

Woord vooraf

Een jaar geleden, na het succesvol afronden van mijn bachelorproef, werd ik verrast door het verzoek om dit onderzoek verder uit te breiden tot een masterproef. Zonder enige aarzeling heb ik besloten deze uitdaging aan te gaan. Na vele uren en lange dagen van intensief werken, kan ik met volle overtuiging zeggen dat ik er geen moment spijt van heb gehad. Dit project is het grootste dat ik tot nu toe in mijn leven heb gerealiseerd, en ik ben enorm trots op het resultaat. Deze ervaring heeft mijn interesse in informatica, en met name in softwareontwikkeling, verder aangewakkerd en mij ertoe aangezet om mijn studie voort te zetten en het diploma van industrieel ingenieur in de informatica te behalen.

Dit onderzoek zou niet mogelijk zijn geweest zonder de begeleiding en ondersteuning van mijn promotor, Prof. dr. ing. Bram Vandoren, en mijn begeleider, ing. Glenn van Vugt. Ik wil Professor Vandoren van harte bedanken voor zijn voortdurende begeleiding, zijn bereidheid om mijn vragen te beantwoorden, en voor het wijzen van de juiste richting in dit project. Eveneens wil ik Ing. van Vugt bedanken voor zijn hulp bij het oplossen van problemen met mijn code en voor het geven van waardevolle tips en suggesties. Verder ben ik dr. ing. Rik Steensels dankbaar voor het beantwoorden van mijn vragen over Eurocode 2. Daarnaast wil ik Prof. dr. ir. Hervé Degée bedanken voor zijn antwoorden op mijn vragen over Eurocode 3 en zijn inspanningen om de formules voor de C_1 - en C_2 -factoren voor de stabiliteitscontrole van staal te achterhalen.

Tot slot wil ik mijn familie en vriendin bedanken voor hun geduld, positieve houding, en enthousiasme bij het zien van mijn voortgang, zelfs als ze niet alles begrepen.

Inhoudsopgave

Woord vooraf	1
Lijst met tabellen	7
Lijst met figuren	11
Verklarende woordenlijst	13
Abstract	21
Abstract in English	23
1 Inleiding	25
1.1 Situering	25
1.2 Probleemstelling	26
1.3 Doelstellingen	26
1.4 Methode	27
1.5 Limitaties	28
1.6 Vooruitblik	28
2 Literatuurstudie	31
2.1 Inleiding	31
2.2 Onderzoek	32
2.2.1 Diamonds	32
2.2.2 SCIA Engineer	35
2.2.3 ConCrete (Plus)	39
2.2.4 Tekla Tedds	41
2.2.5 AxisVM	44
2.2.6 Bachelorproef Gielkens, Jans en Sanen	46
2.2.7 Masterproef Bloemen en Jehaes	48
2.2.8 Overzicht	49
2.3 Conclusie	53
3 Achterliggende formules	55
3.1 Analyse balk	55
3.1.1 Inleiding Eurocode 0	56
3.1.2 Eindige-elementenmethode	59
3.1.3 Voorbeeld	63

3.2	Beton	70
3.2.1	Aannames en beperkingen	70
3.2.2	Algemene eigenschappen	70
3.2.3	Langswapening	78
3.2.4	Dwarskrachtwapening	81
3.2.5	Spanningen	84
3.2.6	Scheurwijdte	85
3.2.7	Doorbuiging	87
3.3	Staal	88
3.3.1	Aannames en beperkingen	88
3.3.2	Algemeen	88
3.3.3	Sterkte	89
3.3.4	Stabiliteit	92
3.3.5	Doorbuiging	93
3.4	Hout	94
3.4.1	Aannames en beperkingen	94
3.4.2	Algemene eigenschappen	94
3.4.3	Sterkte	95
3.4.4	Stabiliteit	98
3.4.5	Doorbuiging	99
3.5	Conclusie	101
4	Back end van het programma	103
4.1	Database	103
4.1.1	Voordelen	103
4.1.2	Opgeslagen informatie	104
4.1.3	Ontwerp	105
4.1.4	Code	106
4.2	Analyse	110
4.2.1	Model balk	110
4.2.2	Belastingscombinaties	114
4.2.3	Mesh	120
4.2.4	Eindige-elementenmethode	121
4.3	Beton	127
4.3.1	Doorsnede analyse	127
4.3.2	Controle doorsnede	130
4.3.3	Optimalisatie wapening	135
4.3.4	Optimalisatie doorsnede en wapening	144
4.3.5	Positionering wapening	145
4.4	Staal	148
4.4.1	Controle doorsnede	148
4.4.2	Optimalisatie profiel	150
4.5	Hout	153
4.5.1	Belastingscombinaties voor kruip	153
4.5.2	Controle doorsnede	154
4.5.3	Optimalisatie doorsnede	156

5	Gebruikersinterface	159
5.1	Geometrie en steunpunten	160
5.2	Belastingsgevallen, belastingen en combinaties	160
5.3	Materialen	162
5.3.1	Beton	162
5.3.2	Staal	164
5.3.3	Hout	165
5.4	Resultaten	165
5.5	Optimalisatie	167
5.5.1	Beton	168
5.5.2	Staal	168
5.5.3	Hout	169
5.6	Berekeningen	169
6	Uitgewerkt voorbeeld	171
6.1	Analyse	172
6.1.1	Moment	172
6.1.2	Dwarskracht	173
6.1.3	Vervorming	173
6.1.4	Conclusie	174
6.2	Beton	174
6.2.1	Langswapening	175
6.2.2	Dwarskrachtwapening	176
6.2.3	Spanningen	177
6.2.4	Scheurwijdte	178
6.2.5	Vervorming met kruip	179
6.2.6	Optimalisatie	180
6.2.7	Conclusie	181
6.3	Staal	181
6.3.1	Sterkte	181
6.3.2	Stabiliteit	182
6.3.3	Vervorming	182
6.3.4	Optimalisatie	183
6.3.5	Conclusie	183
6.4	Hout	183
6.4.1	Sterkte	183
6.4.2	Stabiliteit	184
6.4.3	Vervormingen	185
6.4.4	Optimalisatie	185
6.4.5	Conclusie	186
7	Conclusie	187
	Literatuurlijst	191
	Bijlagen	193

Lijst van tabellen

2.1	Overzicht bestaande softwareprogramma's	50
3.1	Partiële factoren voor belastingen	59
3.2	Belastingenwaarden voorbeeld analyse	63
3.3	Combinaties voor UGT-B	64
3.4	Combinaties voor BGT-K	64
3.5	Combinaties voor BGT-F	64
3.6	Combinaties voor BGT-Q	65
3.7	Reactiekrachten voorbeeld	68
3.8	Interne krachten voorbeeld	69
3.9	Betontypes	75
3.10	Duurzaamheidseisen	76
3.11	Waarden voor $c_{\min, \text{dur}}$ voor constructieklasse S4	76
3.12	Maximale spanningen in bruikbaarheidsgrenstoestand	84
3.13	Aanbevolen waarden voor de imperfectiefactor α_{LT}	92
3.14	Waarden voor k_{mod}	96
3.15	Waarden voor k_{def}	101

Lijst van figuren

2.1	Balk modelleren in Diamonds	33
2.2	Scherf steunpunten selecteren in Diamonds	33
2.3	Resultaten in Diamonds	34
2.4	Scherf berekeningen opvragen in Diamonds	35
2.5	Modelleren van een balk in SCIA Engineer	36
2.6	Resultaten in SCIA Engineer	37
2.7	Modelleren van wapening in beton in SCIA Engineer	38
2.8	Modelleren ConCrete	40
2.9	Resultaten in ConCrete	40
2.10	Wapeningsplan in ConCrete Plus	40
2.11	Berekening beton in Tekla Tedds	42
2.12	Modelleren in Tekla Tedds	42
2.13	Wapening in Tekla Tedds	43
2.14	Modelleren van steunpunten in AxisVM	44
2.15	Resultaten in AxisVM	45
2.16	Balk modelleren in programma bachelorproef	47
2.17	Modelleren in applicatie masterproef	48
2.18	Resultaten in applicatie masterproef	49
3.1	Aangenomen assenstelsel	55
3.2	Tekenconventie voor dwarskracht en moment	55
3.3	Visuele voorstelling van de methode van de partiële factoren	57
3.4	Tekenconventie eindige-elementenmethode	60
3.5	Equivalenten knooppunten	61
3.6	Configuratie voorbeeld analyse	63
3.7	Configuratie voorbeeld analyse met ingevulde waarden	65
3.8	Knooppunten voorbeeld	66
3.9	Schets wapening doorsnede	71
3.10	Parabool-rechthoekdiagram voor beton onder druk	73
3.11	Bi-lineair spanning-rekdiagram voor betonstaal	73
3.12	Maximale breedte-dikte verhoudingen voor op druk belaste onderdelen	89
3.13	Verschillende bijdragen doorbuiging hout	100
4.1	Tabel voor interne krachten en vervormingen	105
4.2	Tabel voor analyse wapening	128
4.3	Tabel grootheden doorsnede	129
4.4	Schematisering beugel	146

4.5	Tabel voor beugels	146
4.6	Schematisering continue langswapening	147
4.7	Afwijking langswapening door kromming beugel	147
4.8	Schematisering extra langswapening	148
5.1	Scherf voor geometrie en steunpunten	160
5.2	Scherf voor belastingsgevallen, belastingen en belastingscombinaties	161
5.3	Scherf voor configuratie beton	162
5.4	Scherf voor betondekking te bepalen	163
5.5	Scherf voor kruipcoëfficiënt te bepalen	163
5.6	Scherf voor configuratie staal	164
5.7	Scherf voor configuratie hout	165
5.8	Voorbeeld van resultaten belastingswaarden	166
5.9	Resultaten momenten en controle langswapening	167
5.10	Optimalisatiescherf voor beton	168
5.11	Optimalisatiescherf voor staal	168
5.12	Optimalisatiescherf voor hout	169
5.13	Voorbeeld berekeningen beton	169
5.14	Voorbeeld berekeningen hout	170
6.1	Schets configuratie uitgewerkt voorbeeld	171
6.2	Momentendiagram van ontwikkeld programma	172
6.3	Momentendiagram van Diamonds	172
6.4	Dwarskrachtendiagram van ontwikkeld programma	173
6.5	Dwarskrachtendiagram van Diamonds	173
6.6	Vervormingen van ontwikkeld programma	174
6.7	Vervormingen van Diamonds	174
6.8	Benodigde langswapening volgens ontwikkeld programma	175
6.9	Benodigde langswapening volgens Diamonds	175
6.10	Benodigde dwarskrachtwapening volgens ontwikkeld programma	176
6.11	Benodigde dwarskrachtwapening volgens Diamonds	176
6.12	Berekeningen van sectie grootheden met kruip	178
6.13	Berekeningen van spanningen in BGT-Q	178
6.14	Resultaten scheurwijdte in ontwikkeld programma	179
6.15	Resultaten scheurwijdte in Diamonds	179
6.16	Resultaten vervormingen met kruip in ontwikkeld programma	179
6.17	Resultaten vervormingen met kruip in Diamonds	180
6.18	Resultaat controles wapening na optimalisatie	180
6.19	Resultaten sterkte in ontwikkeld programma	181
6.20	Resultaten sterkte in Diamonds	181
6.21	Resultaten stabiliteit in ontwikkeld programma	182
6.22	Resultaten stabiliteit in ontwikkeld Diamonds	182
6.23	Resultaat optimalisatie in ontwikkeld programma	183
6.24	Resultaat optimalisatie in Diamonds	183
6.25	Resultaten sterkecontrole in ontwikkeld programma	183
6.26	Resultaten sterkecontrole in Diamonds	184
6.27	Resultaten sterkecontrole in ontwikkeld programma	184

6.28 Resultaten stabiliteitcontrole in Diamonds	184
6.29 Resultaten vervorming met kruip in ontwikkeld programma	185
6.30 Resultaten vervorming met kruip in Diamonds	185
6.31 Resultaten geoptimaliseerde doorsnede	186

Verklarende woordenlijst

Anisotropisch	Een materiaal is anisotropisch wanneer de materiaaleigenschappen afhankelijk zijn van de richting.
Blijvende ontwerpsituatie in uiterste grenstoestand (UGT-B)	Een situatie van onbepaalde duur waarbij constructies worden ontworpen met het oog op hun mogelijke falen. Hierbij wordt rekening gehouden met de sterkte van de constructie en met instabiliteitsverschijnselen zoals knik en kip.
Bruikbaarheidsgrenstoestand (BGT)	Een toestand waarin de bruikbaarheid of het uiterlijk van de constructie een bepaalde grens overschrijdt.
Combinatiefactor	Een factor die wordt toegepast bij het opstellen van verschillende belastingscombinaties. De factor wordt altijd vermenigvuldigd met de karakteristieke waarde van een veranderlijke belasting om deze te reduceren. Er zijn drie verschillende combinatiefactoren, aangeduid als ψ_0 , ψ_1 en ψ_2 .
Frequente combinatie voor bruikbaarheidsgrenstoestand (BGT-F)	Een combinatie die rekening houdt met de kans dat meerdere belastingen tegelijkertijd optreden, maar niet allemaal hun maximale waarde bereiken. Deze combinatie is bedoeld voor gebruik in omkeerbare grenstoestanden.
Gebruikersinterface	Het beeld dat de gebruiker ziet en gebruikt om het programma te besturen. Dit bevat alle visuele componenten zoals knoppen, schermen en andere.
Isotropisch	Een materiaal is isotropisch wanneer de materiaaleigenschappen hetzelfde zijn in alle richtingen.
Karakteristieke combinatie voor bruikbaarheidsgrenstoestand (BGT-K)	Een specifieke combinatie van belastingfactoren waarbij wordt aangenomen dat bepaalde belastingen hun maximale waarde bereiken, maar niet noodzakelijk allemaal tegelijkertijd. Deze combinatie is bedoeld voor gebruik in onomkeerbare grenstoestanden.
Net	De verzameling van posities die elementenranden vertegenwoordigen in een eindige-elementenmethode.
Partiële factor	Een factor die de karakteristieke waarde verhoogt of verlaagd om rekening te houden met onzekerheden.
Quasi-blijvende combinatie voor bruikbaarheidsgrenstoestand (BGT-Q)	Een combinatie die rekening houdt met de impact van langdurige en variabele belastingen op een constructie, zonder dat alle belastingen hun maximum bereiken. Dit wordt toegepast bij grenstoestanden die lang aanhouden.
Uiterste grenstoestand (UGT)	Een toestand waarbij het falen van de constructie optreedt.

