye Clustere som kan kontrollere dette programmeres. Dette gjelder også for forbindelsene, dersom det er flere enn tre elementer i forbindelsen, må dette det programmeres en ny Cluster-komponenten som kan visualisere og beregne dette. Det er heller ikke sikkert at alle former på en forbindelse med tre elementer kan visualiseres og beregnes av Cluster-komponenten som ble programmeret i denne oppgaven. Dette er noe som begrenser mulighetene for å form- og topologi-optimalisere med komponentene som er programmerte i denne oppgave. Samtidig til flere beregninger og visualiseringer som utføres i Grasshopper til tregere blir modellen.

For at det skal kunne utføres en form- og topologioptimalisering, må modellen programmeres på en annen måte. Flere strukturelle konsepter av fagverket må programmeres inn i Grasshopper-modellen, og deretter programmert et system som gjør det mulig å utforske dem. Dette er noe som ble utført i forskningsartikkelen om Beaver, som ble presentert i kapittel 3.7.3 [6]. I vårt tilfelle ble

modellen laget helt fra grunnen av, og oppbyggingen ble dermed veldig kompleks. Dersom fagverket hadde blitt programmert på nytt, ville det litt brukt andre metoder som kunne gjort det mulig å formoptimalisere strukturen.

8.6.2 Optimaliseringen av limtrefagverket

I størrelsesoptimaliseringen av tverrsnittene til limtrefagverket ble Opossum brukt som optimaliseringsverktøy i Grasshopper, siden dette verktøyet kunne ha flere enn ett formål. En begrensning med denne optimaliseringen er at det var flere krav å ta hensyn til, enn objekt-formål som kunne brukes av Opossum. I fagverket er det totalt ni elementer som har tretten kontroller. I tillegg er det to knutepunkter i hver stav, ti knutepunkter i overgurten og åtte knutepunkter i undergurten, som igjen har totalt ni kontroller hver. Hver kontroll vil ikke være utslagsgivende, men det er likevel mange krav å ta hensyn til og optimaliseringsverktøyet kan ikke ta hensyn til alle krav.

Det ble bestemt å maksimalt ha fire formål som Opossum skulle optimaliserer tverrsnittstørrelsene etter, siden til flere formål, til verre blir det for optimaliseringsalgoritmen. I tillegg ble det valgt å optimalisere et element om gangen, og velge kontrollene som hadde de størst utnyttelse fra resultatet presentert i kapittel 7.4 som optimaliserings formål. Dette var noe som fungerte for overgurten, ettersom sammenligningen mellom resultatet fra element analysen i kapittel 7.4.1 og knutepunktsanalysene i kapittelet 7.5 og 7.6 viser at elementkontrollen av overgurten er det mest kritiske. For stavene og undergurten derimot fungerte dette veldig dårlig, ettersom alle måtte korrigeres for å tilfredsstillte utnyttelsene i knutepunktene.

Ved å sammenligne elementanalysene fra kapittel 7.4 og knutepunktsanalysene fra kapittel 7.5 for de diagonale og vertikale stavene, viser resultatet at knutepunktene er det mest kritiske. Tverrsnittstørrelsen skulle dermed blitt optimalisert i henhold til utnyttelsen av knutepunktene og ikke elementanalysene som ble utført. Dette ville mest sannsynlig gitt et bedre resultat for optimaliseringen med Opossum av stavene.

Problemet med optimaliseringen av undergurten, var at bredden ble mindre enn bredden til diagonal 3. Resultatet fra elementkontrollen og knutepunktene i undergurten, viser at elementkontrollen var mest kritisk for undergurten. Tverrsnittstørrelsen Opossum generert var optimal for undergurten, problemet var modellen ikke er programmert slik undergruten ikke kan ha mindre bredde enn stavene. Dette gjorde at optimaliseringsresultatet måtte korrigeres. Noe som kan løse dette er å bruke en parameter, der minsteverdien for bredden til undergurten er større enn bredden til stavene.

Det ble valgt å bruke en ny Grasshopper-fil for selve optimaliseringen av tverrsnittene, der flere av knutepunkts kontrollene ble fjernet fra modellen. Grunnen til dette var å effektivisere modellen, siden den var veldig treg med alle knutepunkt kontrollene. Dette gjorde at valgene som ble valgt fra optimaliserings løsningene til Opossum, ble kun valgt basert på utnyttelsene til elementene og ikke knutepunktene. Ikke før tverrsnitt størrelsene ble lagt inn i modellen med alle knutepunkts kontrollene ble det bemerket at stavene ikke holder i knutepunktene, deretter ble modellen korrigeret. Dette en begrensning for Grasshopper når det kommer til optimalisering. Grasshopper hadde en veldig treg respons når det var flere tusen objekter som utfører spesifikke oppgaver, noe

som dermed begrenser optimaliserings mulighetene. Dette er en av grunnene til at det ikke ble brukt mer tid på å optimalisere tverrsnittene.

8.6.3 Problemer og eventuelle løsninger av optimaliseringen

Et problem med optimaliseringen, var at Opossum søker etter den minste utnyttelsen. Dette gjør at løsninger som gir bedre utnyttelse av elementene ikke vurderes, ettersom den søker etter de laveste utnyttelsesverdiene og vil dermed ikke vurdere flere løsninger som for eksempel har utnyttelse på 70%. Det var ikke mulig å la Opossum generere løsninger for en bestemt verdi, og i tillegg var ikke Cluster-komponentene for både knutepunktene og elementanalysene programmert slik at det kan hindre optimaliseringsverktøy i å generere løsninger for over 100% utnyttelse. En løsning som kan gjøre for at Opossum skal finne flere løsninger innenfor en bestemt utnyttelse, er å programmere inn i Cluterene at dersom utnyttelsen blir mindre enn den ønskete utnyttelsen, legges det til en stor verdi. Dette er noe som kanskje kan fungere når Opossum søker etter de minste verdiene for formålene. Dersom Opossum skal søke etter utnyttelse som er rett under 100%, kan det i dette tilfellet programmere inn at når utnyttelsen går under 100%, reduseres verdien til et veldig lavt tall.

En annen mulighet som gjøre at Opossum kan finne flere løsninger for en bestemt utnyttelse, er å fjerne variabler fra parameteren som gir for lave og for store utnyttelser. Dette gjør at den kan lettere finne den optimaliserte løsningen for det bestemte elementet. Spørsmålet her er om det vil være mer effektivt å undersøke dette manuelt. Ettersom modellen er parametrisk, og vil dermed direkte gi et resultat basert endringene som gjøres. Selv om Opossum kan utføre flere iterasjoner, viste resultatet fra optimaliseringen at manuell endring på tverrsnittene ga det beste resultatet. I dette tilfellet burde det ha blitt utført flere forsøk på optimaliseringen av fagverket, slik at en gunstig måte for optimaliseringen av fagverket hadde blitt funnet.

Før størrelsesoptimaliseringen ble tverrsnitt parameterne omgjort for å både fjerne antallet på variablene, og for å skape mer reelle limtrebjelker som skal følge standard tverrsnittstørrelser. Dette fungerte for tverrsnitt høyden til elementene, ettersom limtrebjelker økes med 45 mm i høyden. For bredden derimot var dette mer utfordrende. I realiteten vil bredden til limtrebjelker bestemmes av antallet på limtre-lameller som limes sammen, og siden limtre-lameller har en bredde på 90 mm, 115 mm og 140 mm, vil ikke systemet som ble programmert i Grasshopper skape reelle bjelker. Dersom det hadde blitt funnet en løsning som gjør at optimaliseringsverktøy kan finne løsninger basert på reelle limtre størrelser, ville dette redusert antallet på variablene som er i parameterne, og i tillegg skapt reelle tverrsnitt i fagverket.

Et annet problem med tverrsnittbredden, er at parameteren som bestemmer bredden til tverrsnittet ikke henger sammen med parameteren som bestemmer avstanden mellom stålplatene. Dette gjør at det oppstår et problem når stavene eventuelt skal optimaliseres i henhold til utnyttelsen til knutepunktene. I realiteten henger avstanden mellom stålplatene sammen med standard breddestørrelse for limtre. Stålplatene plasseres fortrinnsvis i skjøten mellom lamellene før de limes sammen, noe som er med fordelaktig enn å frese ut sporet i etterkant. Det må dermed programmeres inn en måte som gjør at oppbyggingen av limtrebredden med antall lameller, henger sammen med tykkelsen på innerfeltet mellom stålplatene og dimensjonen til slissene. Dette er noe som også kan

redusere mengden på variabler i Grasshopper-modellen, og noe som kan gjøre det enklere å finne en optimal løsning.