Symbolen

Latijnse hoofdletters

A	oppervlakte
A_c	oppervlakte van beton
$A_{c,eff}$	effectieve oppervlakte van beton
A_{ct}	oppervlakte van beton in de trekzone
A_s	oppervlakte van wapening
A_{s1}	oppervlakte van primaire wapening
A_{s2}	oppervlakte van secundaire wapening
A_{sw}	oppervlakte van dwarskrachtwapening
A_v	werkzame afschuifoppervlak
C_1	constante die de belasting en het steunpuntgedrag in rekening brengt
C_2	constante die de belasting en het steunpuntgedrag in rekening brengt
E	belastingeffect; elasticiteitsmodulus
E_c	elasticiteitsmodulus van beton
E_{cm}	secans-elasticiteitsmodulus van beton
E_d	rekenwaarde van belastingseffect
E_s	elasticiteitsmodulus van staal
$E_{0,05}$	5-percentielwaarde van de elasticiteitsmodulus evenwijdig aan de vezelrichting
F	belasting
F_d	rekenwaarde van de belasting
G	blijvende belasting; glijdingsmodulus
G_{mean}	gemiddelde waarde van de glijdingsmodulus evenwijdig aan de vezelrichting
G_k	karakteristieke waarde van een blijvende belasting
$G_{0,05}$	5-percentielwaarde van de glijdingsmodulus evenwijdig aan de vezelrichting
I	traagheidsmoment
I_t	torsieconstante
I_{tor}	torsietraagheidsmoment
I_y	traagheidsmoment om de y-as
I_z	traagheidsmoment om de z-as
I_w	de krommingsconstante is
K_{def}	modificatiefactor die rekening houdt met het vochtgehalte
K_{mod}	modificatiefactor die rekening houdt met de belastingsduur en het vochtgehalte

L	lengte
M	moment
M_{cr}	kritisch moment
$M_{\text{c,Rd}}$	rekenwaarde van de weerstand van de doorsnede tegen buiging
M_{Ed}	rekenwaarde van het optredend moment
$M_{\text{pl,Rd}}$	rekenwaarde van de weerstand van de doorsnede tegen buiging op basis van de plastische rekenmethode
P_{k}	karacteristieke waarde van de voorspankracht
Q	veranderlijke belasting
Q_{k}	karacteristieke waarde van een veranderlijke belasting
R	weerstand
R_{d}	rekenwaarde van de weerstand
R_{k}	karacteristieke waarde van de weerstand
RH	relatieve vochtigheid
S	statisch moment
$S_{\text{r,max}}$	maximale scheurafstand
V	dwarskracht
$V_{\text{c,Rd}}$	rekenwaarde van de weerstand van de doorsnede tegen afschuiving
V_{Ed}	rekenwaarde van de dwarskracht
$V_{\text{Rd,s}}$	rekenwaarde van weerstand van dwarskrachtwapening tegen dwarskracht
W	weerstandsmoment; doorbuiging
W_{inst}	onmiddellijke doorbuiging
$W_{\text{inst,G}}$	onmiddellijke doorbuiging enkel door blijvende belastingen
$W_{\text{inst,Q}_1}$	onmiddellijke doorbuiging enkel door de overheersende veranderlijke belasting
$W_{\text{inst,Q}_i}$	onmiddellijke doorbuiging enkel door niet-overheersende veranderlijke belastingen
W_{fin}	uiteindelijke doorbuiging
$W_{\text{fin,G}}$	uiteindelijke doorbuiging enkel door blijvende belastingen
$W_{\text{fin,Q}_1}$	uiteindelijke doorbuiging enkel door de overheersende veranderlijke belasting
$W_{\text{fin,Q}_i}$	uiteindelijke doorbuiging enkel door niet-overheersende veranderlijke belastingen
W_{pl}	platisch weerstandsmoment
X	materiaaleigenschap
X_{d}	rekenwaarde van de materiaaleigenschap
X_{k}	karacteristieke waarde van de materiaaleigenschap

Latijnse kleine letters

a	factoren
$a_{\perp}$	doorbuiging voor gescheurde toestand bij beton
$a_{ }$	doorbuiging voor niet-gescheurde toestand bij beton
b	breedte
c	betondekking
$c_{\min}$	minimale betondekking
$c_{\min,b}$	minimale betondekking op basis van aanhechtingseisen
$c_{\min,dur}$	minimale betondekking op basis van milieu-omstandigheden
d	verplaatsing; nuttige hoogte
d_1	afstand tussen zwaartepunt primaire wapening en uiterste vezel beton
d_2	afstand tussen zwaartepunt secundaire wapening en uiterste vezel beton
f_{cd}	rekenwaarde van druksterkte van beton
f_{ck}	karacteristieke waarde van druksterkte van beton
f_{cm}	gemiddelde druksterkte van beton
f_{ctm}	gemiddelde treksterkte van beton
$f_{ct,eff}$	gemiddelde treksterkte van beton op het tijdstip waarop voor het eerst scheuren kunnen worden verwacht
f_y	vloeigrens van staal
f_{yd}	rekenwaarde van treksterkte van staal
f_{yk}	karacteristieke waarde van treksterkte van staal
$f_{m,k}$	karacteristieke rekenwaarde van buigsterkte
$f_{m,d}$	rekenwaarde van buigsterkte
$f_{m,y,d}$	rekenwaarde van buigsterkte om y-as
$f_{m,z,d}$	rekenwaarde van buigsterkte om z-as
$f_{v,d}$	rekenwaarde van de afschuifsterkte
g	waarde voor blijvende verdeelde belasting
h	hoogte
h_0	theoretische dikte van de balk
k	coëfficiënten
k_{crit}	reductiefactor die rekening houdt met kip
l_{bd}	verankeringslengte
l_{ef}	meewerkende balklengte
n	aantal wapeningsstaven
n_1	aantal primaire wapeningsstaven

n_2	aantal secundaire wapeningsstaven
q	waarde voor veranderlijke verdeelde belasting
r_{db}	straal van de doorndiameter
r_s	straal van een wapeningsstaaf
s	zwaartepunt; tussenafstand
t	tijd
t_0	initieel tijdstip
u	omtrek van de doorsnede dat in contact komt met de buitenlucht
w_k	scheurwijdte
x	betondrukhoogte
y	afstand van het punt waar de spanning wordt berekend tot de neutrale lijn
z_g	de afstand tussen het aangrijpingspunt van de belasting en het afschuifmiddelpunt

Griekse hoofdletters

Δ_{dev}	waarde voor betondekking in functie van uitvoeringstolerantie
Φ_{LT}	waarde voor het bepalen van de reductiefactor χ_{LT}

Griekse kleine letters

α	verhouding elasticiteitsmodulus staal op elasticiteitsmodulus beton
α_{cc}	factor die rekening houdt met de manier waarop de belasting werkt en de mogelijke langetermijneffecten op de druksterkte van beton
α_{LT}	imperfectiefactor
β	coëfficiënt is die rekening houdt met de belastingsduur of herhaalde belasting op de gemiddelde rek
$\beta(f_{cm})$	factor die rekening houdt met betonsterkte voor kruipcoëfficiënt
$\beta(t_0)$	factor die rekening houdt met ouderdom beton voor kruipcoëfficiënt
γ	partiële factor
γ_c	partiële factor voor beton
γ_F	partiële belastingsfactor
γ_G	partiële factor voor blijvende belasting
γ_M	partiële materiaalfactor
γ_{M0}	partiële factor voor de weerstand van doorsneden bij staal
γ_{M1}	partiële factor voor de weerstand op het gebied van stabiliteit bij staal
γ_P	partiële factor voor voorspankrachten

γ_Q	partiële factor voor veranderlijke belasting
γ_s	partiële factor voor betonstaal
ε	rek
ε_c	rek van beton
ε_{c2}	rekgrens van beton waarbij de spanning in beton constant is indien deze grens overschreden is
ε_{cm}	gemiddelde betonrek tussen scheuren
ε_s	rek van staal
ε_{sm}	gemiddelde rek van de wapening
ε_{s1}	rek van primaire wapening
ε_{s2}	rek van secundaire wapening
ζ	verhouding hefboomsarm in doorsnede en de nuttige hoogte; verdelingsfactor is die rekening houdt met "tension stiffening" in een doorsnede
θ	rotatie; schoorhoek
$\bar{\lambda}_{LT}$	relatieve slankheid voor kippen
$\lambda_{rel,m}$	relatieve slankheid
μ_d	gereduceerd moment
ν	coëfficiënt van Poisson
ν_1	sterktereductiefactor voor beton met dwarskrachtscheuren
ρ_{eff}	wapeningsverhouding; reductiefactor voor combinatie buiging en dwarskracht
ρ_w	dwarskrachtwapeningspercentage
$\rho_{w,min}$	minimale dwarskrachtwapeningspercentage
σ	spanning
σ_{s1}	spanning in primaire wapening
σ_{s2}	spanning in secundaire wapening
$\sigma_{m,d}$	rekenwaarde van de buigspanning
$\sigma_{m,y,d}$	rekenwaarde van de buigspanning om y-as
$\sigma_{m,z,d}$	rekenwaarde van de buigspanning om z-as
$\sigma_{m,crit}$	kritische buigspanning voor kip
σ_c	spanning in beton
σ_s	spanning in staal
τ_d	rekenwaarde van de schuifspanning
ϕ_b	diameter van beugel
ϕ_{eq}	equivalente diameter
ϕ_s	diameter van wapeningsstaaf
ϕ_{s1}	diameter van primaire wapeningsstaaf

ϕ_{s2}	diameter van secundaire wapeningsstaaf
$\varphi_{(t,t_0)}$	kruipcoëfficiënt voor beton
φ_0	theoretisch kruipcoëfficiënt voor beton
φ_{RH}	factor voor invloed relatieve vochtigheid op kruipcoëfficiënt beton
χ_{LT}	reductiefactor voor kip
ψ_0	factor voor de combinatie waarde van een veranderlijke belasting
ψ_1	factor voor de frequente waarde van een veranderlijke belasting
ψ_2	factor voor de quasi-blijvende waarde van een veranderlijke belasting

Abstract

Deze masterproef bouwt voort op een bachelorproef uit het academiejaar 2022-2023 geschreven door P. Gielkens, J. Jans en J. Sanen, waarin een gebruiksvriendelijk programma voor het analyseren en dimensioneren van betonnen balken werd ontwikkeld. Het doel van de huidige masterproef was het optimaliseren en uitbreiden van dat programma naar stalen en houten balken, met de nadruk op het creëren van een waardevol hulpmiddel voor studenten bouwkunde.

Het programma moest aan drie eisen voldoen. Ten eerste moest het nauwkeuriger zijn, aangezien het bestaande programma vereenvoudigde formules toepaste. Ten tweede moest het gebruikers de mogelijkheid bieden om berekeningen op te vragen voor beter begrip. Ten derde moest de gebruikersinterface eenvoudig en intuïtief zijn.

Het programma werd ontwikkeld in Python en gebruikt de eindige-elementenmethode (EEM) voor de analyse van balken onder verschillende belastingscombinaties en grenstoestanden. De ontwerpprincipes, formules en controles zijn gebaseerd op de relevante Eurocodes voor elk materiaal.

Het programma controleert gewapende betonnen balken op sterkte, toelaatbare spanningen, scheurwijdte en doorbuiging. Daarnaast kan het de wapening en afmetingen van de balk optimaliseren. Voor stalen en houten balken controleert het programma op sterkte, stabiliteit en doorbuiging en kan het ook de doorsnede optimaliseren. Deze studie biedt een basis voor verder onderzoek waarbij de gebruikersinterface verder kan worden verbeterd en het aantal mogelijke berekeningen kan worden uitgebreid.

Abstract in English

This master's thesis builds upon a bachelor's thesis from the 2022-2023 academic year written by P. Gielkens, J. Jans and J. Sanen, which developed a user-friendly programme for analysing and designing concrete beams. The aim of the current master's thesis was to optimise and extend that programme to also include steel and timber beams, with an emphasis on creating a valuable tool for civil engineering students.

The programme had to meet three requirements. First, it had to be more accurate, since the existing programme applied simplified formulas. Second, it had to allow users to access calculations for better understanding. Third, the user interface had to be simple and intuitive.

The programme was developed in Python and uses the finite element method (FEM) to analyse beams under different load combinations and limit states. The design principles, formulas and checks are based on the relevant Eurocodes for each material.

The programme checks reinforced concrete beams for strength, allowable stresses, crack width and deflection. In addition, it can optimise the reinforcement and dimensions of the beam. For steel and timber beams, the programme checks for strength, stability and deflection and can also optimise the cross-section. This study provides a basis for further research in which the user interface can be further improved and the number of possible calculations expanded.

Hoofdstuk 1

Inleiding

Dit hoofdstuk biedt een uiteenzetting van de inhoud van deze masterproef. Het behandelt de situering van het onderzoek, de probleemstelling die wordt aangepakt, de specifieke doelstellingen, de limitaties van het onderzoek en de toegepaste methodologie. Ten slotte wordt er een vooruitblik gegeven op de komende hoofdstukken.

1.1 Situering

Balken vormen een essentieel onderdeel van de draagstructuur van gebouwen. Ze hebben de cruciale taak om krachten, zoals die boven deuropeningen en raamkozijnen, over te brengen en vloerplaten te ondersteunen. Voor bouwkundige ingenieurs is een diepgaand begrip van de mechanica en berekeningen die bij deze balken betrokken zijn, dan ook van groot belang. Daarnaast moet worden opgemerkt dat balken kunnen variëren in materiaal, zoals beton, staal of hout, waarbij elk type onderhevig is aan specifieke ontwerpnormen en voorschriften, zoals, respectievelijk, Eurocode 2 [1], Eurocode 3 [2] en Eurocode 5 [3].

Hoewel er al bestaande softwareprogramma's zijn die in staat zijn om volledige structuren en gebouwen te berekenen, blijkt dat wanneer er specifiek gefocust wordt op geautomatiseerde berekeningen voor balken, uitgevoerd door eindige-elementenprogramma's zoals SCIA Engineer [4], Diamonds [5] en AxisVM [6], elk van deze programma's zijn eigen sterke en zwakke punten heeft. In een onderwijscontext, waarbij er aan ingenieursstudenten dient worden toegelicht wat de impact is van bijvoorbeeld het toevoegen van steunpunten, is het gebruik van voorgaande programma's evenmin optimaal (zie probleemstelling en literatuurstudie).

In het academiejaar 2022-2023 werd een vergelijkbaar Python-gebaseerd programma ontwikkeld als onderdeel van een bachelorproef [7]. Het is echter belangrijk op te merken dat dit programma nog optimalisatie behoeft voordat het volledig operationeel kan worden ingezet, zowel in een onderwijs- als praktijkcontext. Deze masterproef beoogt deze Python-software verder te ontwikkelen om iso- en hyperstatische balken snel en in overeenstemming met de relevante normen te analyseren en te ontwerpen. Bovendien zal het programma de mogelijkheid bieden aan gebruikers om berekeningen op te vragen. Op deze manier zal het programma inzichten verschaffen die studenten kunnen helpen een beter begrip te krijgen van deze balken.

1.2 Probleemstelling

Bestaande softwareprogramma's zoals SCIA Engineer [4] en BuildSoft Diamonds [5] bieden de mogelijkheid om stabiliteitsberekeningen en -controles uit te voeren voor complete gebouwen. Dit vereist echter de modellering van een volledig 3D-gebouw binnen het programma, wat een aanzienlijke tijdsinvestering met zich meebrengt. Hetzelfde geldt voor het controleren van individuele balken, waarbij verschillende stappen moeten worden doorlopen. Ten eerste moet er een project worden aangemaakt, vervolgens moeten de geometrische en andere eigenschappen van de balk worden ingevoerd, daarna moeten de randvoorwaarden en belastingen worden gespecificeerd, en tot slot moeten belastingscombinaties worden gedefinieerd. Elk van deze stappen vereist het gebruik van verschillende knoppen en opent nieuwe vensters, wat resulteert in een omslachtig en tijdrovend proces.

Naast het tijdsintensieve proces voor individuele balkberekeningen, voldoen de bestaande softwareprogramma's ook niet aan de behoeften van studenten als het gaat om educatieve waarde. Het huidige proces biedt studenten weinig inzicht, omdat het simpelweg een controle uitvoert die kan resulteren in het al dan niet voldoen van de balk aan berekeningsnormen. In het geval van een falende berekening, wordt er vaak niet gedetailleerd aangegeven waarom de balk is gefaald. Dit gebrek aan transparantie en feedback maakt het voor studenten moeilijk om de onderliggende concepten volledig te begrijpen.

De software ontwikkeld tijdens de voormelde bachelorproef [7] vertoont enkele kritieke problemen die moeten worden aangepakt. Ten eerste voert het programma de balkanalyse incorrect uit door geen rekening te houden met belastingscombinaties en grenstoestanden zoals voorgeschreven in Eurocode 0 [8]. Bovendien is het alleen mogelijk om balken te analyseren met een uniform verdeelde belasting die over de gehele lengte van de balk aangrijpt. Voor het ontwerp van betonnen balken schiet de software ook tekort: het berekent enkel de langswapening en dwarswapening met vereenvoudigde formules die de benodigde hoeveelheid wapening schatten. Andere controles, zoals controle op optredende spanningen, scheurwijdte en doorbuiging, zijn niet mogelijk. Dit kan leiden tot onnauwkeurige resultaten die niet voldoen aan de normatieve eisen van de constructie-industrie. Daarnaast is er weinig aandacht besteed aan de gebruikersinterface, wat resulteert in een gebrek aan gebruiksvriendelijkheid. Het optimaliseren van de gebruikersinterface is essentieel om de software toegankelijker te maken voor zowel professionals als studenten. Tot slot beperkt de software zich momenteel tot het uitvoeren van berekeningen voor betonnen balken, wat de toepasbaarheid ervan beperkt. Het is van groot belang om de software uit te breiden naar andere materialen om tegemoet te komen aan de diverse behoeften van de bouwsector.