En mulig løsning som kan ta hensyn til flere krav samtidig og i tillegg ha færrest mulig formål for optimaliseringsverktøyet, er å summere alle utnyttelses verdiene og vekten til fagverket. Deretter kan optimaliseringsverktøyet finne de minste verdiene av dette. Dette ble utført i masteroppgavene som ble gjennomgått i kapittel 3.7.5, og burde blitt forsøkt i denne oppgaven. Fortsatt vil det være utfordrerne å ta hensyn til alle kontrollene som er for limtre. Derimot, vil et forsøk som summerer utnyttelsene av elementkontrollene for over- og undergurtene, og knutepunkts kontrollene for stavene sammen med vekten, og som også fjerner variablene fra parameterne som skaper urealistiske tverrsnitt i fagverket, har potensiale til å optimalisere hele strukturen på en gang. Det vil fortsatt være viktig å undersøke resultatene for alle utnyttelsene, og ikke få et «black box»-design, slik som neves i forskningsartikkelen for i kapittel 3.7.1 [8].

8.6.4 Sammenligning av optimaliseringsresultat med case-fagverket.

Resultatet for optimaliseringen viste at det beste var å optimalisere fagverket manuelt. I tillegg viser dette resultatet med det som ble nevnt i kapittel 3.7.1, der Edmonds never at til flere parametere som er i modellen til vanskeligere er det å finne en optimal løsning, noe som var tilfellet i optimaliseringsresultatet, siden det ikke ga et ønsket resultat. Vekten til den korrigerte fagverksmodellen, kunne sikkert blitt reduserte mer, men dessverre var det ikke tid til å gjøre en større forskning innenfor optimaliseringsdelen. Ved å sammenligne resultat fra optimaliseringen og Sweco fagverkt, viser resultatet at løsningen til Sweco er en optimal løsning, og en stor reduksjon av vekten ville ikke vært mulig.

Ved å sammenligne tverrsnittstørrelsene til det korrigerte fagverket, og det fagverket fra casen, er de største forskjellene bredden og høydene på stavene. I case-modellen er bredden på stavene større og høydene er mindre enn den korrigeret modellen. Det er også samme høyde på diagonal stavene 0, 1, 2 og 3. I tillegg er det samme bredde på de vertikale stavene og diagonal stavene 2, 3 og 4. Dette er ikke i den korrigerte modellen, der er alle stavene tilfeldige, noe som begrenser byggbarheten til korrigeret modellen.

Det er nødvendigvis ikke ønskelig å redusere limtrevekten, ettersom det vil være ønskelig å ha litt tyngde på konstruksjonen. Tre er et materiale som har lav densitet i forhold til stål og betong. Det vil dermed være ønskelig å øke vekten, slik at konstruksjonen kan være mer motstandsdyktig mot belastningene. I tillegg til større mengde av tre som brukes til bedre vil de være for miljøet [9]. Det som vil være mer ønskelig å redusere er mengden på stålet i forbindelsene. Dette vil være fordelaktig både for miljøet og for den økonomiske delen. I Grasshopper-modellen som programmert, er det enkelt å prøve forskjellige konfigurasjoner av avstander og antallet av dybler i hvert knutepunkt. Dette gjør at man kan manuelt finne en løsning som reduserer stål mengden i konstruksjonen, men dette kan være utfordrerne. Ettersom man må ta hensyn til begge knutepunktene i stavene, og antall dybler bestemmes av kun to parametere i hvert knutepunkt.

8.6.5 Sammenligning med andre optimaliseringer

Selv om flere studier fra bakgrunnskunnskapskapittelet, viste tilfeller hvor optimaliseringsalgoritmer hadde redusert vekten til fagverket [99, 104], er det vanskelig å kunne sammenligne et resultat med dem. Dette kommer av at disse studiene hadde brukt stål som materiale for fagverkene, og dette materialet har andre egenskaper og krav enn det limtre har. I tillegg virker det som at Karamba3D er mer egnet for å parametriske designe fagverkt stål enn limtre. I studien som er presentert i kapittel 3.7.43.7.4, ble Karamba3D også brukt for strukturell analyse. Her ble vekten som vises i Karamba3D, sammenkoblet med optimaliseringsalgoritmer, og basert på resultatet for denne studien, virker det som at optimalisering med stål er mye mer egnet for Karamba3D [99]. Denne antagelsen forsterkes basert på YouTube videoer som ble sett i opplæringsfasen av Grasshopper. I tillegg har Karamba3D en komponent som velger kan optimaliserer både type materiale og størrelsen på stålelementene. Denne fungerte ikke når tre ble brukt som materialet, men dette ble prøvd i starten av prosjektet når det var flere problemer med modellen, og en tidligere versjon av Karamba3D ble brukt.

8.6.6 Algoritmen for optimalisering

Opossum hadde flere typer algoritmer for optimaliseringen av størrelsen til tverrsnittene. I denne oppgaven er ikke optimaliseringsalgoritmer grundig undersøkt, og RBFOpt-algoritmen ble brukt for resultatet av optimaliseringen. Resultatet ville mest sannsynlig vært annerledes dersom et annen algoritme hadde blitt brukt, og dersom samme algoritme hadde blitt brukt igjen. Optimaliseringen gir ikke den beste løsningen, men den vil gi en optimal løsning på problemet.

Grunnen for at resultatet fra selve optimaliseringen ikke var en god løsning, kommer ikke av at optimaliseringsalgoritmen var dårlig. Optimaliseringsvektøyet og algoritmen gjør optimalisering i henhold til variablene og objekt-formålene som den får. Problemet er at limtre-fagverket som er programmert i Grasshopper ikke er programmert for å kunne gi et reelt resultat. Det vil dermed være vanskelig å sammenligne resultatet med en annen algoritme, når resultatet mest sannsynlig ikke vil fungere.

Flere av de potensielle løsningene som ble diskutert i kapittel 8.6.3 for å forbedre optimaliseringen, påpeker at det er bedre å parametriske designe for at optimaliseringsalgoritmene skal kunne optimalisere fagverket, enn at optimaliseringsverktøyet skal ta hensyn til det som er programmert. Dersom disse løsningene hadde fungert for størrelses optimaliseringen, ville sammenligningen av flere typer algoritmer vært veldig aktuelt.

9 Konklusjon

Her trekkes konklusjonene basert på resultat og diskusjon. Oppgavens forskerspørsmål er: «*Hvordan utfordrer visuell programmering i Grasshopper den tradisjonelle metoden for prosjektering av limtrefagverk?*». Følgende punkter besvarer problemstillingen og dens underspørsmål:

- Ved bruk av visuell programmering i Grasshopper kan det bygges ett helhetlig, integrert system som fjerner unødvendige manuelle prosesser ved overføring mellom programvarer. Systemet vil også minske sannsynligheten for skrivefeil, da alle overføringer skjer automatisk.
- Ved sammenligning av dimensjoneringsresultatene fra Grasshopper mot tradisjonelle beregninger, ble det funnet avvik. Dette kommer av at kreftene fra FEM-analysen i plug-in programvaren «Karamba» gir uregelmessige resultater. Observasjonene tyder på at Karamba er tilrettelagt for bruk på stålkonstruksjoner, og at den derfor har manglende evner til å håndtere ortotrope materialer (som tre). Avvikene kan også indikere at programvaren har redusert potensiale for påføring av skjevt fordelte laster, ettersom avvikene er særlig tydelige ved disse lastkombinasjonene. Avviket fra kreftene vil påføre følgefeil videre i systemet for kontroll av både bruddgrensetilstand og knutepunkter.

Videre er det bevisst benyttet andre formler i knutepunksberegningene når det gjelder effektiv lengde av dybelen i tykke ytterfelt. Også dette gir avvik opp mot Sweco sine tradisjonelle beregninger. Dessuten er systemet bygget feil for utførelse av bruddgrensekontroll for knekking i gurtene, da det benytter både maks moment- og trykk av alle gurtens deler. Her burde hver av delene betraktes for seg selv. I tillegg har beregning av stålplatekontroller vært en nedprioritert del av oppgaven, og disse beregningene kan se ut til å gi feil i dimensjoneringsresultatene.

- Bruken Grasshopper kan være fordelaktig for optimalisering av limtre-fagverkets tekniske utforming. Ved bruk av parametrisk design, vil systemet tilrettelegge for å enkelt kunne prøve forskjellige løsninger ved å variere nøkkelparametere. I tillegg kan det benyttes optimaliseringsverktøy, men da må modellen programmeres slik at man kan få full utnyttelse av disse. Tre er et materiale med mange krav og hensyn som må tilfredsstilles, og derfor må det undersøkes måter som gjør det mulig å optimalisere strukturen i henhold til dette.
- Visuell programmering og parametrisk design kan effektivisere dimensjoneringsprosessen for takstoler i tre, spesielt for bruk i gjentakende prosjekter. Visuell programmering er også ett usedvanlig brukervennlig programmeringsspråk, og derfor vil opplæringstiden for bruk av programvaren være kort. Dette gjelder særlig dersom nybegynnere bare skal benytte ett ferdig etablert system.