1.3 Doelstellingen

Het hoofddoel van deze masterproef is het ontwikkelen van een softwareprogramma dat in staat is om zowel iso- als hyperstatische balken op een snelle en nauwkeurige wijze te analyseren en te ontwerpen, waarbij de resultaten op een visueel begrijpelijke manier gepresenteerd worden zodat dit programma als hulpmiddel kan dienen voor studenten. Om dit algemene doel te verfijnen, zijn de volgende specifieke doelstellingen vastgesteld.

1. **Materialen en Profielen:** Het programma moet flexibel zijn en de mogelijkheid bieden om balken van verschillende materialen, zoals beton, staal en hout, te berekenen. Gebrui-

kers moeten kunnen kiezen uit een reeks voorgeprogrammeerde materialen. Daarnaast zal het programma een bibliotheek bevatten met een aantal profielen waaruit de gebruiker moet kiezen voor de berekening van stalen balken.

2. **Normen en Controles:** Het programma moet voldoen aan de relevante normen en voorschriften, met name de Eurocodes, om ervoor te zorgen dat de berekeningen geschikt zijn voor professioneel gebruik. Hierbij worden zowel UGT (uiterste grenstoestand) als BGT (bruikbaarheidsgrenstoestand) controles uitgevoerd en wordt speciale aandacht besteed aan instabiliteitsverschijnselen zoals kip bij houten en stalen balken.
3. **Efficiëntie en Gebruiksvriendelijkheid:** Met ‘snel’ wordt bedoeld dat het programma een structuur moet hebben die zowel efficiënt als gemakkelijk te begrijpen is voor gebruikers. Dit houdt in dat het aantal stappen en handelingen om tot de berekeningen te komen tot een minimum wordt beperkt. Het programma moet een intuïtieve en gestroomlijnde gebruikersinterface bieden, waarin alle benodigde informatie duidelijk wordt gepresenteerd.
4. **Visuele Representatie:** Het programma zal berekeningen visueel weergeven om het leerproces te verbeteren, vooral voor studenten. Dit omvat het visualiseren van de positie van wapening in balken, interne krachtdiagrammen, spanningen, vervormingen en de mogelijkheid om berekeningen op te vragen om alles beter te begrijpen.

1.4 Methode

In deze masterproef is Python [9] gekozen als de programmeertaal voor de ontwikkeling van de softwarecode. Python is een veelzijdige opensourceprogrammeertaal die vrij beschikbaar is [10]. De keuze voor Python is gebaseerd op het feit dat het een programmeertaal is met een eenvoudige syntaxis, die gemakkelijk te leren is, zelfs voor beginners. Daarnaast bevat het programma een database die beheerd wordt met DB Browser [11].

Voorafgaand aan het dimensioneren van balken is het noodzakelijk om de interne krachten en vervormingen te berekenen, een proces dat de analyse van een balk genoemd wordt. Deze berekeningen worden uitgevoerd aan de hand van de eindige-elementenmethode (EEM). De methode verdeelt de balk in een eindig aantal kleinere elementen, waarbij elk uiteinde van zo’n element wordt voorgesteld door een knoop. Elk element wordt voorgesteld door een elementstijfheidmatrix. Deze elementstijfheidmatrices worden samengevoegd tot een globale stijfheidmatrix [12]. Vanuit deze globale matrix kunnen de verplaatsingen en rotaties voor elke knoop worden afgeleid. Vervolgens worden deze verkregen verplaatsingen en rotaties gebruikt om in elke elementmatrix de dwarskrachten, momenten en doorbuiging te berekenen bij elke knoop.

De balk kan onderworpen worden aan diverse blijvende en veranderlijke belastingsgevallen. Conform Eurocode 0 [8] dienen daarvoor verschillende belastingscombinaties te worden opgesteld. Een belastingscombinatie bestaat uit een samenstelling van belastingsgevallen met bijbehorende partiële combinatiefactoren, waarvan de waarden te raadplegen zijn in Eurocode 0 [8]. Deze norm maakt tevens onderscheid tussen belastingscombinaties voor uiterste grenstoestanden en die voor bruikbaarheidsgrenstoestanden. De specifieke belastingscombinaties worden verder uitvoerig toegelicht in Sectie 3.1. De balk dient bijgevolg ontworpen en gecontroleerd te worden om te voldoen aan deze diverse grenstoestanden en belastingscombinaties.

De UGT-controle omvat een analyse van de interne krachten en de bijbehorende spanningen, die vervolgens worden vergeleken met de materiaalcapaciteit. Hierdoor wordt de balk ontworpen met een focus op zijn sterkte. De weerstand tegen het instabiliteitsverschijnsel kip wordt ook gecontroleerd in UGT. Daarnaast ondergaat de BGT-controle een evaluatie op basis van de interne krachten en de doorbuiging van de balk. Hierbij worden zowel onmiddellijke vervormingen als latere vervormingen, zoals kruip, meegenomen in de beoordeling. Voor betonnen balken wordt bovendien een extra controle uitgevoerd om de scheurwijdte van het beton te beoordelen. Deze BGT-controle houdt rekening met zowel de conformiteit aan de gebruikerseisen als het esthetische aspect van de balk.

Voor het ontwerpen van de gebruikersinterface in Python wordt gebruik gemaakt van Qt Designer [13]. Deze tool vereenvoudigt het proces door het mogelijk te maken om knoppen, tekstvakken en andere elementen eenvoudig naar een ontwerpveld te slepen. Deze elementen worden automatisch omgezet in Python-code. Om de interface functioneel te maken, wordt extra code toegevoegd die de gewenste interacties en reacties bij het indrukken of wijzigen van elementen regelt. Ter voorbereiding op het ontwerp van de gebruikersinterface zijn bestaande softwareprogramma's geanalyseerd om hun sterke en zwakke punten te evalueren (zie literatuurstudie). Op basis van deze analyse is een interface ontwikkeld die tegemoetkomt aan de eisen van deze masterproef.

1.5 Limitaties

Om de scope van de masterproef te definiëren, zijn er enkele beperkingen aan het programma gesteld. Ten eerste kunnen alleen verticale krachten en momenten rond de z-as worden aangebracht (zie Sectie 3.1 voor de tekenconventie). Dit betekent dat er nooit een normaalkracht optreedt en dat de doorsnede altijd alleen in de y-richting wordt belast. Een andere beperking is dat het programma geen tweede-orde berekeningen kan uitvoeren. Daarnaast kunnen alleen belastingscombinaties voor blijvende ontwerpsituaties worden gemaakt, dus niet voor buitengewone of aardbevingsontwerpsituaties. Dit betekent dat balken niet ontworpen kunnen worden voor deze accidentele belastingen.

Voor betonnen balken is de analyse beperkt tot niet-voorgespannen gewapende balken met een rechthoekige doorsnede. Bij stalen liggers zijn alleen een aantal standaard doorsneden beschikbaar. Houten liggers zijn ook beperkt tot rechthoekige doorsneden. Bovendien blijft de doorsnede constant over de volledige lengte van de balk. Verdere specifieke aannames en beperkingen voor elk materiaal worden toegelicht in het betreffende onderdeel van Hoofdstuk 3.

1.6 Vooruitblik

In het volgende hoofdstuk wordt de literatuurstudie beschreven, waarbij diverse bestaande softwareprogramma's zijn geanalyseerd. Hoofdstuk 3 behandelt de gebruikte formules, methoden en controles voor het analyseren en ontwerpen van balken. Eerst wordt uitgelegd hoe verschillende belastingscombinaties voor verschillende grenstoestanden worden gemaakt en hoe de eindigelementenmethode werkt. Vervolgens worden de formules en berekeningen voor de verschillende materialen besproken. Hoofdstuk 4 richt zich op de implementatie van deze formules in Python en legt de structuur van het programma en bijhorende database uit. Hoofdstuk 5 gaat dieper

in op het ontwerp van de gebruikersinterface. In Hoofdstuk 6 worden drie balken berekend met behulp van het ontworpen programma en vergeleken met dezelfde balken berekend in de eerder besproken softwareprogramma's. Ten slotte biedt Hoofdstuk 7 een conclusie en bespreekt mogelijke uitbreidingsmogelijkheden voor het programma.

Hoofdstuk 2

Literatuurstudie

2.1 Inleiding

Voor de literatuurstudie wordt onderzoek uitgevoerd naar bestaande softwareprogramma's, waarbij elk programma wordt geëvalueerd aan de hand van criteria. Deze criteria omvatten de mogelijkheid tot het uitvoeren van berekeningen met betrekking tot balken, de snelheid van het modelleringsproces, de gebruikersinterface, de educatieve waarde voor studenten en de kostprijs.

Met betrekking tot berekeningen ligt de focus uitsluitend op die met betrekking tot balken, aangezien andere berekeningen niet relevant zijn voor deze masterproef. Specifiek wordt gekeken naar het berekenen van inwendige krachten volgens Eurocode 0 [8] voor een balk met verschillende belastingcombinaties op basis van diverse grenstoestanden. Daarnaast wordt de mogelijkheid van specifieke berekeningen voor beton, staal en hout onderzocht.

Bij het beoordelen van de snelheid van het modelleren wordt gekeken naar de efficiëntie van het proces, waarbij wordt onderzocht of er aanzienlijke stappen nodig zijn voordat resultaten worden verkregen. Ook wordt gekeken naar de beschikbaarheid van eventuele sneltoetsen om veelvoorkomende types snel en effectief te modelleren.

De evaluatie van de gebruikersinterface richt zich op de vanzelfsprekendheid en logische structuur van de software. Daarnaast wordt er gekeken naar hoe gebruiksvriendelijk het programma is voor beginners en de esthetiek.

Bij het beoordelen van de educatieve waarde wordt onderzocht of studenten of personen met beperkte kennis het programma kunnen gebruiken om een beter inzicht te krijgen. Hierbij wordt gekeken naar de visuele presentatie van de resultaten en de mogelijkheid om verduidelijkende berekeningen op te vragen die de resultaten verhelderen.

De prijsanalyse van de software omvat de kosten voor aanschaf of licentie van het softwarepakket, evenals de beschikbaarheid van studentenlicenties.

Het is belangrijk op te merken dat de resultaten van dit onderzoek deels subjectief zijn, aangezien enkel de auteur van deze masterproef, gedurende een beperkte periode, de software's heb onderzocht. Bovendien kunnen bekende programma's hoger scoren omdat de auteur al bekend ben met het gebruik ervan. Een laatste opmerking is dat er enkel studentenversies gebruikt werden tijdens het onderzoek. Dit kan de oorzaak zijn waarom sommige functies niet werkten.

De onderzochte softwareprogramma's omvatten Diamonds [5], SCIA Engineer [4], ConCrete (Plus) [14], Tekla Tedds [15] en AxisVM [6]. Het is belangrijk op te merken dat er naast deze programma's nog andere bestaan, zoals Frilo [16], StruSoft [17], Technosoft [18] en vele anderen. Het is echter niet haalbaar om al deze programma's te onderzoeken binnen de beschikbare tijd. Tot slot wordt ook de bachelorproef [7] die voorloper is van deze masterproef onderzocht en een vergelijkbare masterproef uit het academiejaar 2016-2017. Voor deze laatste werd een app ontwikkeld die eveneens in staat is om hyperstatische balken te berekenen [19].

2.2 Onderzoek

In deze sectie zal elk programma geëvalueerd worden voor de vijf bovenstaande criteria. Eerst zal elk programma apart besproken worden, gevolgd door een tabel die een globaal overzicht biedt.

2.2.1 Diamonds

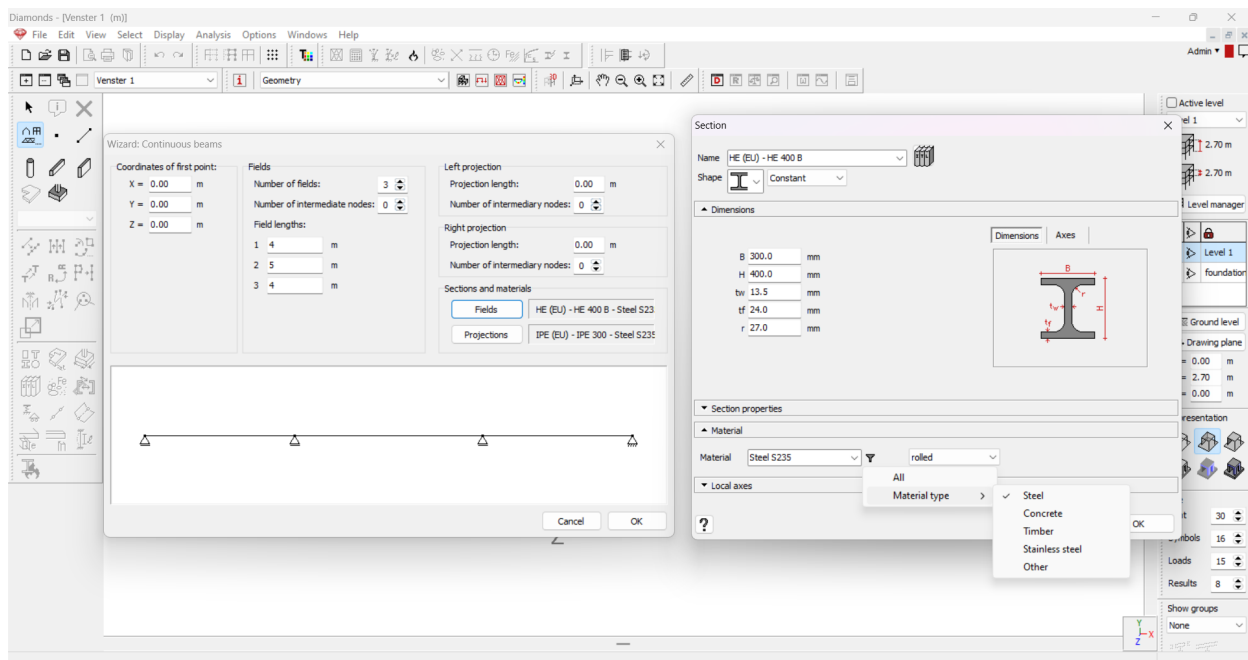
Diamonds is een eindige-elementensoftware ontwikkeld door BuildSoft, een Belgisch bedrijf dat zich specialiseert in het ontwerpen en produceren van snelle en gebruiksvriendelijke software voor de analyse van bouwkundige structuren. De software is gericht op bouwkundig ingenieurs, architecten, aannemers en bouwbedrijven [20].

Mogelijke berekeningen

Binnen Diamonds worden interne krachten, spanningen en verplaatsingen berekend volgens Eurocode 0. Het programma is in staat om op basis van de aangebrachte belastingen automatisch de benodigde belastingcombinaties te genereren. Daarnaast heeft de gebruiker ook de mogelijkheid zelf combinaties aan te maken waarbij hij elke factor zelf kan instellen. Naast het analyseren van een balk kan Diamonds berekeningen uitvoeren voor specifieke materialen volgens hun overeenkomstige Eurocode. Bijvoorbeeld kan het programma voor beton de benodigde hoeveelheid langs- en dwarskrachtswapening berekenen. Bovendien kan Diamonds de scheurwijdte en de doorbuiging op lange termijn van de balk berekenen. Echter kan het programma geen voorstel doen van een te plaatsen wapeningsconfiguratie. Voor stalen en houten balken controleert het programma de sectie op zowel sterkte als stabiliteit en het biedt de mogelijkheid om de sectie te optimaliseren op basis van deze twee criteria.