Fra et RIB-perspektiv vil enkelte deler av systemet være mer effektive enn andre. Det at overgangene mellom FEM-analyse og kontroller går automatisk, samt at knutepunktene modelleres automatisk basert på beregningene, sees på som særlig fordelaktig. På den andre siden, kan prosessen med å bygge selve pinnemodellen parametrisk være ineffektivt ved enkelte prosjekter. Dette vil påføre prosessen unødvendig kompleksitet, og vil derfor ikke være lønnsomt dersom systemet ikke skal gjenbrukes.

En stor begrensning for effektiviteten, er at systemet er såpass saktegående. Dette gjelder i alle fall ved bruk av datamaskinene som er benyttet i dette prosjektet.

10 Anbefalinger

I dette kapittelet presenteres anbefalinger til oppdragsgiver og eventuelle andre brukere av Grasshopper-systemet. Først presenteres anbefalinger for bruk av systemet, og deretter presenteres anbefalinger for videre utvikling av det. Til slutt presenteres andre anbefalinger, som f.eks. i hvilke prosjekter det lønner seg å bruke parametrisk design.

10.1 Anbefalinger før bruk av systemet

Programmet bør prøvekjøres på en kraftig datamaskin, da det fullstendige Grasshopper-systemet er nokså ressurstungt. Ved bruk av egne PCer og universitetets datamaskiner, med henholdsvis i7- og i9 prosessorer, er systemet såpass saktegående at det setter store begrensninger for effektiviteten. Det er avgjørende for brukbarheten av systemet, at dette fungerer bedre ved bruk av en kraftigere maskin.

I tillegg er det ikke verifisert at alle løsninger som systemet foreslår er pålitelige, da systemet bare er sammenlignet opp mot Swecos løsning med 5 diagonale staver. Ved sammenligningen ble det også funnet avvik i resultatene fra FEM-analysen. Det bør derfor utføres mer feilsøking i modellen for å finne eventuelle årsaker til feilkildene, samt verifisere løsningene som blir gitt ved andre utforminger. I tillegg har kontrollberegningene for stålplatene vært en nedprioritert del av prosessen, og disse beregningene bør derfor kontrolleres grundig.

10.2 Anbefalinger ved bruk av systemet

Ved bruk av systemet bør kreftene kontrolleres mot standardiserte FEM-programvarer, og Eurokode-kontrollene bør sammenlignes med andre beregningsprogrammer. På denne måten kan man bekrefte at systemet stemmer. Man bør også være oppmerksom på at parametrisk design og optimaliseringsverktøy vil gi et begrenset antall løsninger, og at man derfor bør være kritisk og vurdere andre designløsninger.

10.3 Anbefalinger for videre utvikling av systemet

For videre utvikling av systemet bør en først og fremst undersøke om systemet har defekter som har blitt oversett under oppbyggingsprosessen. I tillegg bør systemet korrigeres og videreutvikles med hensyn til forbedringspotensialene som er beskrevet i kapittel 8.5.2. Dette gjelder:

- Fremstilling av tredelene.
- Utførelse av knutepunkts-beregninger i begge ender av stavene.
- Tilrettelegging for flere stålplater.
- Implementering av skjøter i gurtene.
- Tilrettelegging for ulikt antall dybler i hver av dybelgruppene rekker.
- Videreutvikle modellerings-clusteren for knutepunktet mellom over- og undergurt.

Dessuten kan metoden for definering av lastkombinasjoner utbedres (som beskrevet i kapittel 8.5.1), samt gjøres mer fleksibelt for variasjon av modifikasjonsfaktor k_{mod} og antall lastkombinasjoner. Videre kan det være verdt ett forsøk å programmere clusterne for bruddgrensetilstands- og knutepunktskontroller i script-komponenter, for å se om dette vil kreve mindre maskinkapasitet.

Samtidig kan systemet videreutvikles med tanke på dokumentasjon, hvorav Grasshopper har gode overføringsmuligheter via Excel og lignende. Det vil også være en enkel sak å beregne materialmengder og kostnad av materialene, dersom dette er ønskelig.

I tillegg bør optimaliseringsverktøyene undersøkes ytterligere, både med tanke på form, topologi og størrelse. Modellen som ble programmert var ikke egnet for utforskning av optimaliserings metoder med algoritme-basert optimaliseringsverktøy. Det er flere krav som må tas hensyn til, noe som gjør det utfordrende å optimalisere strukturen. Det hadde vært interessant å utforske flere metoder som kan ta hensyn til dette. Komponentene som ble programmet, har mulighet til å hjelpe til med både form- og størrelsoptimalisering, men Grasshopper-modellen og komponentene må omprogrammeres for at de kan brukes i en topologioptimalisering. Det trengs dermed videre arbeid for å utforske dette.

Det ble ikke utført noen optimalisering av knutepunktene, og er dermed noe som å undersøkes videre. Tverrsnittstørrelsen til stavene bestemmes basert på utnyttelsene til knutepunktene, og hadde vært interessant å utforske muligheter for at optimaliseringsverktøy kan finne optimale tverrsnittstørrelser med å bruke antall dybler som variabler.

Avslutningsvis kan systemet bli noe mer brukervennlig dersom resultatene fra kontrollene er mer synlige, for eksempel ved at de lyser rødt ved overskridelse. Dessuten vil systemet bli enklere for andre å benytte, dersom en oversiktsramme plasseres øverst i systemet. Oversiktsrammen bør i den anledning inneholde beskrivelser av hvilke plug-ins som benyttes, hvilke excel-ark som må tilkobles osv.

10.4 Andre anbefalinger

Man bør også være oppmerksom på at bruk av parametrisk design egner seg for bruk i gjentakende prosjekter. Dersom det er ønskelig å bygge lignende systemer for andre typer strukturer, bør man vurdere om systemet skal kunne gjenbrukes. Dersom strukturen er svært forskjellig fra prosjekt til prosjekt, lønner det seg ikke å bygge parametrisk fra gang til gang. Parametriske design kan i noen tilfeller gi unødvendig kompleksitet.

En eventuell ombygging av systemet, er å hente moment-, skjær-, og aksialkrefter fra standardiserte FEM-analyse programmer istedenfor å benytte Karamba. Dette er et tiltak som kunne gjort systemet mer gjenbrukbart for RIB-bedrifter som Sweco, da det i hovedsak er systemene for modellering- og beregning av knutepunkter som de betrakter som effektiviserende tiltak.