Modelleren in Diamonds

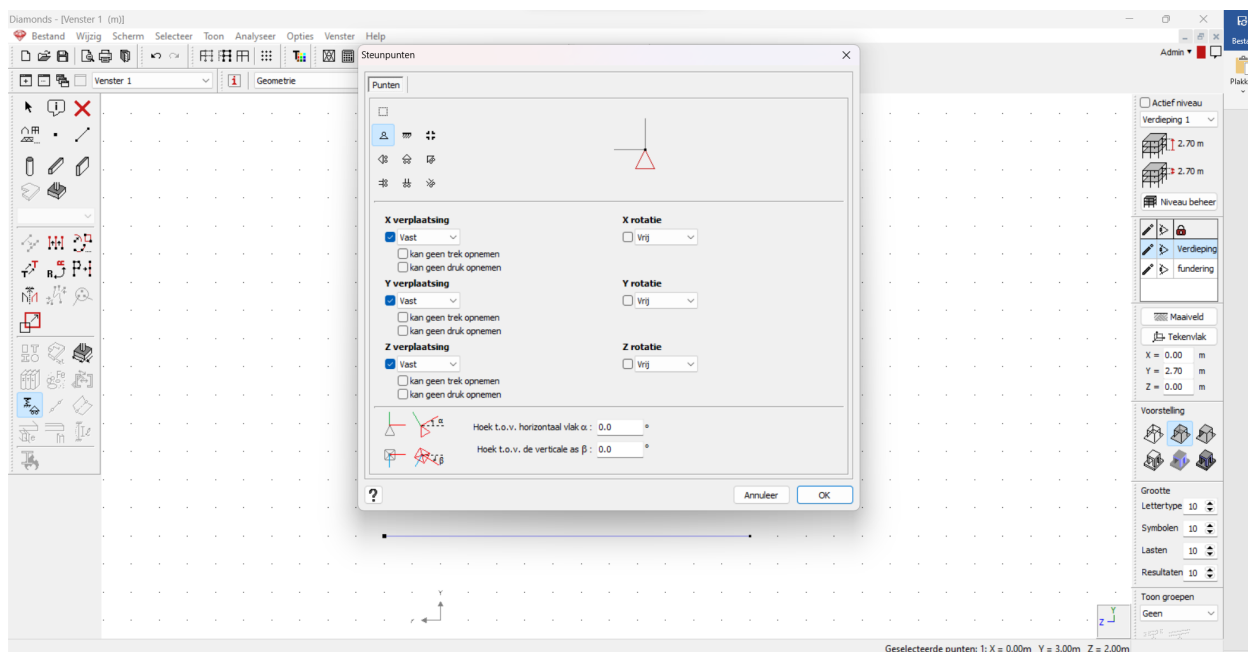
Het modelleren van een balk in Diamonds verloopt efficiënt en rechtstreeks. Het is niet vereist om eerst een project aan te maken, gebruikers kunnen meteen aan de slag met het modelleren. Het programma biedt een snelle manier om een balk te modelleren via een speciale sneltoets. Wanneer deze sneltoets wordt gebruikt, opent zich een nieuw scherm waarin gebruikers het aantal velden kunnen aanduiden en voor elk veld de lengte kunnen specificeren. Ook kunnen gebruikers in dit scherm de sectie en het materiaal van de balk kiezen, waarvoor opnieuw een nieuw venster wordt geopend. Figuur 2.1 toont dit proces. Het venster aan de linkerzijde van Figuur 2.1 is bedoeld om de geometrie van de balk te specificeren, terwijl het geopende venster aan de rechterzijde dient voor kiezen van de doorsnede en het materiaal.



Figuur 2.1: Balk modelleren in Diamonds

De knop met de afbeelding van de drie boeken, die te zien is bovenaan het geopende venster in Figuur 2.1, opent een scherm waarin verschillende vooraf geprogrammeerde profielen zijn opgeslagen. Het materiaal wordt gekozen met behulp van de *dropdownmenu*.

Het programma plaatst automatisch één scharnier en voor alle andere steunpunten een rolop-
legging. De gebruiker kan dit aanpassen door het scherm te openen dat wordt weergegeven in Figuur 2.2.

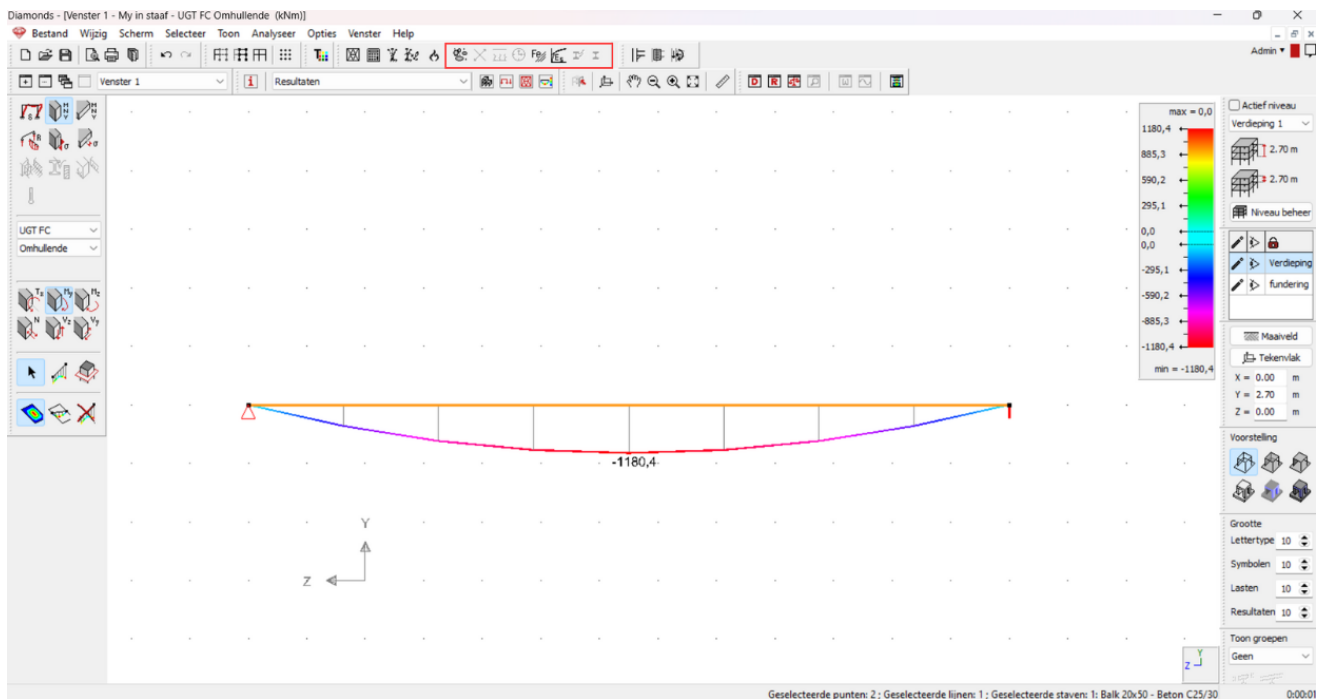


Figuur 2.2: Scherm steunpunten selecteren in Diamonds

Zoals geïllustreerd in Figuur 2.2, kan de keuze voor een oplegging, scharnier of inklemming snel worden gemaakt door de overeenkomstige sneltoets te selecteren. Echter kan de gebruiker ook zelf een steunpunt creëren.

Om belastingen op de balk aan te brengen, moet de gebruiker het dropdownmenu bovenaan, waar in Figuur 2.1 en 2.2 “geometrie” staat, wijzigen naar “lasten”. In dit scherm kunnen diverse belastingen worden toegepast voor verschillende lastengroepen. Standaard worden drie lastengroepen gecreëerd, namelijk eigengewicht, permanente lasten en mobiele lasten. De gebruiker heeft de flexibiliteit om zelf extra lastengroepen te definiëren. Op de balk kunnen belastingen worden toegepast, zoals een verdeelde belasting, een verticale puntlast, een geconcentreerd puntmoment en een verdeeld moment. Ten slotte dient de gebruiker belastingscombinaties te maken of te laten genereren voor de gewilde toestanden.

De laatste stap vereist dat de gebruiker de balk laat berekenen door bovenaan op de knop met het symbool van een rekenmachine te klikken. Het programma stelt een *mesh* oftewel net voor, waarna de gebruiker nog aanvullende instellingen kan aanpassen alvorens de berekening uit te voeren. Een mogelijk beeld van de resultaten wordt weergegeven in Figuur 2.3.



Figuur 2.3: Resultaten in Diamonds

In Figuur 2.3 wordt getoond dat de gebruiker diverse resultaten kan opvragen met behulp van de knoppen aan de linkerkant van het scherm. De gebruiker kan kiezen voor welke belastingsgroep of belastingscombinatie de resultaten worden getoond. De resultaten worden in verschillende kleuren weergegeven, waardoor de variatie in waarden eenvoudig visueel wordt gepresenteerd.

Na het berekenen van de balk is het enkel mogelijk om de doorbuigingen, spanningen en interne krachten op te vragen. Om resultaten te verkrijgen voor het gekozen materiaal moet er eerst een berekening worden uitgevoerd voor het materiaal. Dit kan door op de knoppen bovenaan het scherm te klikken, aangeduid in de rode rechthoek in Figuur 2.3.

Gebruikersinterface

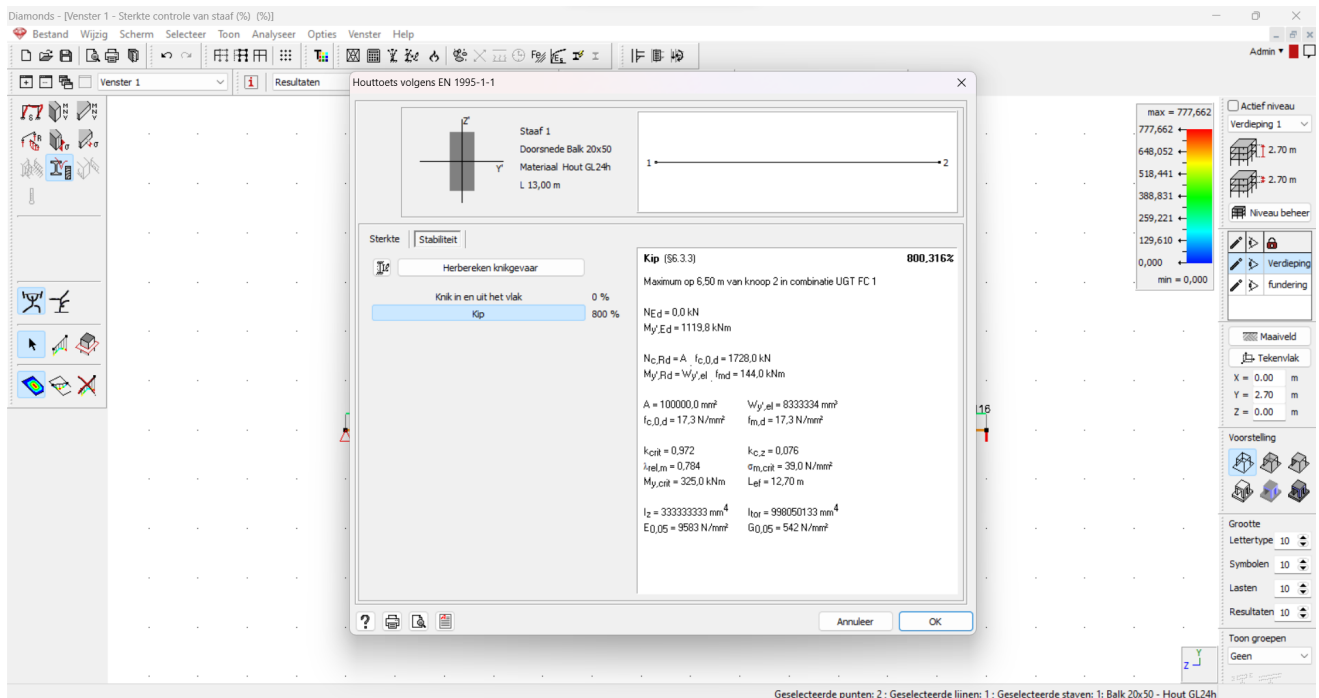
De gebruikersinterface van Diamonds oogt eenvoudig, maar is tegelijkertijd efficiënt en intuïtief. De structuur is helder ingedeeld, waarbij gerelateerde knoppen logisch gegroepeerd zijn en de pictogrammen duidelijk herkenbaar zijn. Mocht een pictogram toch niet duidelijk zijn, dan kan

de gebruiker de muis erover bewegen, waarbij een korte beschrijving verschijnt over de functie van de knop.

De geavanceerde instellingen zitten verscholen achter aparte knoppen, waardoor de interface niet overweldigend is. Dit zorgt ervoor dat het programma gebruikersvriendelijk is voor zowel beginners als professionals.

Educatieve waarde

Na de berekeningen en controles voor stalen en houten balken is het mogelijk om de resultaten op te vragen door tweemaal te klikken op de desbetreffende balk. Het bijbehorende scherm wordt weergegeven in Figuur 2.4.



Figuur 2.4: Scherm berekeningen opvragen in Diamonds

Figuur 2.4 toont dat de houten balk niet voldoet aan de stabiliteitseisen. Deze bijbehorende berekeningen kunnen van aanzienlijke meerwaarde zijn voor studenten om dit resultaat beter te begrijpen, waardoor het een educatieve waarde heeft. Niettemin is het belangrijk op te merken dat niet alle berekeningen opgevraagd kunnen worden. Zo is het bijvoorbeeld niet mogelijk om berekeningen omtrent beton op te vragen.

Prijs Diamonds

De kostprijs voor Diamonds 2D is €129 per maand indien het bedrag jaarlijks wordt betaald en €207 per maand indien er maandelijks wordt betaald [21]. Voor studenten is het wel mogelijk om jaarlijks een gratis studentenlicentie aan te vragen.

2.2.2 SCIA Engineer

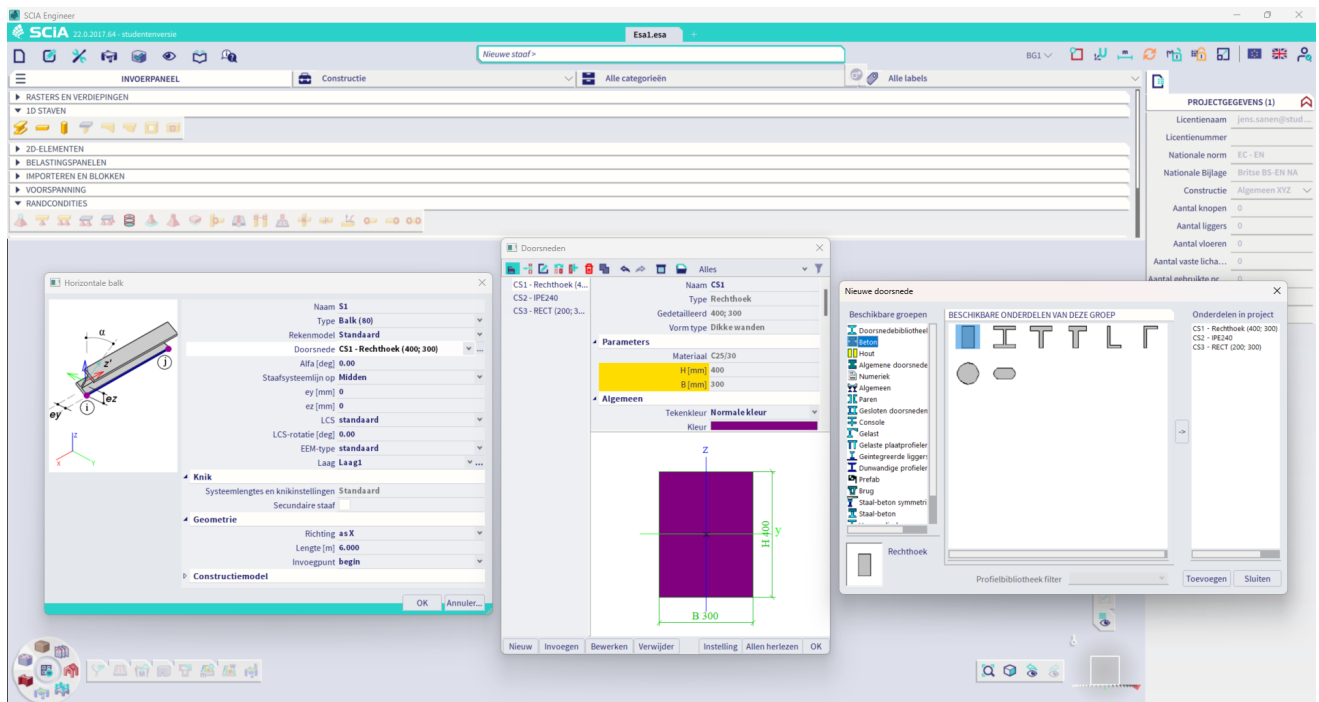
SCIA Engineer is een softwarepakket van SCIA. SCIA is een Belgisch bedrijf gespecialiseerd in het ontwerpen van software voor de analyse van bouwkundige constructies [22].