11 Referanser

- [1] K. Nodeland Frøysnes, "Samtale (veiledning for masterprosjekt)," Sweco, 18.01.2023 - 09.05.2023, Unpublished.
- [2] W. Jabi, *Parametric Design for Architecture*, 1st ed. London: Laurence King Publishing, 2013. [Online]. Available: https://books.google.no/books?hl=no&lr=&id=I_UgEAAAQBAJ&oi=fnd&pg=PT9&dq=why+us+e+parametric+design+&ots=pnM55O49IH&sig=6SeVtq0OLmAFyf83C7L4g1HPxQM&redir_esc=y#v=onepage&q=why.
- [3] G. Gudnason and R. Scherer, *eWork and eBusiness in Architecture, Engineering and Construction*, 1st ed. Boca Raton, London, New York, Leiden, Reykjavík: CRC Press/Balkema Taylor & Francis Group, 2012. [Online]. Available: https://books.google.no/books?hl=no&lr=&id=z1z83fcH4zMC&oi=fnd&pg=PA453&dq=parametric+design+building+information+modeling&ots=T5T_Omzj5V&sig=IWtnAIzX68IRBg6uEL9QpYvMvwg&redir_esc=y#v=onepage&q=parametric%20design%20building%20information%20modeling&f=false.
- [4] M. K. Henriksen and J. T. Johansen, "Optimalisering av fagverksbuer for store frispennkonstruksjoner," Master thesis, Institutt for konstruksjonsteknikk, Norges teknisk-naturvitenskapelige universitet (NTNU), Trondheim, 2205, 2020. [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2779982/no.ntnu%3ainspera%3a56731605%3a35324633.pdf?sequence=1&isAllowed=y>
- [5] M. B. G. d. Boer, "Design and optimization of structures with re-used primary elements and new robotic manufactured connection," Master, Department of Built Environment, Eindhoven University of Technology, Eindhoven 0942378, 2021. [Online]. Available: https://pure.tue.nl/ws/portalfiles/portal/189388573/Boer_de_0942378_SED_Teuffel.pdf
- [6] M. Souza and J. T. pini, "Beaver: A computational parametric approach for conception, analysis and design of timber structures.," *World Conference On Timber Engineering 2020*, vol. At: Santiago, Chile, p. 10, 24.08.2020. [Online]. Available: https://www.researchgate.net/publication/366230198_Beaver_A_computational_parametric_approach_for_conception_analysis_and_design_of_timber_structures
- [7] Canaan. "La Sagrada familia." 99% INVISIBLE. <https://99percentinvisible.org/episode/la-sagrada-familia-2/> (accessed 22.02, 2023).
- [8] A. Edmonds, T. Mourtis, and M. Boyle, "Parametric Design—A Drive Towards a Sustainable Future," *Innovation in Construction* no. 1, pp. 221–257, First Online: 24.03.2022. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-95798-8_10
- [9] D. Milutinovic. "Timber in construction industry." Acetra. <https://acetra.rs/news/how-is-timber-changing-the-construction-industry/> (accessed 11.01, 2023).
- [10] F. Luo, R. Y. Man, L. M. J. C. Crabbe, and R. Pu, "Economic development and construction safety research: A bibliometrics approach," *Safety Science*, vol. 145, no. 18, p. 10, Available online: 14.10.2021, doi: <https://doi.org/10.1016/j.ssci.2021.105519>.
- [11] Statista Research Department. "Global construction market size from 2020 to 2021, with forecasts up until 2030." Statista. <https://www.statista.com/statistics/1290105/global-construction-market-size-with-forecasts/> (accessed 11.01, 2023).
- [12] P. Gerbert, S. Castagnino, C. Rothballer, A. Renz, and A. Renz. "The Transformative Power of Building Information Modeling." Boston Consulting Group (BCG). <https://www.bcg.com/publications/2016/engineered-products-infrastructure-digital-transformative-power-building-information-modeling> (accessed 11.01, 2023).
- [13] V. Bertollini. "This Is What Trillions in Global-Infrastructure Investment Look Like." Redshift by Autodesk. <https://redshift.autodesk.com/infographics/global-infrastructure> (accessed 11.01, 2023).

- [14] V. Manickam and I. V. Muralikrishna. "Corporate Social Responsibility." ScienceDirect. <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/corporate-social-responsibility> (accessed 11.01, 2023).
- [15] B. Xia, A. Olanipekun, Q. Chen, L. Xie, and Y. Liu, "Conceptualising the state of the art of corporate social responsibility (CSR) in the construction industry and its nexus to sustainable development," *Journal of Cleaner Production*, vol. 196, no. 14, p. 14, Available online: 21.05.2018, doi: <https://doi.org/10.1016/j.jclepro.2018.05.157>.
- [16] E. M. Langeland. "Arealplaner." Bergen Kommune. (accessed 01.11, 2023).
- [17] Statista Research Department. "Total employed persons in the United States in 2021, by industry." Statista. <https://www.statista.com/statistics/200143/employment-in-selected-us-industries/> (accessed 11.01, 2023).
- [18] M. Norouzi, M. Châfer, L. F. Cabeza, and L. J. D. Boer, "Circular economy in the building and construction sector: A scientific evolution analysis," *Journal of Building Engineering*, vol. 44, no. 6, p. 18, Available online: 11.03.2021, doi: <https://doi.org/10.1016/j.jobe.2021.102704>.
- [19] W. Jiang and J. K. W. Wong, "Key activity areas of corporate social responsibility (CSR) in the construction industry: a study of China," *Journal of Cleaner Production*, vol. 113, no. 12, p. 11, Available online: 03.11.2015, doi: <https://doi.org/10.1016/j.jclepro.2015.10.093>.
- [20] FN-Sambandet. "FNs bærekraftsmål." FN-Sambandet UNITED NASJONS ASSOCIATION OF NORWAY. <https://www.fn.no/om-fn/fns-baerekraftsmaal> (accessed 01.11, 2023).
- [21] Norsk Stål. "Stålproduksjon i dag." Norsk Stål AS. <https://www.norskstaal.no/om-oss/miljoe-og-baerekraft/staalproduksjon-i-dag> (accessed 11.01, 2023).
- [22] K. Bell, *Dimensjonering av trekonstruksjoner* 1st ed. Bergen: Fagbokforlaget, 2017.
- [23] J. Svanæs, B.-I. Andersson, and T. Opdal. "Tre og miljø." Trefokus. <http://www.trefokus.no/resources/filer/fokus-pa-tre/8-Tre-og-miljo.pdf> (accessed 11.01, 2023).
- [24] Treindustrien. "Miljø." treindustrien.no. <https://www.treindustrien.no/miljo> (accessed 11.01, 2023).
- [25] FN-Sambandet. "13 Stoppe klimaendringene." FN-Sambandet (FN) United Nations Association of Norway. <https://www.fn.no/om-fn/fns-baerekraftsmaal/stoppe-klimaendringene> (accessed 11.01, 2023).
- [26] binderholz. "Solid timber is natural, beautiful and cozy." binderholz.com. https://www.binderholz.com/fileadmin/user_upload/books/en/mass_timber_solutions/ (accessed 11.01, 2022).
- [27] Treteknisk. "Bygge i tre " Norsk Treteknisk Institutt. <https://www.treteknisk.no/bygge-i-tre> (accessed 11.01, 2023).
- [28] FN-Sambandet. "15 Livet på land." FN-Sambandet (FN) United Nations Association of Norway. <https://www.fn.no/om-fn/fns-baerekraftsmaal/livet-paa-land> (accessed 11.01, 2023).
- [29] FN-Sambandet. "8 Anstendig arbeid og økonomisk vekst." FN-Sambandet United Nations Association of Norway. <https://www.fn.no/om-fn/fns-baerekraftsmaal/anstendig-arbeid-og-oekonomisk-vekst> (accessed 11.01, 2023).
- [30] M. A. Hossan and A. Nadeem, "TOWARDS DIGITIZING THE CONSTRUCTION INDUSTRY: STATE OF THE ART OF CONSTRUCTION 4.0," *Interdependence between Structural Engineering and Construction Management*, p. 7, May 2019. [Online]. Available: <https://www.researchgate.net/publication/334670417>
- [31] M. M. Burnett and D. W. McIntyre, "Visual programming," *COMPUTER-LOS ALAMITOS-*, vol. 28, p. 14. [Online]. Available: <https://www.cs.auckland.ac.nz/courses/compsci732s1c/archive/2005/lectures/WhatIsVP.pdf>
- [32] B.-g. Hwang, M. Shan, and N. N. B. Supa'at, "Green commercial building projects in Singapore: Critical risk factors and mitigation measures," *Sustainable Cities and Society*, vol. 30, no. 6, pp. 237-247doi: <https://doi.org/10.1016/j.scs.2017.01.020>.