Mogelijke berekeningen

SCIA Engineer voert berekeningen uit voor interne krachten, verplaatsingen en spanningen in overeenstemming met Eurocode 0. Gebruikers hebben de mogelijkheid om verschillende belastinggevallen en combinaties te definiëren. Het programma biedt ondersteuning voor betonnen, stalen en houten balken en voert voor elk materiaal controles uit, waaronder wapenings-, doorbuigings-, spannings- en scheurwijdtecontroles voor beton. Hoewel het programma geen standaard wapeningsconfiguratie voorstelt, kunnen gebruikers zelf de wapening plaatsen en deze daarna laten controleren. Voor stalen en houten balken voorziet het programma in controles voor zowel de uiterste grenstoestand (sterkte en stabiliteit) als de gebruiksgrenstoestand (doorbuiging). Bovendien kan het programma geplaatste houten en stalen secties optimaliseren om aan de ontwerpeisen te voldoen. Tenslotte kan SCIA Engineer ook berekeningen en controles uitvoeren voor staal-beton liggers maar deze worden niet besproken in deze masterproef.

Modellering in SCIA Engineer

Voordat de gebruiker aan de modellering van een balk kan beginnen, is het noodzakelijk om eerst een project te selecteren of aan te maken. Om een balk te plaatsen, kan de gebruiker de functie “1D-STAAF” of “LIGGER” kiezen bovenaan in de werkbalk. Het verschil tussen deze twee functies is dat bij deze laatste een vaste lengte op voorhand wordt gekozen door de gebruiker en bij de andere wordt de lengte gekozen bij het plaatsen van de balk in het tekenveld. Door te klikken op de functie “LIGGER” wordt het meest linkse venster in Figuur 2.5 geopend. Het dwarsprofiel en het materiaal kunnen worden geselecteerd door op de knop bij doorsnede te drukken, waardoor het middelste venster wordt geopend.



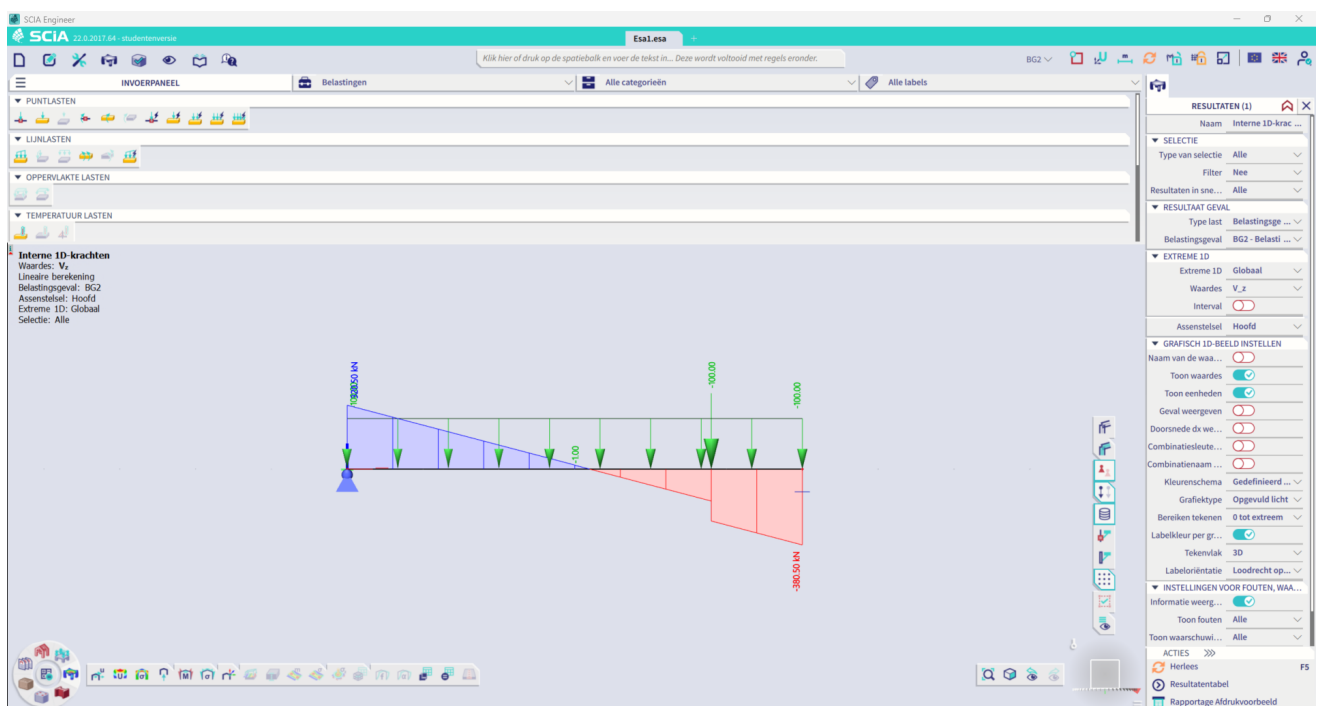
Figuur 2.5: Modelleren van een balk in SCIA Engineer

Zoals te zien is in het rechtse venster in Figuur 2.5, is er een uitgebreide bibliotheek met verschillende profielen waaruit kan worden gekozen. Het materiaal kan, net zoals bij Diamonds, worden geselecteerd door op het dropdownmenu te klikken. Nadat de doorsnede is bepaald, hoeft alleen

nog maar een lengte aan de balk te worden toegewezen en moet de balk in het tekenvenster worden geplaatst. Ten slotte kunnen de steunpunten worden toegevoegd met behulp van sneltoetsen of door handmatig een steunpunt te creëren.

De belastingen kunnen op de balk worden toegepast door het werkstation "Belastingen" te selecteren. Eerst dient een nieuw belastingsgeval te worden aangemaakt. Vervolgens kunnen verticale puntlasten, momenten op één punt, verdeelde belastingen en verdeelde momenten worden aangebracht. SCIA Engineer genereert automatisch de belastingscombinaties voor de uiterste grenstoestand in blijvende ontwerpsituatie, voor de gebruiksgrenstoestand met karakteristieke combinaties en voor de gebruiksgrenstoestand in quasi-permanente situatie. De gebruiker heeft de mogelijkheid om extra belastingscombinaties naar wens toe te voegen. Tenslotte wordt de balk berekend door op de knop in het midden van het wiel linksonder van het scherm te klikken.

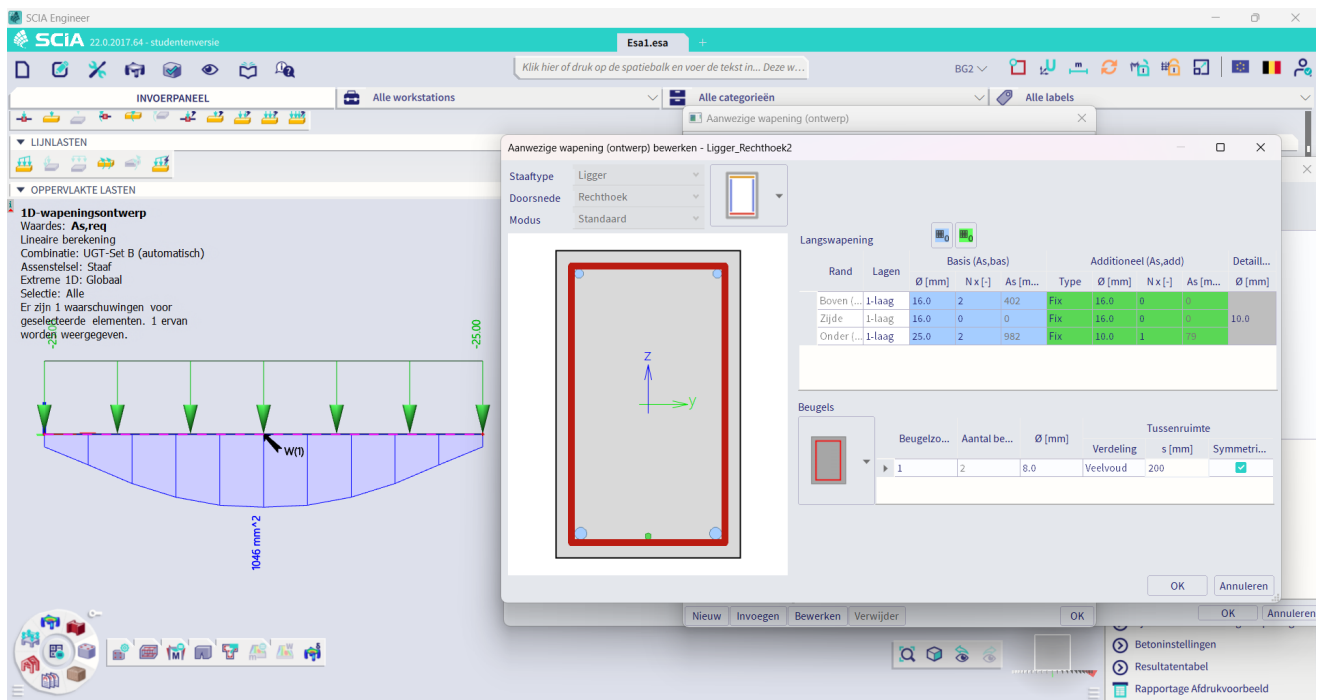
De resultaten kunnen worden opgevraagd door op het icoon met de hittekaart in het wiel te klikken. Hierdoor opent een horizontale balk met verschillende functies, zoals te zien is in Figuur 2.6.



Figuur 2.6: Resultaten in SCIA Engineer

Zoals geïllustreerd in Figuur 2.6, kunnen verschillende resultaten worden opgevraagd via de kolom aan de rechterkant van het scherm. Hierbij kunnen diverse krachten worden weergegeven en dit voor verschillende belastingsgevallen of belastingscombinaties. Naast het verkrijgen van interne krachten, spanningen en vervormingen, is het mogelijk om verschillende functies voor elk type materiaal op te vragen door het bijbehorende pictogram op het wiel te selecteren.

Om wapening in betonnen balken te plaatsen, moet de gebruiker eerst de benodigde hoeveelheid wapening laten berekenen. Dit kan door het selecteren van beton in het wiel, gevolgd door de keuze van de functie met de afbeelding van de krachten. Vervolgens kan de gebruiker zelf de wapening plaatsen. Figuur 2.7 illustreert het scherm waarop de wapening gemodelleerd kan worden.



Figuur 2.7: Modellerings van wapening in beton in SCIA Engineer

Zoals weergegeven in Figuur 2.7, kan de gebruiker eenvoudig het gewenste aantal staven en de diameter kiezen, waarbij het programma de wapeningssectie berekent en een duidelijke doorsnede-tekening genereert. Ook kan de diameter en tussenafstand van de beugels hier worden gekozen. Nadat de wapening is gemodelleerd, kan de gebruiker de gewenste controles uitvoeren. Voor staal en hout kunnen deze controles direct worden uitgevoerd.

Gebruikersinterface

De gebruikersinterface van SCIA Engineer heeft een moderne uitstraling en biedt een volledige 3D-weergave. Gebruikers hebben de flexibiliteit om de interface aan te passen naar hun eigen voorkeur door bepaalde elementen te verplaatsen en op verschillende posities vast te zetten. Daarnaast kunnen functies bij het wiel eenvoudig worden toegevoegd of verwijderd door ze te slepen. Dit maakt het programma gebruiksvriendelijk.

Voor beginners kan de gebruikersinterface echter minder ideaal zijn. Veel functies zijn op meerdere plaatsen beschikbaar, wat kan leiden tot verwarring over waar bepaalde functies te vinden zijn. Bovendien openen nieuwe vensters vaak met een overvloed aan parameters en knoppen, wat overweldigend kan zijn voor nieuwe gebruikers. Het analyseren van resultaten is ook omslachtig omdat er steeds het juiste belastingsgeval of -combinatie moet worden gekozen.

Educatieve waarde

De resultaten van een controle in SCIA Engineer worden altijd weergegeven op een schaal van één, waarbij een waarde onder één voldoet aan de gestelde eisen en een waarde boven één niet voldoet. Voor meer gedetailleerde informatie kan een rapport worden gegenereerd, waarin de gebruikte waarden, factoren en normen worden toegelicht. Helaas worden hierin de uitgeschreven formules niet vermeld. Ondanks dit tekort kan het rapport studenten helpen om een beter begrip te krijgen van de berekeningen. Een voorbeeld van een dergelijk rapport is te vinden in bijlage A.

Prijs SCIA Engineer

Het exacte kostenplaatje is complex en varieert per gebruiker omdat SCIA met diverse modules werkt. Gebruikers hebben de mogelijkheid om een gepersonaliseerde softwareconfiguratie samen te stellen. De basismodule, die altijd vereist is, kost €99 per maand. Voor de module die stabiliteitsberekeningen zoals knik en kip mogelijk maakt, bedraagt de maandelijkse kost €32. Specifieke berekeningen voor verschillende materialen vereisen afzonderlijke aankopen. De staalmodule en de module voor gewapend beton hebben elk een maandelijkse prijs van €99, terwijl de houtmodule €69 per maand kost. Bij de keuze van meerdere modules wordt er een korting toegepast waardoor het moeilijk is om een exacte prijs vast te stellen. Deze prijzen zijn gebaseerd op jaarlijkse betaling, indien er maandelijks wordt betaald zijn de kosten hoger [23]. Voor studenten is er wel een gratis studentenlicentie beschikbaar.

2.2.3 ConCrete (Plus)

ConCrete (Plus) is een softwareproduct van BuildSoft, specifiek ontworpen voor de berekening van zowel isostatische als hyperstatische betonnen balken. Het softwarepakket is opgedeeld in drie modules. ConCrete voert berekeningen uit voor de doorlopende balk, Concrete Plus genereert gedetailleerde wapeningsplannen voor de berekende balk, en ConCrete List creëert overzichten zoals de buigstaat en borderels [14].

Mogelijke berekeningen

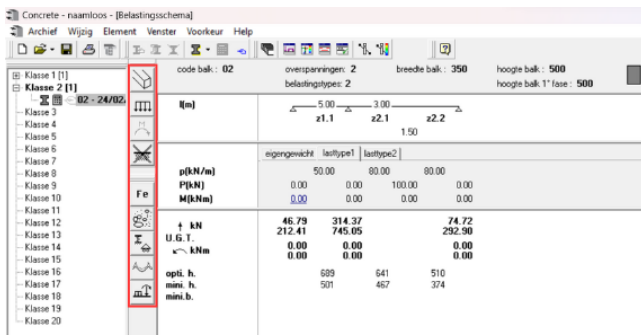
ConCrete is specifiek ontwikkeld voor het uitvoeren van berekeningen met betrekking tot betonnen liggers. Het programma maakt gebruik van de voorschriften van Eurocode 2 om controles uit te voeren en de benodigde wapening voor de betonnen ligger te bepalen. Voordat het programma overgaat tot het berekenen van de wapening van de betonnen ligger, worden eerst de interne krachten volgens Eurocode 0 berekend. ConCrete genereert vervolgens een voorstel voor de benodigde wapening en presenteert dit samen met een bijbehorend overzicht (borderel).