- [33] R. Y. M. Li, "Robots for the Construction Industry," *An Economic Analysis on Automated Construction Safety* no. 1, pp. 23–46, First Online: 11.08.2017, doi: https://doi.org/10.1007/978-981-10-5771-7_2.
- [34] R. Maskuriy, A. Selamat, P. Maresova, O. Krejcar, and O. O. David, "Industry 4.0 for the Construction Industry: Review of Management Perspective," *Economies* vol. 7, no. 3, Published: 04.07.2019, doi: <https://doi.org/10.3390/economies7030068>.
- [35] G. Akos and R. Parsons, *FOUNDATIONS THE GRASSHOPPER PRIMER THIRD EDITION*, R. M. Associates, ed., 3rd ed. modelab.design.com: MODELAB, 2014. [Online]. Available: <https://aae280.files.wordpress.com/2014/10/mode-lab-grasshopper-primer-third-edition.pdf>.
- [36] kissflow. "What Is Visual Programming and How Does It Work?" <https://kissflow.com/low-code/visual-programming-overview/> (accessed 15.02, 2023).
- [37] DATAMYTE. "What Is Visual Programming & Why Is It Important? A General Guide." <https://datamyte.com/visual-programming/> (accessed 15.02, 2023).
- [38] B. Nordbø. "para-." STORE NORSKE LEKSIKON (SNL). <https://snl.no/para-> (accessed 20.02, 2023).
- [39] H. Holden. "metrisk (matematikk)." STORE NORSKE LEKSIKON (SNL). https://snl.no/metrisk_-_matematikk (accessed 20.02, 2023).
- [40] I. Caetano, L. Santos, and A. Leitão, "Computational design in architecture: Defining parametric, generative, and algorithmic design," vol. 9, no. 2, pp. 287-300, Available online: 23.01.2020, doi: <https://doi.org/10.1016/j.foar.2019.12.008>.
- [41] D. Davis. "A History of Parametric." <https://www.danieldavis.com/a-history-of-parametric/> (accessed 21.02, 2023).
- [42] M. Ucal. "parametric equation." Encyclopedia Britannica. <https://www.britannica.com/science/parametric-equation> (accessed 21.02, 2023).
- [43] L. Koumari. "What Is Parametric Design In Architecture?" <https://parametric-architecture.com/what-is-parametric-design-in-architecture/> (accessed 21.02, 2023).
- [44] Parametric Architecture. "Parametric Design in Architecture." YouTube. https://www.youtube.com/watch?v=vlcGTBRoKlg&t=1s&ab_channel=ParametricArchitecture (accessed 22.02, 2023).
- [45] E. MELTZER, "A Parametric Workflow: from Grasshopper3D to Autodesk Inventor," *The Design of Parametric Space*, vol. 1, no. 1. [Online]. Available: http://iiis.org/CDs2012/CD2012IMC/DEMSET_2012/PapersPdf/DC300XS.pdf
- [46] S. Davidson. "About Grasshopper..." Grasshopper ALGORITHMIC MODELING FOR RHINO. <https://www.grasshopper3d.com/> (accessed 20.01, 2023).
- [47] S. Sotiropoulos, G. Kazakis, and N. D. Lagaros, "Conceptual design of structural systems based on topology optimization and prefabricated components," *Computers & Structures*, vol. 226, no. 16, p. 18, Available online: 02.11.2019, doi: <https://doi.org/10.1016/j.compstruc.2019.106136>.
- [48] C. Preisinger, "Linking Structure and Parametric Geometry," *Architectural Design*, vol. 83, no. 2, pp. 110-113, 12.03.2013, doi: <https://doi.org/10.1002/ad.1564>.
- [49] C. Preisinger. "The official guide to using Karamba3D 2.2.0." Karamba3D parametric engineering. <https://manual.karamba3d.com/> (accessed 31.01, 2023).
- [50] D. Rutten, "Galapagos: On the Logic and Limitations of Generic Solvers," *Architectural Design*, vol. 83, no. 2, pp. 132-135, 12.03, doi: <https://doi.org/10.1002/ad.1568>.
- [51] A. Brøgger and K. Lysne. "utviklingslæren." STORE NORSKE LEKSIKON (SNL). <https://snl.no/utviklingsl%C3%A6ren?fbclid=IwAR3wcXoEms7UTxVbH3874Ja7pVHsLp9Ctttn6byR-ke740AeBs29GRhZhhE> (accessed 12.02, 2023).
- [52] Wikipedia. "Genetisk algoritme." Wikipedia Den frie encyklopedi. https://no.wikipedia.org/wiki/Genetisk_algoritme?fbclid=IwAR0bk_TY2EVVCCc-L6DD7XaB5vBfQcgN7bwZZWU3O4171qdEsrFZ3X-B6dg (accessed 12.02, 2023).

- [53] MathWorks. "What Is the Genetic Algorithm?" https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html?fbclid=IwAR20R8zTMhYtgNUAUvUFdXFSC8ehIk-60cmvmflvRTiulm4On3y_6ijWe2Y (accessed 11.02, 2023).
- [54] MathWorks. "What Is Simulated Annealing?" <https://se.mathworks.com/help/gads/what-is-simulated-annealing.html?fbclid=IwAR2raeaPhbMelgDxWBJkzvrkE2N331EfVZFrLnjfZFKjWGfOvZR-1hfy0> (accessed 12.02, 2023).
- [55] Y. S. Lee, E. Graham, G. Jackson, A. Galindo, and C. S. Adjiman. "Simulated Annealing Algorithm." ScienceDirect. <https://www.sciencedirect.com/topics/engineering/simulated-annealing-algorithm?fbclid=IwAR0gi0fa4RQPy5WXjjjvPx4uWAY859ir-VHzF2oFhO3O76nIFulBQ-iqs> (accessed 12.02, 2023).
- [56] Y. Krymsky. "Optimization with Galapagos." Responsive Skins An exploration of Paratonic Surfaces in Architecture. https://yazdanistudioresearch.wordpress.com/2014/01/31/optimization-with-galapagos/?fbclid=IwAR0Y-Tjri8Eng_xUsdyurGbE4FjrmklqLPn5eCp89XI1Uap1ye-O8fMAUJg (accessed 11.02, 2023).
- [57] H. J. Blaß and C. Sandhaas, *Timber Engineering - Principles for Design*, 1st ed. Karlsruhe: KIT Scientific Publishing, 2017. [Online]. Available: <https://publikationen.bibliothek.kit.edu/1000069616>. Accessed on: 20.02.2023.
- [58] R. Crocetti *et al.*, *Limtreboka*, 1st ed. Finland, Norge og Sverige: Norske Limtreprodusenters Forening, 2015. [Online]. Available: https://www.moelven.com/globalassets/moelven-limtre/limtreboka_2015_el2.pdf.
- [59] D. Pasca, "Review of Timber Properties," University of Agder, Forelesning, 29.08.2022, unpublished.
- [60] Dag Gundersen. "nominell." Store norske leksikon. <http://snl.no/nominell> (accessed 27.02.2023, 2023).
- [61] Helmut Ormestad. "Elastisitetsmodul." Store Norske Leksikon. <https://snl.no/elastisitetsmodul> (accessed 01.03, 2023).
- [62] T. C. Ian Mills, K. Homann, N. Kallay, and Kozo Kuchitsu, *Quantities, Units and Symbols in Physical Chemistry*, 2nd ed., 1988. [Online]. Available: https://old.iupac.org/publications/books/gbook/green_book_2ed.pdf.
- [63] *Eurokode 5: Prosjektering av trekonstruksjoner. Del 1-1: Allmenne regler og regler for bygninger*, NS-EN 1995-1-1:2004+A1:2008+NA:2010, Standard Norge, standard.no, 01.07.2010 2010. [Online]. Available: <https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=436137>
- [64] K. Bell and H. Liven, *Limtreboka Beregningseksempler*, 2nd ed. Trondheim/Moelv: Norske Limtreprodusenters Forening, 2018. [Online]. Available: https://www.moelven.com/globalassets/moelven-limtre/limtre_eksmpl_digital.pdf.
- [65] Think Wood. "Connection Options for Wood-Frame and Heavy Timber Buildings." Continuing Education Center Architecture + Construction. https://continuingeducation.bnpmmedia.com/article_print.php?C=1464&L=312 (accessed 05.03, 2023).
- [66] M. B. U. Pedersen, L. Damkilde, and P. Hoffmeyer, "Dowel Type Timber Connections Strength modelling," in "Byg Rapport," Technical University of Denmark, Kgs. Lyngby, Denmark, Rapport Published - May 2001 2001. [Online]. Available: <https://core.ac.uk/download/pdf/13738109.pdf>
- [67] W. Lin and T. Yoda. "Truss." ScienceDirect. <https://www.sciencedirect.com/topics/engineering/truss> (accessed 05.03, 2023).

- [68] T. Dyken, "Trebruer," Vegdirektoratet, Norge, 422, 2017. Accessed: 06.03.2023. [Online]. Available: <https://vegvesen.brage.unit.no/vegvesen-xmlui/bitstream/handle/11250/2670309/Rapport%20422%20Trebruer.pdf?sequence=1&isAllowed=y>
- [69] G. Greschik, "Truss Beam with Tendon Diagonals: Mechanics and Designs," *AIAA Journal*, vol. 46, no. 3, Published Online: 02.05.2012, doi: <https://doi.org/10.2514/1.25249>.
- [70] Z. Li, W. Feng, J. Ou, F. Liang, and M. He, "Experimental investigations into the mechanical performance of glulam dowel-type connections with either bolts or screws as fasteners," *Journal of Wood Science* volume vol. 67, no. 25, Published: 17.12.2021. [Online]. Available: <https://jwoodscience.springeropen.com/articles/10.1186/s10086-021-02002-5>
- [71] M. He, J. Zhang, and Z. Li, "Influence of cracks on the mechanical performance of dowel type glulam bolted joints," *Construction and Building Materials*, vol. 153, no. 13, pp. 445-458, Available online: 21.07.2017, doi: <https://doi.org/10.1016/j.conbuildmat.2017.07.095>.
- [72] Z. Shu *et al.*, "Reinforced moment-resisting glulam bolted connection with coupled long steel rod with screwheads for modern timber frame structures," *Earthquake Engineering & Structural Dynamics*, vol. 52, no. 4, pp. 845-864, Published: 31.12.2022, doi: <https://doi.org/10.1002/eqe.3789>.
- [73] S. Tulebekova, K. A. Malo, A. Rønquist, and P. Nåvik, "Modeling stiffness of connections and non-structural elements for dynamic response of taller glulam timber frame buildings," *Engineering Structures*, vol. 261, no. 18, Available online: 22.04.2022, doi: <https://doi.org/10.1016/j.engstruct.2022.114209>.
- [74] T. Gečys, T. K. Bader, A. Olsson, and S. Kajėnas, "Influence of the rope effect on the slip curve of laterally loaded, nailed and screwed timber-to-timber connections," *Construction and Building Materials*, vol. 228, no. 15, p. 13, Available online: 21.08.2019, doi: <https://doi.org/10.1016/j.conbuildmat.2019.116702>.
- [75] R. Jockwer, G. Fink, and J. Köhler, "Assessment of the failure behaviour and reliability of timber connections with multiple dowel-type fasteners," *Engineering Structures*, vol. 172, no. 14, pp. 76-84, Available online: 15.06.2018, doi: <https://doi.org/10.1016/j.engstruct.2018.05.081>.
- [76] Vegdirektoratet, "Bruprosjektering," in "Prosjektering av bruer, ferjekaier og andre bærende konstruksjoner," Statens vegvesen Vegvesen, 04/2015, 2015. Accessed: 01.03.2023. [Online]. Available: <https://www.vegvesen.no/globalassets/fag/handboker/hb-n400-bruprosjektering.pdf>
- [77] M. Yurrita, J. M. Cabrero, and E. Moreno-Zapata, "Brittle failure in the parallel-to-grain direction of timber connections with small diameter dowel-type fasteners: A new design model for plug shear," *Engineering Structures*, vol. 241, no. 17, p. 14, Available online: 19.05.2021, doi: <https://doi.org/10.1016/j.engstruct.2021.112450>.
- [78] A. Hanhijärvi and A. Kevarinmäki, "Timber failure mechanisms in highcapacity dowelled connections of timber to steel," *Experimental results and design*, vol. PUBLICATIONS 677, p. 98. [Online]. Available: <https://www.vttresearch.com/sites/default/files/pdf/publications/2008/P677.pdf>
- [79] T. V. Holvik and M. S. Strøm, "Store dybelforbindelser i fagverksbruer av tre : analyse av sprø bruddmekanismer og global oppførsel," Masteroppgave, Byggeteknikk og arkitekturt Fakultet for realfag og teknologi Norges miljø- og biovitenskapelige universitet (NMBU), 1527, 2018. [Online]. Available: <https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2563228>
- [80] P. Moore and G. Booth. "Ductile Failure." ScienceDirect. <https://www.sciencedirect.com/topics/engineering/ductile-failure> (accessed 01.03, 2023).
- [81] P. Zarnani and P. Quenneville, "Strength of timber connections under potential failure modes: An improved design procedure," *Construction and Building Materials*, vol. 60, no. 10, pp. 81-90, Available online: 22.03.2014, doi: <https://doi.org/10.1016/j.conbuildmat.2014.02.049>.

- [82] P. Zarnani and P. Quenneville, "Group Tear-Out in Small-Dowel-Type Timber Connections: Brittle and Mixed Failure Modes of Multinail Joints," *Journal of Structural Engineering*, vol. 141, no. 2, p. 10doi: [http://dx.doi.org/10.1061/\(ASCE\)ST.1943-541X.0001053](http://dx.doi.org/10.1061/(ASCE)ST.1943-541X.0001053).
- [83] D. Pasca, "Timber Connections," Universitetet i Agder (UiA), Forelesning, 10.10.2022, Unpublised, BYG514-G 22H Konstruksjonsteknikk VI - tre
- [84] J.-Q. Yang, P. Feng, and S. T. Smith, "Strengthening single-bolt timber joints with externally bonded CFRP composites," *Structures* vol. 28, no. 1, pp. 2671-2685, Available online: 11.11.2020, doi: <https://doi.org/10.1016/j.istruc.2020.10.024>.
- [85] J. L. Jensen and P. Quenneville, "Experimental investigations on row shear and splitting in bolted connections," *Construction and Building Materials*, vol. 25, no. 5, pp. 2420-2425, Available online: 30.12.2010, doi: <https://doi.org/10.1016/j.conbuildmat.2010.11.050>.
- [86] Z. Cui, S. Tang, M. Xu, D. Huang, and Z. Zhang, "Experimental and theoretical study on mechanical performance of dowel joints with inserted steel plates constructed of laminated bamboo lumber," *Journal of Building Engineering*, vol. 61, no. 7, p. 12, Available online: 27.09.2022, doi: <https://doi.org/10.1016/j.jobbe.2022.105320>.
- [87] M. Yurrita, J. M. Cabrero, and P. Quenneville, "Brittle failure in the parallel-to-grain direction of multiple shear softwood timber connections with slotted-in steel plates and dowel-type fasteners," *Construction and Building Materials*, vol. 216, no. 15, pp. 296-313, Available online: 09.05.2019, doi: <https://doi.org/10.1016/j.conbuildmat.2019.04.100>.
- [88] J. Sjodin, E. Serrano, and B. Enquist, "An experimental and numerical study of the effect of friction in single dowel joints," *European Journal of Wood and Wood Products Holz als Roh- und Werkstoff*, vol. 66, no. 5, pp. 363-372, Published: 21.08.2008, doi: <https://doi.org/10.1007/s00107-008-0267-z>.
- [89] J. L. Jensen and P. Quenneville, "Fracture mechanics analysis of row shear failure in dowelled timber connections," *Wood Science and Technology*, vol. 44, no. 4, pp. 639-653, Available online: November 2010. [Online]. Available: <https://link.springer.com/article/10.1007/s00226-009-0295-9>
- [90] M. Dorn, K. d. Borst, and J. Eberhardsteiner, "Experiments on dowel-type timber connections," *Engineering Structures*, vol. 47, no. 9, pp. 67-80, Available online: 24.11.2012, doi: <https://doi.org/10.1016/j.engstruct.2012.09.010>.
- [91] M. Yurrita and J. M. Cabrero, "Effective thickness of timber elements for the evaluation of brittle failure in timber-to-steel connections with large diameter fasteners loaded parallel-to-grain at the elastic range: A new method based on a beam on elastic foundation," *Engineering Structures*, vol. 209, no. 16, p. 14, Available online: 06.12.2019, doi: <https://doi.org/10.1016/j.engstruct.2019.109959>.
- [92] J. H. Prestegarden, "Kontroll av fagverksbruer i tre," Region øst Ressursavdelingen faggruppen, Vegvesen.no, September, 2016. Accessed: 08.04.2023. [Online]. Available: <https://www.vegvesen.no/globalassets/nyheter/bilder/komm/kontroll-av-fagverksbruer-i-tre.pdf>
- [93] K. Sawata, T. Sasaki, and S. Kanetaka, "Estimation of shear strength of dowel-type timber connections with multiple slotted-in steel plates by European yield theory," *Journal of Wood Science volume* vol. 52, no. 800, pp. 496-502, Published: 01.12.2006, doi: <https://doi.org/10.1007/s10086-006-0800-9>.
- [94] P. Aanensen, "Connections in timber structures," Universitetet i Agder (UiA), veiledning masterprosjekt og tilsendt forelesning fra 2021, 27.03.2023, unpublished, BYG514-G 22H Konstruksjonsteknikk VI - tre
- [95] I. Aydogdu, T. O. Ormecioglu, O. Tunca, and S. Carbas, "Design of large-scale real-size steel structures using various modified grasshopper optimization algorithms," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13825-13848, Published: 10.04.2022. [Online]. Available: <https://link.springer.com/article/10.1007/s00521-022-07196-3>