Modellering in ConCrete (Plus)

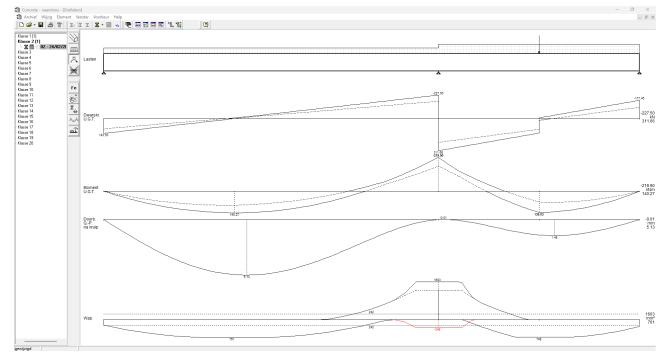
In ConCrete wordt een balk volledig gemodelleerd zonder handmatige tekeningen. Het programma leidt de gebruiker door een reeks vragen waarbij informatie moet worden ingevoerd, zoals het aantal overspanningen, de lengtes ervan, en de belastingwaarden. Elke overspanning kan worden opgedeeld in zones, waarbij puntlasten of momenten kunnen worden geplaatst op de overgangen tussen deze zones. Het overzicht na al deze vragen in te vullen wordt weergegeven in Figuur 2.8.

Zoals geïllustreerd in Figuur 2.8, verschaft het programma een helder overzicht van de lengte, breedte, hoogte van de balk en alle belastingen. Verder is het mogelijk om staal- en betonsterktes, evenals randvoorwaarden en belastingscombinaties, aan te passen via de knoppen in de kolom, aangeduid in de rode rechthoek in Figuur 2.8.

De krachtendiagrammen, doorbuiging en wapening zijn eenvoudig te raadplegen door simpelweg de knop “Grafieken” bovenaan in de balk aan te klikken. De verkregen resultaten worden geïllustreerd in Figuur 2.9.



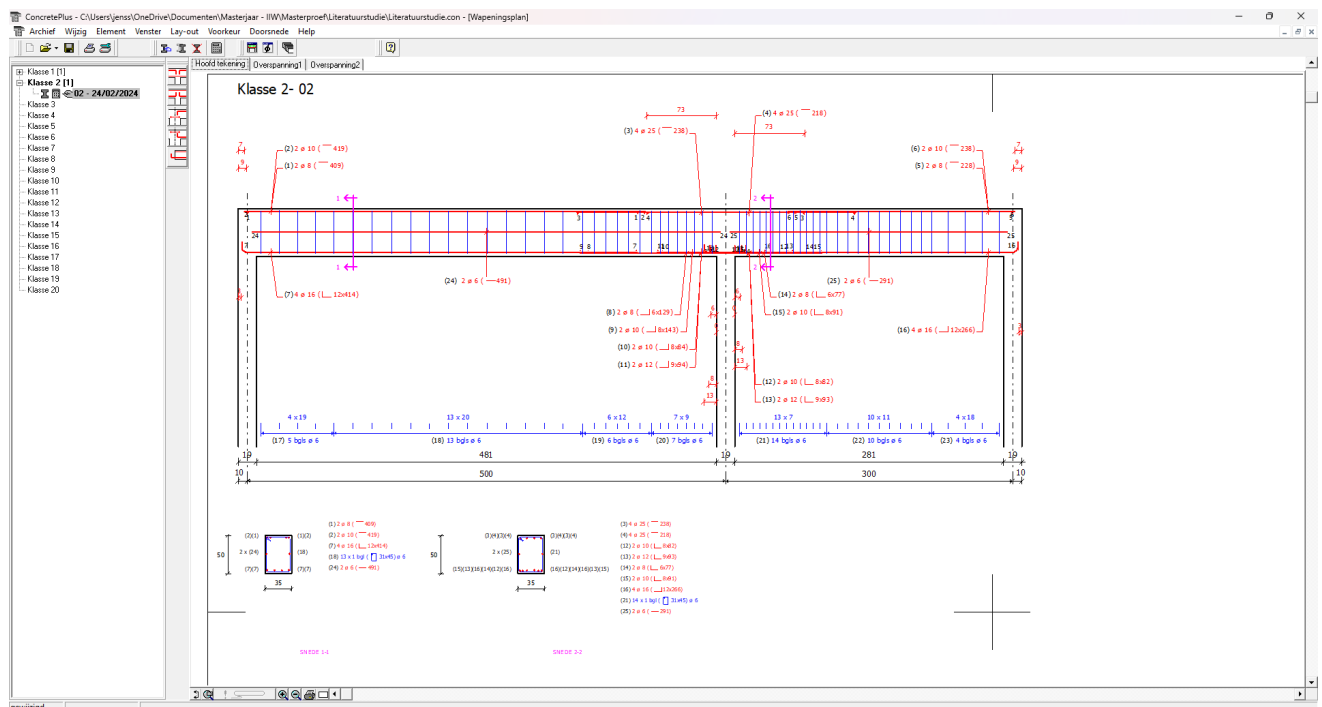
Figuur 2.8: Modellering ConCrete



Figuur 2.9: Resultaten in ConCrete

De resultaten in ConCrete worden duidelijk onder elkaar geplaatst. Dit zorgt voor een duidelijk overzicht van de resultaten, zoals weergegeven in 2.9.

Een wapeningsplan wordt gegenereerd door het bestand in ConCrete Plus te openen. Een voorbeeld van zo een wapeningsplan wordt getoond in Figuur 2.10.



Figuur 2.10: Wapeningsplan in ConCrete Plus

Het wapeningsplan, zoals weergegeven in Figuur 2.10, visualiseert de wapening in zijaanzicht, waarbij de langswapening in het rood en de dwarskrachtwapening in het blauw worden aangegeven. Het programma maakt het mogelijk dwarsdoorsneden te genereren om de wapening vanuit dit perspectief te inspecteren. Daarnaast biedt het gebruikers de flexibiliteit om de wapening handmatig aan te passen.

Ten slotte heeft de gebruiker de mogelijkheid om dit wapeningsplan te importeren in ConCrete List, waarmee vervolgens een buigstaat en een borderel kunnen worden gegenereerd. Het is echter niet gelukt om deze functionaliteit uit te voeren tijdens de literatuurstudie.

Gebruikersinterface

De gebruikersinterface is helder en gebruiksvriendelijk, met duidelijke presentatie van ingevoerde parameters die gemakkelijk aanpasbaar zijn. Vergelijkbaar met Diamonds, zijn geavanceerde instellingen verborgen achter knoppen om het overzicht te behouden, waardoor het zowel voor beginners als gevorderden toegankelijk blijft en een vriendelijke gebruikerservaring biedt.

Educatieve waarde

De heldere weergave van resultaten, waarbij alle relevante informatie in één scherm wordt gepresenteerd, vergemakkelijkt het begrijpen van correlaties tussen deze resultaten. Zo is het bijvoorbeeld duidelijk zichtbaar dat waar het moment groot is, ook de vereiste hoeveelheid wapening groot is. Dit samen met de duidelijke visualisatie van de wapening in de balk maakt het voor studenten eenvoudiger om te begrijpen waarom en hoe wapening op bepaalde plaatsen is toegepast. Echter is het niet mogelijk om gedetailleerde berekeningen op te vragen.

Prijs ConCrete (Plus)

De aanschaf van de drie softwarepakketten is alleen beschikbaar als een bundel. Dit resulteert in een kost van €180 per maand bij jaarlijkse facturatie en €288 per maand bij maandelijks facturatie. Studenten kunnen een gratis studentenlicentie aanvragen.

2.2.4 Tekla Tedds

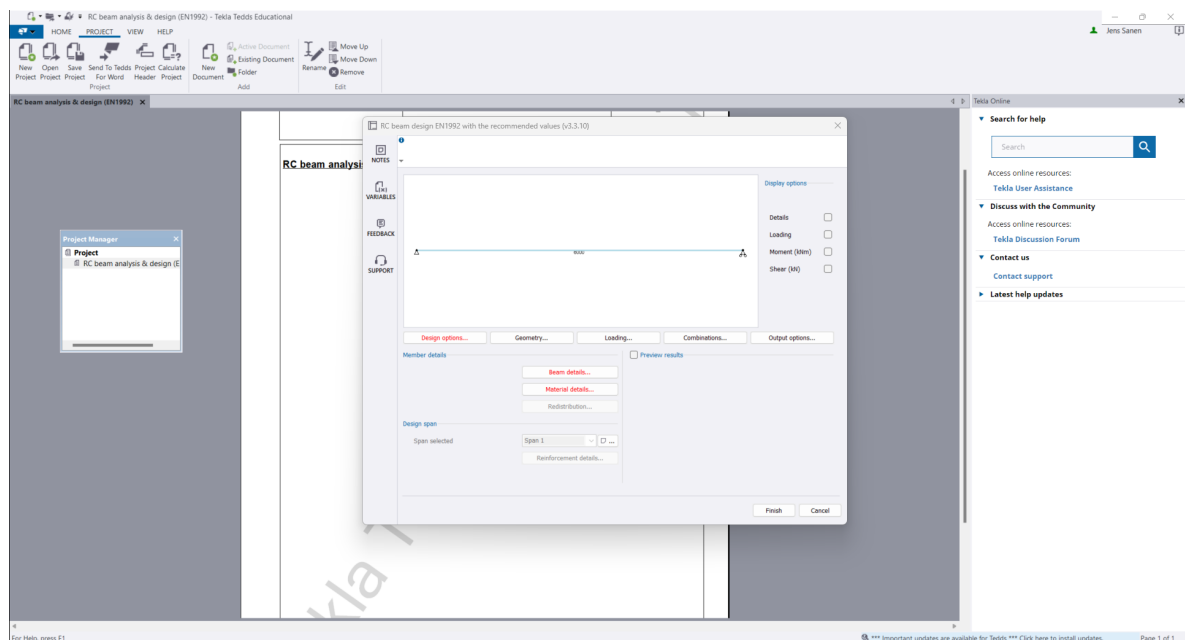
Tekla Tedds is een softwareprogramma, ontworpen door Tekla, dat specifiek is ontwikkeld voor de analyse van bouwkundige constructies. Het biedt snelle en nauwkeurige berekeningen voor diverse materialen, in overeenstemming met de geldende Eurocodes [15].

Mogelijke berekeningen

Tekla Tedds voert de analyse uit conform aan op basis van de vereiste belastingscombinaties en de juiste partiële factoren, in volledige overeenstemming met Eurocode 0 [8]. Bovendien is het programma in staat om berekeningen uit te voeren voor doorlopende betonnen, stalen en houten liggers, conform de relevante Eurocodes. In het geval van betonnen liggers genereert het programma geen wapeningsvoorstellen en voor houten en stalen liggers is er geen automatische optimalisatiefunctie voor profielen beschikbaar. Niettemin wordt tijdens het modelleren altijd de benodigde wapening of capaciteit aangegeven, zodat gebruikers op basis hiervan naar de optimale oplossing kunnen zoeken. Het is echter belangrijk op te merken dat hoewel Tekla Tedds Eurocodes hanteert, het programma geen berekeningen kan uitvoeren op basis van de nationale bijlagen van België, omdat deze niet zijn opgenomen in de software [24].

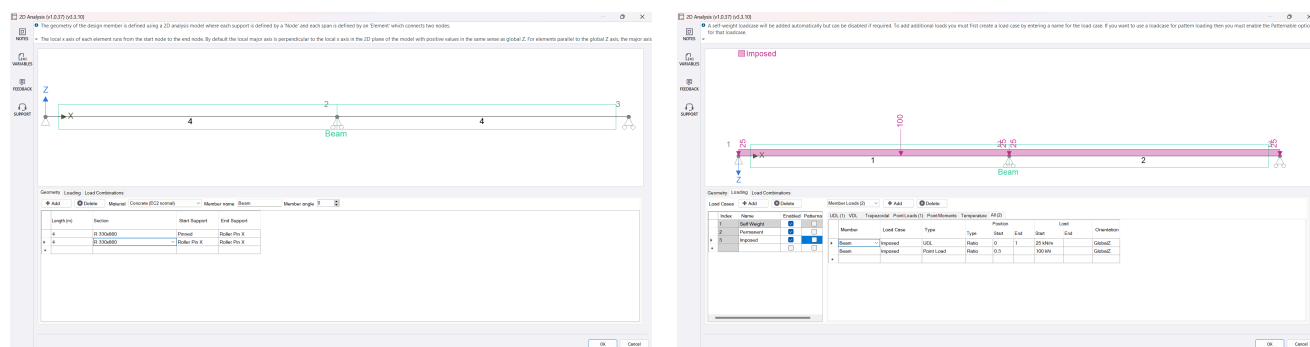
Modellering in Tekla Tedds

Tekla Tedds maakt gebruik van een uitgebreide bibliotheek met vooraf geprogrammeerde formulieren. Wanneer de gebruiker een balk wil berekenen, moet hij het specifieke formulier voor het gewenste materiaaltype selecteren. In dit formulier staan verschillende knoppen die andere schermen openen, zoals bijvoorbeeld knoppen voor geometrie, belastingen en belastingscombinaties. In Figuur 2.11 wordt weergegeven hoe het formulier eruitziet voor het berekenen van een doorlopende betonnen balk volgens Eurocode 2 [1].



Figuur 2.11: Berekening beton in Tekla Tedds

Zoals te zien is op Figuur 2.11, moeten verschillende stappen worden doorlopen om een berekening uit te voeren. Eerst, bij “Design options...”, moet de gebruiker keuzes maken met betrekking tot de plaatsing van wapening, de te gebruiken nationale bijlage en of er controle moet worden uitgevoerd op scheurwijdte. Vervolgens kunnen de geometrie, belastingen en belastingscombinaties worden geconfigureerd, zoals geïllustreerd in Figuur 2.12.



(a) Geometrie in Tekla Tedds

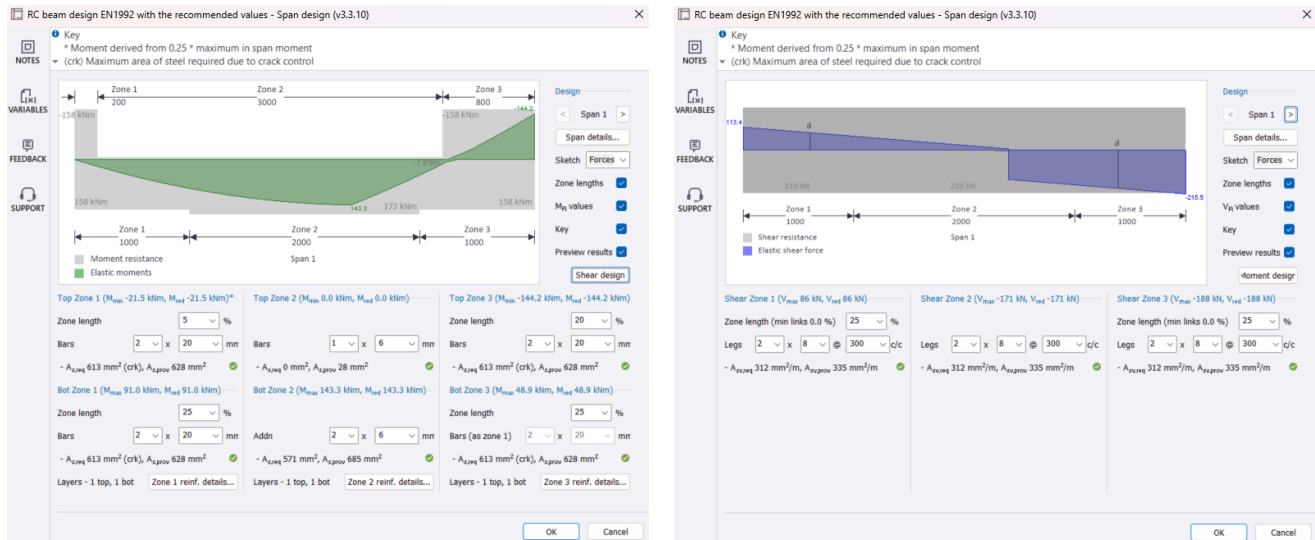
(b) Belastingen in Tekla Tedds

Figuur 2.12: Modelleren in Tekla Tedds

Op Figuur 2.12a moet de gebruiker het aantal overspanningen, lengte van de overspanningen, de dwarsdoorsnede en de steunpunten modelleren. In Figuur 2.12b moet de gebruiker de belastingen toepassen, zowel verdeelde belastingen als puntlasten zijn mogelijk. Tot slot moet de gebruiker belastingscombinaties genereren, waarbij het programma automatisch combinaties genereert voor de uiterste grenstoestand bij blijvende ontwerpsituatie, voor de gebruiksgrenstoestand bij karakteristieke combinaties en voor de quasi-permanente situatie in de gebruiksgrenstoestand.