- [96] E. Ching and J. V. Carstensen, "Truss topology optimization of timber–steel structures for reduced embodied carbon design," *Engineering Structures*, vol. 252, no. 18, Available online: 23.11.2021, doi: <https://doi.org/10.1016/j.engstruct.2021.113540>.
- [97] R. P. Khidmat, H. Fukuda, and Kustiani, "Design Optimization of Hyperboloid Wooden House Concerning Structural, Cost, and Daylight Performance," *Buildings*, vol. 12, no. 2, p. 31doi: <https://doi.org/10.3390/buildings12020110>.
- [98] Z. Aydın, "Size, layout and tendon profile optimization of prestressed steel trusses using Jaya algorithm," *Structures*, vol. 40, no. 7, pp. 284-294, Available online: 15.04.2022, doi: <https://doi.org/10.1016/j.istruc.2022.04.014>.
- [99] B. Malgıt, Ü. Işıkdag, G. Bekdaş, and M. Yücel, "A generative design-to-BIM workflow for minimum weight plane truss design," *Revista de la Construcción*, vol. 21, no. 2, Accepted: 19.08.2022. [Online]. Available: https://www.scielo.cl/scielo.php?pid=S0718-915X2022000200473&script=sci_arttext
- [100] A. Y. Aydođdu, M. Artar, and M. Ergün, "Optimum weight design of steel truss roof systems considering corrosion effect," *Structures*, vol. 49, no. 7, pp. 88-105, Available online: 31.01.2023, doi: <https://doi.org/10.1016/j.istruc.2023.01.099>.
- [101] R. Ponginan. "What is Size, Shape, and Free-Shape Optimization?" ALTAIR University. <https://altairuniversity.com/52528-what-is-size-shape-and-free-shape-optimization/> (accessed 25.03, 2023).
- [102] S. Šilih, M. Premrov, and S. Kravanja, "Optimization of timber trusses considering joint flexibility," *High Performance Structures and Materials III*, vol. 85, p. 9. [Online]. Available: <https://www.witpress.com/Secure/elibrary/papers/HPSM06/HPSM06060FU1.pdf>
- [103] A. Gonzalez and J. Antonio, "Design, analysis and application of innovative connections for optimized steel trusses," Master, Civil Engineering | Building Engineering - Structural Design, Delft University of Technology, TUDelft, 2020. [Online]. Available: <https://repository.tudelft.nl/islandora/object/uuid:89c7f26e-8190-4f95-bb2c-5f74b2c9b41c?collection=education>
- [104] A. Kaveh and M. Khayatazad, "Ray optimization for size and shape optimization of truss structures," *Computers & Structures*, vol. 117, no. 9, pp. 82-94, Available online: 07.01.2013, doi: <https://doi.org/10.1016/j.compstruc.2012.12.010>.
- [105] C. C. Mitropoulou, Y. Fourkiotis, N. D. Lagaros, and M. G. Karlaftis. "Shape Optimization Problem." ScienceDirect. <https://www.sciencedirect.com/topics/engineering/shape-optimization-problem> (accessed 26.03, 2023).
- [106] D. Wang, W. H. Zhang, and J. S. Jiang, "Truss Optimization on Shape and Sizing with Frequency Constraints," *AIAA Journal*, vol. 42, no. 3, Published online: 02.05.2012, doi: <https://doi.org/10.2514/1.1711>.
- [107] W. Lingyun, Z. Mei, W. Guangming, and M. Guang, "Truss optimization on shape and sizing with frequency constraints based on genetic algorithm," *Computational Mechanics*, vol. 35, no. 5, pp. 361–368, Published: 17.11.2004. [Online]. Available: <https://link.springer.com/article/10.1007/s00466-004-0623-8>
- [108] H. M. Gomes, "Truss optimization with dynamic constraints using a particle swarm algorithm," *Expert Systems with Applications*, vol. 38, no. 1, pp. 957-968, Available online: 04.08.2010, doi: <https://doi.org/10.1016/j.eswa.2010.07.086>.
- [109] A. Tam. "A Gentle Introduction to Particle Swarm Optimization." Machine learning mastery. <https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/> (accessed 26.03, 2023).
- [110] J. Noh, T. Badloe, C. Lee, J. Yun, S. So, and J. Rho. "Inverse design meets nanophotonics: From computational optimization to artificial neural network." ScienceDirect. <https://www.sciencedirect.com/topics/computer-science/topology-optimization> (accessed 27.05, 2023).

- [111] E. Suwartadi. "Adjoint Optimization On State Constraint and Second Order Adjoint Computation." Norges teknisk-naturvitenskapelige universitet (NTNU). https://www.ntnu.no/c/document_library/get_file?uuid=c08c69bc-1713-4243-b974-efc89cab34d0&groupId=10361 (accessed 27.03, 2023).
- [112] K. Mam, C. Douthe, R. L. Roy, and F. Consigny, "Shape optimization of braced frames for tall timber buildings: Influence of semi-rigid connections on design and optimization process," *Engineering Structures*, vol. 216, no. 16, Available online: 23.05.2020, doi: <https://doi.org/10.1016/j.engstruct.2020.110692>.
- [113] Q. Cai, R. Feng, and Z. Zhang, "Topology optimization of truss structure considering nodal stability and local buckling stability," *Structures*, vol. 40, no. 7, pp. 64-73, Available online: 09.04.2022, doi: <https://doi.org/10.1016/j.istruc.2022.04.008>.
- [114] Y. K. Yi, A. Tariq, J. Park, and D. Barakat, "Multi-objective optimization (MOO) of a skylight roof system for structure integrity, daylight, and material cost," *Journal of Building Engineering*, vol. 34, no. 6, Available online: 02.12.2020, doi: <https://doi.org/10.1016/j.jobbe.2020.102056>.
- [115] *Eurokode 3: Prosjektering av stålkonstruksjoner. Del 1-8: Knutepunkter og forbindelser*, NS-EN 1993: 1-8:2005+NA:2009, Standard Norge, standard.no, 10.01.2009 2009. [Online]. Available: <https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=398439>
- [116] Sweco, "Beregningsdokument - Limtrefagverk," 25.08.2021, unpublished.
- [117] Wikipedia. "Rhinoceros 3D." Wikipedia the free encyclopedia. https://en.wikipedia.org/wiki/Rhinoceros_3D (accessed 13.02, 2023).
- [118] N. A. JAKOBSSON and S. BOHMAN, "A Generative Design of Timber Structures According to Eurocode," Master Thesis, Department of Civil and Architectural Engineering, Royal Institute of Technology (KTH), STOCKHOLM, 19217, 2019. [Online]. Available: <http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1340578>
- [119] sculpteo. "Rhino 3D software: 3D Modeling Software for Architecture and Industrial Design." sculpteo, A brand of BASF - We create chemistry. <https://www.sculpteo.com/en/glossary/rhinoceros-definition/> (accessed 13.02, 2023).
- [120] Rhinoceros. "What are NURBS?" Rhinoceros design, model, present, analyze, realize. <https://www.rhino3d.com/features/nurbs/> (accessed 13.02, 2023).
- [121] Autodesk. "BIM structural analysis software for engineers." autodesk.com. <https://www.autodesk.com/products/robot-structural-analysis/overview?term=1-YEAR&tab=subscription> (accessed 16.04, 2023).
- [122] Microsoft. "Microsoft Excel." microsoft.com. <https://www.microsoft.com/nb-no/microsoft-365/excel> (accessed 16.04, 2023).
- [123] A. Aamodt, "RIB prosjektering ved implementering av parametrisk verktøy : Prosjektering av reviderbare brudekker, med en arbeidsflyt mellom Grasshopper, Revit og FEM-design," Master, Fakultet for teknologi og realfag Institutt for ingeniørvitenskap, University of Agder, Grimstad, 306, 2021. [Online]. Available: <https://hdl.handle.net/11250/2778891>
- [124] C. Reilly. "Learning Grasshopper." LinkedIn Learning. <https://www.linkedin.com/learning/learning-grasshopper/next-steps?autoplay=true&u=57077761> (accessed 21.01, 2023).
- [125] W. Zesk. "Grasshopper: Generative Design for Architecture." LinkedIn Learning. <https://www.linkedin.com/learning/grasshopper-generative-design-for-architecture/cyborg-designers?autoplay=true&u=57077761> (accessed 22.01, 2023).
- [126] W. Zesk. "Grasshopper: Tips, Tricks, and Techniques." LinkedIn Learning. <https://www.linkedin.com/learning/grasshopper-tips-tricks-and-techniques/tips-tricks-and-techniques-for-grasshopper?autoplay=true&resume=false&u=57077761> (accessed 24.01, 2023).