Na deze stappen kan de gebruiker bij “Beam details...” de betondekking, maximale scheurwijdte en benodigde brandweerstand specificeren. Onder “Material details...” heeft de gebruiker de mogelijkheid de betonkwaliteit te kiezen, terwijl bij “Redistribution...” instellingen kunnen worden aangepast met betrekking tot de herverdeling van krachten. Na deze configuraties kan de

gebruiker overgaan tot het modelleren van de wapening door op “Reinforcement details...” te klikken. De schermen die dan verschijnt, wordt weergegeven in Figuur 2.13.



(a) Langswapening in Tekla Tedds

(b) Dwarskrachtwapening in Tekla Tedds

Figuur 2.13: Wapening in Tekla Tedds

Zoals te zien is in Figuur 2.13a, moet de gebruiker zelf de hoeveelheid en diameter van de wapening bepalen. Het programma geeft echter aan hoeveel wapening er nodig is en controleert tegelijkertijd of de geplaatste wapening aan deze eisen voldoet. Dit geldt ook voor de dwarskrachtwapening, zoals weergegeven in Figuur 2.13b.

Tot slot heeft de gebruiker de mogelijkheid om de resultaten te documenteren in een rapport, waarin alle berekeningen stap voor stap duidelijk en overzichtelijk worden gepresenteerd. Het verslag van de betonnen balk is opgenomen als bijlage B in de bijgevoegde documenten.

De formulieren voor stalen en houten balken vertonen overeenkomsten met deze van beton, zij het met verschillende toepassingen die specifiek zijn voor het betreffende materiaal. Het nadeel aan deze formulieren is dat het niet mogelijk is om snel van materiaal te veranderen voor dezelfde balk.

Gebruikersinterface

De gebruikersinterface van Tekla Tedds is intuïtief. Zodra de gebruiker het gewenste formulier heeft geselecteerd, is het duidelijk welke velden moeten worden ingevuld en wat elke knop doet. Dit zorgt voor een soepele en eenvoudige uitvoering van berekeningen. Bovendien zijn er visuele ondersteuningen beschikbaar om de balk, wapening en krachtdiagrammen helder weer te geven. Over het algemeen heeft de gebruikersinterface een modern en aantrekkelijk ontwerp.

Educatieve meerwaarde

Tekla Tedds biedt aanzienlijke educatieve meerwaarde. Studenten kunnen op een snelle, eenvoudige en nauwkeurige manier balken berekenen en vervolgens alle resultaten analyseren via het gegenereerde formulier met de berekeningen. Bovendien dragen de visuele componenten bij aan een beter begrip van zaken zoals de positionering van de wapening.

Prijs Tekla Tedds

Tekla vermeldt geen prijzen op hun website en voor de aanschaf van de software wordt gevraagd om contact op te nemen met het bedrijf. Er is een beschikbare proefperiode van 30 dagen voor geïnteresseerden. Bovendien hebben studenten de mogelijkheid om een gratis studentenlicentie aan te vragen [15].

2.2.5 AxisVM

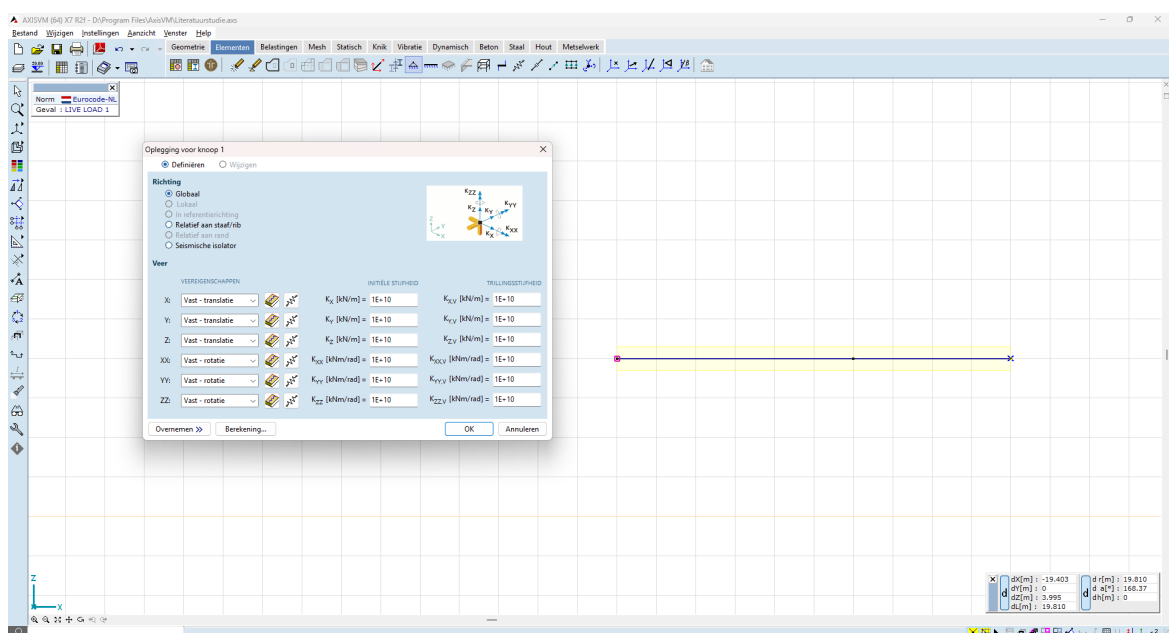
AxisVM is een softwareprogramma voor structurele analyses, ontwikkeld door InterCAD, een bedrijf gevestigd in Hongarije. Het primaire doel van InterCAD is om bouwkundige ingenieurs te ondersteunen bij het creëren van zowel creatieve als efficiënte ontwerpen [25].

Mogelijke berekeningen

AxisVM analyseert de balk volgens Eurocode 0. Het programma is in staat om betonnen, stalen en houten balken te analyseren volgens de respectieve Eurocodes. Bovendien kan AxisVM, in het geval van staal en hout, optimalisaties uitvoeren voor de profielen. Voor beton biedt het programma de mogelijkheid om de betonbalk te controleren, waarbij de gebruiker zelf eerst de wapening moet modelleren.

Modelleren in AxisVM

Om een balk te tekenen, dient de werkbalk bovenaan te worden ingesteld op “Elementen”. Binnen dit werkpakket moet de functie aangeduid met een potlood worden geselecteerd. Bij deze functie moet de gebruiker een materiaal kiezen, waarvoor een bibliotheek beschikbaar is. Daarnaast moet de gebruiker een profiel selecteren, dat kan zowel door zelf een profiel aan te maken als door er één te selecteren uit de bibliotheek. Ten slotte moet de gebruiker de lengte ingeven. De volgende stap is het modelleren van de steunpunten. Hiervoor moet de gebruiker op de knop drukken met de afbeelding van een steunpunt, waardoor het scherm in Figuur 2.14 wordt geopend.

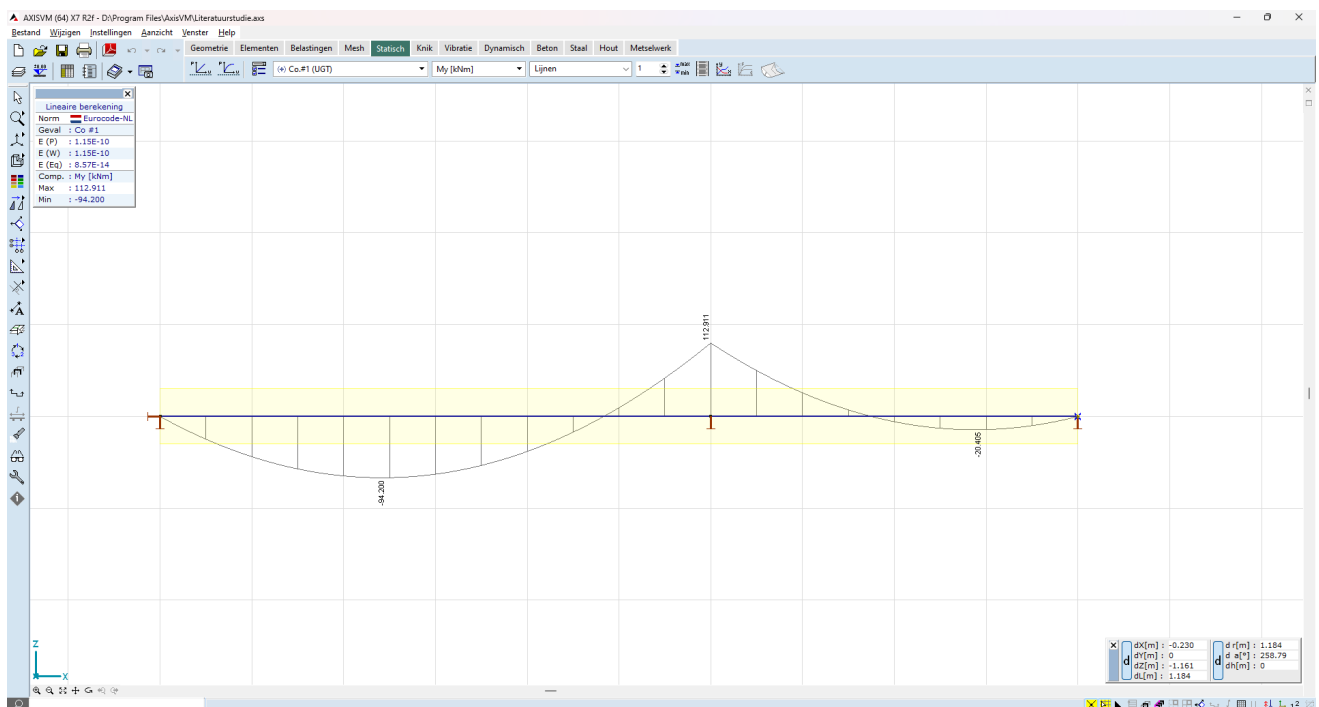


Figuur 2.14: Modelleren van steunpunten in AxisVM

Zoals te zien is in Figuur 2.14, moet de gebruiker zelf de eigenschappen van het steunpunt definiëren. Er zijn geen voorgeprogrammeerde sneltoetsen beschikbaar voor een roloplegging, scharnier of inklemming.

Om belastingen toe te passen moet de werkbalk vanboven op “Belastingen” worden gezet. Hierna moet de gebruiker eerst een nieuw belastingsgeval creëren. Het programma genereert automatisch één belastingsgeval, maar dit is alleen bedoeld voor het eigengewicht van de balk en er kunnen geen andere belastingen aan worden toegevoegd. Nadat dit belastingsgeval is aangemaakt, kan de gebruiker puntlasten, momenten, verdeelde belastingen en verdeelde momenten op de balk toepassen. Ten slotte moet de gebruiker belastingscombinaties laten genereren of handmatig samenstellen.

Voor het uitvoeren van de berekening van de balk moet eerst het net aangemaakt worden. Dit wordt bereikt door de werkbalk in te stellen op “Mesh” en vervolgens de balk te selecteren, waarbij de gewenste elementgrootte of het aantal elementen wordt ingevoerd. Om de balk te berekenen, moet de gebruiker de werkbalk instellen op “Statisch” en de functie kiezen met het pictogram van een lineaire grafiek. Het resulterende beeld wordt weergegeven in Figuur 2.15.



Figuur 2.15: Resultaten in AxisVM

In Figuur 2.15 wordt de momentenlijn voor de eerste belastingscombinatie getoond. In de balk bovenaan heeft de gebruiker de mogelijkheid om de belastingscombinatie, het type kracht of vervorming, en de weergavemodus aan te passen.

Voor specifieke berekeningen met betrekking tot een bepaald materiaal moet de werkbalk bovenaan worden ingesteld op dat specifieke materiaaltipe. Tijdens de literatuurstudie werd opgemerkt dat het niet mogelijk was om houten en stalen balken te optimaliseren. Dit kan mogelijk te wijten zijn aan het feit dat de studentenversie van de software werd gebruikt.

Gebruikersinterface

Het programma biedt volledige 3D-functionaliteit en visuele ondersteuning. De functies zijn echter goed georganiseerd, elk in hun eigen werkbalk. De resultaten worden duidelijk gepresenteerd, met name bij betonberekeningen, waarbij alle resultaten netjes onder elkaar staan net zoals bij ConCrete. Aan de andere kant opent het programma vaak nieuwe schermen met diverse functies en parameters, wat nieuwe gebruikers kan overweldigen.

Educatieve waarde

AxisVM toont overzichtelijke resultaten maar voor de rest is er geen educatieve waarde. Er is geen mogelijkheid om resultaten op te vragen en er is geen visuele presentatie van de wapening in betonbalken.

Prijs AxisVM

Op de site van AxisVM worden geen prijzen vermeld. De prijs wordt bepaald op basis van de geselecteerde modules, voor een exacte prijs moet contact worden opgenomen met het bedrijf. Voor studenten is er wel een gratis studentenlicentie beschikbaar.