- [127] Rhino Grasshopper. YouTube. <https://www.youtube.com/@ParametricHouse/videos> (accessed 15.01, 2023).
- [128] Parametric Engineer. "Your destination to learn more about Grasshopper and parametric engineering for Structural Engineers." YouTube. <https://www.youtube.com/@parametricengineer7046/videos> (accessed 15.01, 2023).
- [129] D. Apellániz. "Diego Apellániz." Youtube. <https://www.youtube.com/@DiegoApellaniz/about> (accessed 10.01, 2023).
- [130] D. Copete. "DCO Architecture." Youtube. <https://www.youtube.com/@DCOArchitecture/about> (accessed 11.01, 2023).
- [131] BIM Corner. "Learn BIM Easy." YouTube. <https://www.youtube.com/@bimcorner6331/about> (accessed 25.01, 2023).
- [132] Structured Parametrics. "The Structured Parametrics YouTube channel for demos of our services, parametric tutorials and other information relating to Engineering (especially Structural), BIM, modelling, exporting and exploring a digital future." YouTube. <https://www.youtube.com/@StructuredParametrics/about> (accessed 09.01, 2023).
- [133] I. S. Torp. "Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora." De nasjonale forskningsetiske komiteene. <https://www.forskningsetikk.no/retningslinjer/hum-sam/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora/> (accessed 20.05, 2023).
- [134] Sleeping Giant media. "7 Reasons Why Google Is The Search Engine Of Choice." <https://www.sleepinggiantmedia.co.uk/7-reasons-why-google-is-the-search-engine-of-choice/> (accessed 25.01, 2023).
- [135] Tower. "Why Google is the Best Search Engine (and Why Businesses Should Care) – In My Opinion." towermarketing.net. <https://www.towermarketing.net/blog/google-best-search-engine/> (accessed 25.01, 2023).
- [136] Enago Author First Quality First. "The Stages of the Publishing Process." enago.com. <https://www.enago.com/author-hub/the-stages-of-the-publishing-process> (accessed 30.01, 2023).
- [137] Kildekompasset. "Kildekritikk." kildekompasset.no. <https://kildekompasset.no/kildekritikk/> (accessed 03.02, 2023).
- [138] A. Dahl and A. Hille. "engelsk." STORE NORSKE LEKSIKON (SNL). <https://snl.no/engelsk> (accessed 10.02, 2023).
- [139] T. Henriksen. "nøkkelord." STORE NORSKE LEKSIKON (SNL). <https://snl.no/n%C3%B8kkelord> (accessed 05.04, 2023).
- [140] S. Elo and H. Kyngäs, "The qualitative content analysis process," *JAN Leaning Global Nursing Research*, vol. 62, no. 1, pp. 107-115, First published: 18.03.2008, doi: <https://doi.org/10.1111/j.1365-2648.2007.04569.x>.
- [141] M. Bengtsson, "How to plan and perform a qualitative study using content analysis," *NursingPlus Open*, vol. 2, no. 2, pp. 8-14, Available online: 17.02.2016, doi: <https://doi.org/10.1016/j.npls.2016.01.001>.
- [142] McNeel Europe. "CONTACT." food4Rhino. <https://www.food4rhino.com/en/contact> (accessed 02.02, 2023).
- [143] ShapeDriver. "What Are Rhino Grasshopper Plugins? | feat. Food4Rhino." YouTube. https://www.youtube.com/watch?v=srsTDJr09R0&ab_channel=ShapeDiver (accessed 06.04, 2023).
- [144] W. Pearson. "Creating a Grasshopper Plug-In Package." RhinoDeveloper. <https://developer.rhino3d.com/guides/yak/creating-a-grasshopper-plugin-package/> (accessed 06.04, 2023).
- [145] R. Rodricks. "Grasshopper Addons and Plugins." Grasshopper Docs. <https://grasshopperdocs.com/> (accessed 04.02, 2023).

- [146] ekimroyrp. "PUFFERFISH " food4Rhino. <https://www.food4rhino.com/en/app/pufferfish> (accessed 20.01, 2023).
- [147] A. Bahrami. "HETEROPTERA " food4Rhino. <https://www.food4rhino.com/en/app/heteroptera> (accessed 24.01, 2023).
- [148] karamba. "KARAMBA3D." food4Rhino. <https://www.food4rhino.com/en/app/karamba3d> (accessed 20.01, 2023).
- [149] Karamba3D. "License." <https://karamba3d.com/get-started?subnav=download> (accessed 25.01, 2023).
- [150] N. Miller. "LUNCHBOX." food4Rhino. <https://www.food4rhino.com/en/app/lunchbox> (accessed 25.01, 2023).
- [151] Rhino Grasshopper. "Rhino Grasshopper Tutorial (Beginner)." YouTube. https://www.youtube.com/watch?v=Y66bSN4QFUE&t=2420s&ab_channel=RhinoGrasshopper (accessed 20.01, 2023).
- [152] COREstudio. "TT TOOLBOX." food4Rhino. <https://www.food4rhino.com/en/app/tt-toolbox> (accessed 08.02, 2023).
- [153] Beaver Structures. "BEAVER." food4Rhino. <https://www.food4rhino.com/en/app/beaver> (accessed 31.02, 2023).
- [154] Rhinoceros Forums. "Karamba for advanced timber analysis." discourse.mcneel.com. <https://discourse.mcneel.com/t/karamba-for-advanced-timber-analysis/65977> (accessed 24.01, 2023).
- [155] weArk. "SNAPPINGGECKO." food4Rhino. <https://www.food4rhino.com/en/app/snappinggecko> (accessed 05.02.2023, 2023).
- [156] BIM Corner. "5 GRASSHOPPER TRICKS TO DEFEAT SPAGHETTI MONSTER." YouTube. https://www.youtube.com/watch?v=jjrn0cXc6O8&ab_channel=BIMCorner (accessed 05.02, 2023).
- [157] marcsyp. "TELEPATHY " food4Rhino. <https://www.food4rhino.com/en/app/telepathy> (accessed 05.02, 2023).
- [158] marcsyp. "Bifocals." food4Rhino. <https://www.food4rhino.com/en/app/bifocals> (accessed 20.01, 2023).
- [159] gnannicini. "RBFOpt library for black-box optimization." github.com. <https://github.com/coin-or/rbfopt> (accessed 11.05, 2023).
- [160] S. Alarie, C. Audet, A. E. Gheribi, M. Kokkolaras, and S. L. Digabel, "Two decades of blackbox optimization applications," *EURO Journal on Computational Optimization*, vol. 9, Available online: 24.09.2021, doi: <https://doi.org/10.1016/j.ejco.2021.100011>.
- [161] Wikipedia. "Radial basis function interpolation." en.Wikipedia.org. https://en.wikipedia.org/wiki/Radial_basis_function_interpolation (accessed 11.05, 2023).
- [162] H. Chen and B. R. Bakshi. "Radial Basis Function." sciencedirect.com. <https://www.sciencedirect.com/topics/veterinary-science-and-veterinary-medicine/radial-basis-function> (accessed 11.05, 2023).
- [163] Opossum Support. "OPTIMIZATION SOLVER WITH SURROGATE MODELS." <https://www.food4rhino.com/en/app/opossum-optimization-solver-surrogate-models> (accessed 11.05, 2023).
- [164] Eurokode — Grunnlag for prosjektering av konstruksjoner, NS-EN 1990:2002+A1:2005+AC:2010+NA:2016, Standard Norge, standard.no, 05.01.2026 2016. [Online]. Available: <https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1452922>
- [165] Eurokode 1: Laster på konstruksjoner - Del 1-3: Allmenne laster - Snølaster, NS-EN 1991-1-3:2003+NA:2008, Standard Norge, standard.no, 20.02.2008 2008.
- [166] Beaver Structures. "Beaver Tutorial Ep1: Basic Workflow [EN]." YouTube. <https://www.youtube.com/@beaverstructures8597/videos> (accessed 10.02, 2023).

- [167] *Eurokode 3: Prosjektering av stålkonstruksjoner. Del 1-1: Allmenne regler og regler for bygninger*, NS-EN 1993-1-1:2005+A1:2014+NA:2015, S. Norge, standard.no, 06.01.2015 2015. [Online]. Available:
<https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=755701>
- [168] Robert Vierlinger_1. "OCTOPUS." food4Rhino. <https://www.food4rhino.com/en/app/octopus> (accessed 10.02, 2023).
- [169] Moelven. "Moelven Limtre - prislisterNY." moelven.com.
<https://www.moelven.com/globalassets/moelven-limtre/dimensjoner-ny-profil.pdf> (accessed 07.05, 2023).
- [170] Standard Norge. "NS-EN 1995-1-1/prNA:2023." <https://sn-enq-prod-sv-app-app.azurewebsites.net/product/257f526a-f265-4e25-69f6-08db1003760b/nb> (accessed 26.04.2023, 2023).
- [171] *Eurokode 3: Prosjektering av stålkonstruksjoner - Del 1-5: Plater påkjent i plateplanet*, NS-EN 1993-1-5:2006+AC+A1:2017+A2:2019+NA:2019, Standard Norge, standard.no, 10.01.2019 2019. [Online]. Available:
<https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1094402>
- [172] P. Aanensen, "Samtale (veiledning for masterprosjekt)," Universitetet i Agder (UiA), 10.03.2023-15.05.2023, Unpublished.

12 Vedlegg

Vedlegg 1: Beaver CSV fil

Vedlegg 2: Materialinnhenting via Excel

Vedlegg 3: Utnyttelsesberegninger elementer Excel

Vedlegg 4: Knutepunkts beregninger Excel

Vedlegg 5: Komplette Grasshopper fil

Vedlegg 6: Grasshopper-filen for optimalisering

Vedlegg 7: Optimaliserings iterasjoner Excel

Vedlegg 8: Grasshopper-fil korrigerende av optimalisering.

Vedlegg 9: Bekreftelse fra Sweco.

Vedlegg 10: A3-poster