2.2.6 Bachelorproef Gielkens, Jans en Sanen

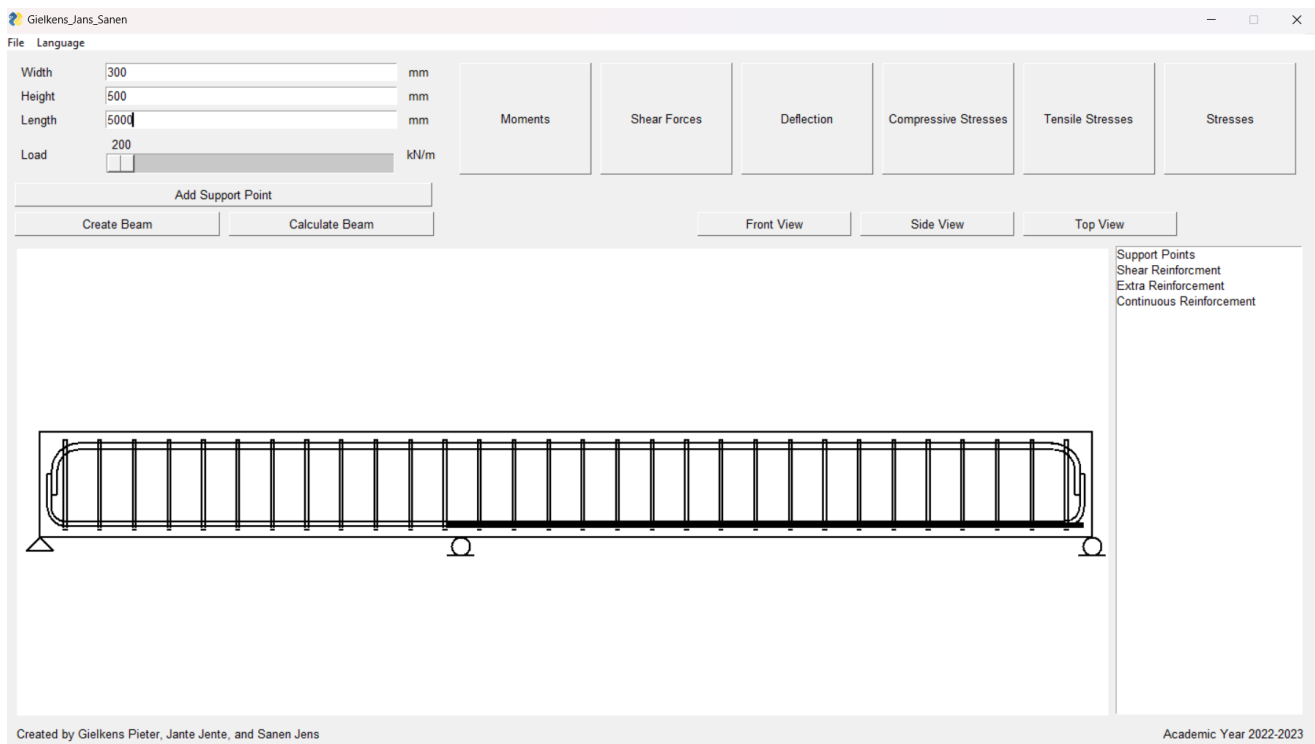
In het academiejaar 2022-2023 ontwikkelden Pieter Gielkens, Jente Jans en Jens Sanen, drie studenten van UHasselt, een Python-programma voor de berekening en visualisatie van betonnen balken. Het hoofddoel van dit programma is om studenten een dieper inzicht te bieden in de werking van iso- en hyperstatische betonnen balken, met speciale aandacht voor de positionering van wapening binnen deze balken [7].

Mogelijke berekeningen

Het programma is in staat om dwarskrachten, momenten, doorbuigingen en spanningen te berekenen. Echter, kan er enkel een verdeelde belasting over de volledige lengte van de balk aangebracht worden. Bovendien werd er geen rekening gehouden met Eurocode 0. Dit betekent dat er geen belastingscombinaties werden gemaakt en dat er geen rekening werd gehouden met verschillende grenstoestanden. Voor betonberekeningen kan het programma automatisch de benodigde wapening berekenen en op basis daarvan een wapeningsconfiguratie genereren. Het is echter belangrijk op te merken dat vereenvoudigde formules worden gebruikt, waardoor het programma niet voldoet aan Eurocode 2. Daarnaast kan het programma ook geen controles uitvoeren voor doorbuiging, scheurwijdte en spanningen.

Modellering

Bij het openen van het programma kan de gebruiker onmiddellijk de balk dimensioneren. Hij moet de breedte, hoogte en totale lengte van de balk invoeren, waarna hij de balk kan creëren. Nadat de balk is gecreëerd, kan de gebruiker extra steunpunten toevoegen en de waarde van de verdeelde belasting kiezen met behulp van de schuifbalk. Vervolgens kan de wapening worden berekend. Figuur 2.16 toont het beeld dat verkregen wordt na de berekening van de wapening.



Figuur 2.16: Balk modelleren in programma bachelorproef

Zoals te zien is op Figuur 2.16, tekent het programma de balk met de berekende wapening. De berekende wapening kan getekend worden voor het voor-, zij- en bovenaanzicht. Aan de rechterzijde kan de gebruiker extra informatie over de wapening opvragen, zoals de diameter, lengte en positie. Bovenaan kan de gebruiker de interne krachten, doorbuiging en spanningen opvragen.

Gebruikersinterface

Doordat er weinig functionaliteiten zijn, worden alle knoppen op één scherm weergegeven, wat het gebruik van het programma eenvoudig maakt. Echter kan het uiterlijk van de interface worden verbeterd.

Educatieve waarde

De weergave van de wapening in combinatie met het momentendiagram en de optredende spanningen kunnen een nuttig hulpmiddel zijn voor studenten om het concept beter te begrijpen. Echter, doordat het programma geen formules en controles van Eurocode 2 hanteert, is dit geen correcte weerspiegeling van de realiteit. Bovendien ontbreekt de mogelijkheid om berekeningen op te vragen.

Prijs

Aangezien dit programma specifiek is ontwikkeld voor een bachelorproef en niet beschikbaar is voor algemeen gebruik, is er geen prijs.

2.2.7 Masterproef Bloemen en Jehaes

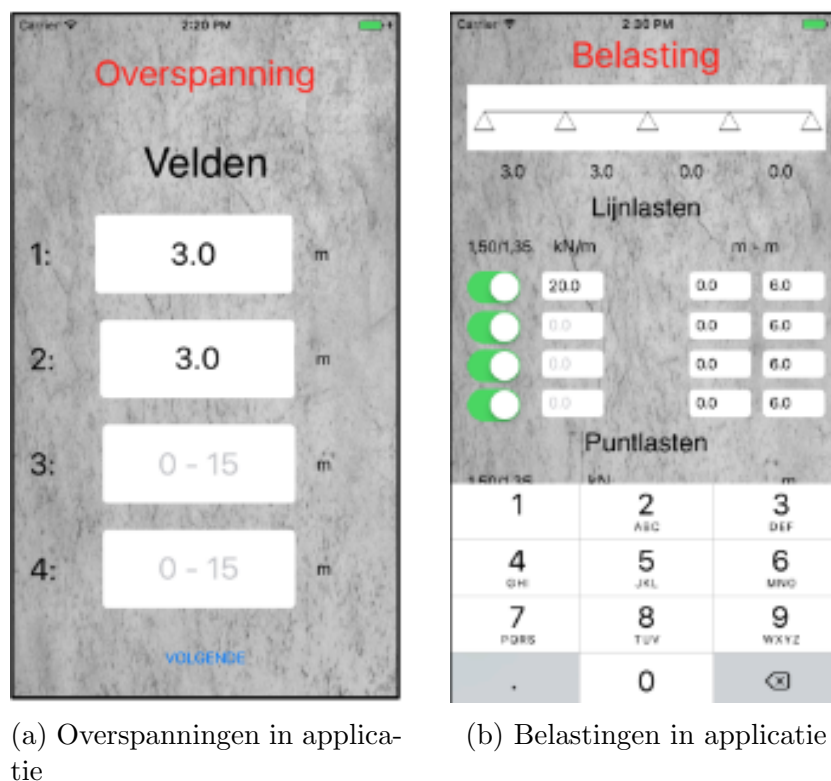
In het academiejaar 2016-2017 werd door Ruben Bloemen en Jelle Jehaes, twee studenten van UHasselt, een mobiele iOS-applicatie ontwikkeld. Het doel van deze applicatie is om stalen en betonnen balken te berekenen [19].

Mogelijke Berekeningen

Het programma is in staat liggers te analyseren met maximaal vier overspanningen. De interne krachten worden berekend volgens de voorschriften van Eurocode 0. Voor betonnen constructies kan de applicatie de optimale wapening, hoogte en controles op spanningen, scheurwijdte en doorbuiging uitvoeren, conform Eurocode 2. Bij stalen liggers kan de software het optimale profiel bepalen en de doorbuiging controleren volgens Eurocode 3 [19].

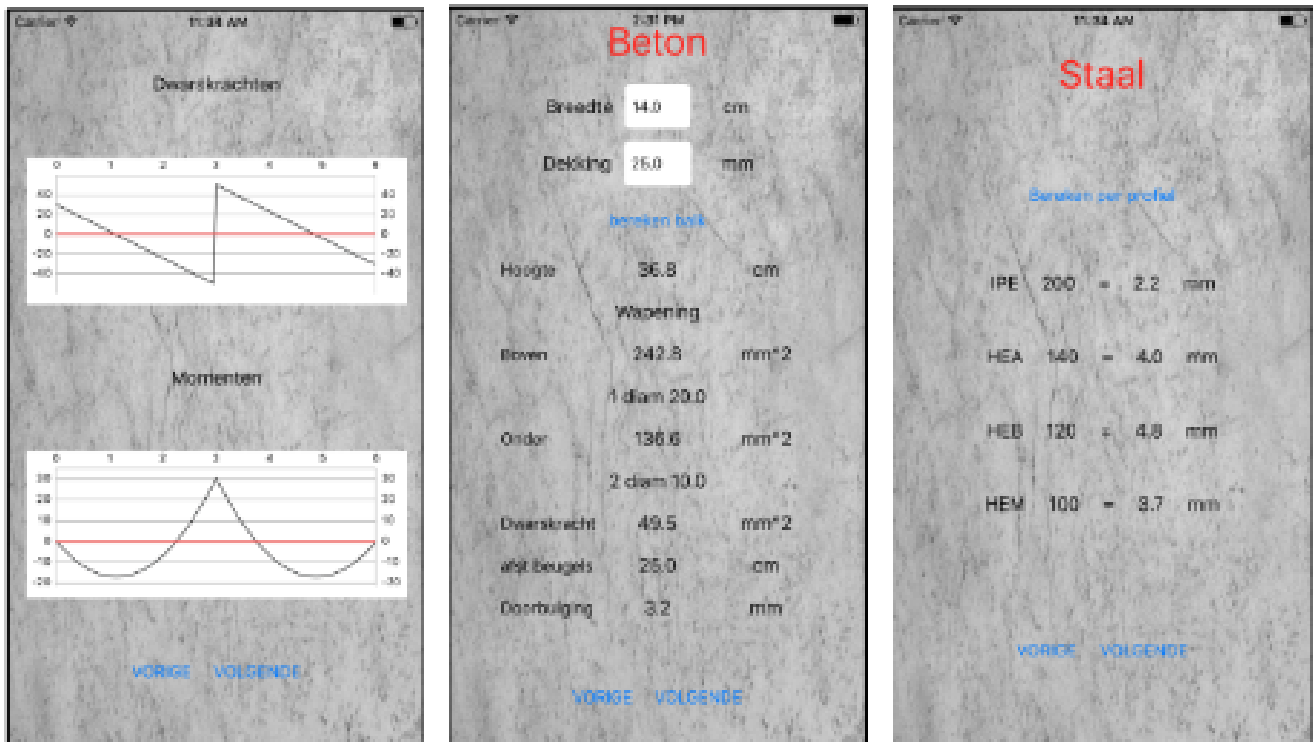
Modellering

Aangezien deze app het resultaat is van een masterproef, was het niet mogelijk om deze zelf te onderzoeken tijdens de literatuurstudie. De informatie en figuren die hieronder zijn opgenomen, zijn allemaal afkomstig uit de scriptie van de masterproef [19]. In Figuur 2.17 wordt geïllustreerd hoe men de ligger en bijbehorende belastingen moet configureren binnen de applicatie.



Figuur 2.17: Modellering in applicatie masterproef

Zoals te zien is in Figuur 2.17a is de maximale overspanning gelijk aan vijftien meter. Indien de gebruiker slechts twee overspanningen wilt, vult hij enkel de eerste twee vakken in. In Figuur 2.17b wordt getoond hoe een lijnlast op de balk geplaatst kan worden. De resultaten voor de krachtendiagrammen, de optimale betonnen balk en optimale stalen balk worden getoond in Figuur 2.18.



(a) Krachtsdiagrammen in applicatie (b) Resultaat betonnen balk in applicatie (c) Resultaat stalen balk in applicatie

Figuur 2.18: Resultaten in applicatie masterproef

Zoals geïllustreerd in Figuur 2.18a, worden de krachtsdiagrammen op een eenvoudige manier weergegeven. In Figuur 2.18b wordt gedemonstreerd dat de gebruiker een breedte en betondekking moet specificeren, waarna de applicatie automatisch de benodigde wapening berekent. Figuur 2.18c toont voor staal het optimale profiel en bijbehorende doorbuiging voor vier typen profielen.

Gebruikersinterface

De app is intuïtief, aangezien er slechts één pad is dat de gebruiker kan volgen. Daarentegen oogt de interface eenvoudig en mist het professionele uitsraling.

Educatieve waarde

Deze applicatie heeft weinig educatieve waarde. Enerzijds kunnen er geen berekeningen worden opgevraagd. Anderzijds ontbreekt een visuele weergave van de wapening, waardoor het voor studenten minder duidelijk is waar de wapening precies moet worden geplaatst.

2.2.8 Overzicht

In dit gedeelte wordt een overkoepelend overzicht gegeven van de onderzochte softwareprogramma's. Dit overzicht wordt weergegeven in Tabel 2.1. In de kolom voor de kenmerken van de gebruikersinterface worden "+" en "-" gebruikt om respectievelijk positieve en negatieve kenmerken van de gebruikersinterface aan te geven.

Tabel 2.1: Overzicht bestaande softwareprogramma's

Software	Berekeningen	Snelheid modellering	Gebruikersinterface	Educatieve waarde	Prijs
Diamonds	Analyse: - Belastingscombinaties - Interne krachten - Spanningen - Doorbuigingen Beton: - Nodige wapening - Scheurwijdte - Doorbuiging Staal en hout: - Controle sterkte - Controle stabiliteit - Sectie optimaliseren	Stappen: 1) Geometrie specificeren 2) Sectie + materiaal kiezen 3) Eventueel steunpunten wijzigen 4) Belastingen plaatsen 5) Combinaties aanmaken 6) Net genereren 7) Balk berekenen Opmerkingen: - Shortcut voor stap 1-3	+ Efficiënt + Goede structuur + Duidelijke pictogrammen + Geavanceerde instellingen verborgen + Vriendelijk voor beginners	Voor staal en hout controles mogelijk om rekenwaarden op te vragen	129 €/maand of studenten-licentie
SCIA Engineer	Analyse: - Belastingscombinaties - Interne krachten - Spanningen - Doorbuigingen Beton: - Nodige wapening - Scheurwijdte - Doorbuiging	Stappen: 1) Project aanmaken 2) Sectie + materiaal kiezen 3) Geometrie specificeren 4) Steunpunten toevoegen 5) Belastingsgeval creëren 6) Belastingen plaatsen 7) Balk uitrekenen	+ Modern + Goede structuur + Duidelijke pictogrammen + Aanpasbaar - Resultaten omslachtig - Niet vriendelijk voor beginners	Mogelijk rapport te laten genereren met gebruikte factoren, normen en rekenwaarden	Op basis van gekozen modules of studenten-licentie

Software	Berekeningen	Snelheid modellering	Gebruikersinterface	Educatieve waarde	Prijs
SCIA Engineer	Staal en hout: - Controle sterkte - Controle stabiliteit - Sectie optimaliseren	Opmerkingen: - Shortcuts voor steunpunten en belastingen			
ConCrete (Plus)	Analyse: - Belastingscombinaties - Interne krachten Beton: - Scheurwijdte - Doorbuiging - Nodige wapening - Wapeningsvoorstel - Wapeningsschema	Balk wordt volledig uitgerekend op basis van antwoorden op voorgeprogrammeerde vragen Opmerkingen: - Gegevens achteraf mogelijk aanpasbaar	+ Gegevens duidelijk samengevat + Intuïtief + Duidelijke pictogrammen	Resultaten helder weergegeven + gedetailleerd plan wapening	€180/maand of Studenten-licentie
Tekla Tedds	Analyse: - Belastingscombinaties - Interne krachten Beton: - Scheurwijdte - Nodige wapening Staal en hout: - Controle sterkte - Controle stabiliteit	Balk wordt volledig uitgerekend op basis van gekozen formulier Opmerkingen: - Niet mogelijk om snel van materiaal te veranderen voor dezelfde balk	+ Vanzelfsprekend + Duidelijk wat er ingevuld moet worden + Goede visuele ondersteuning + Modern uiterlijk	Duidelijk rapport met resultaten en gebruikte formules en rekenwaarden	prijs op aanvraag of studenten-licentie

Software	Berekeningen	Snelheid modellering	Gebruikersinterface	Educatieve waarde	Prijs
AxisVM	Interne krachten: - Belastingscombinaties - Interne krachten - Verplaatsingen - Spanningen Beton: - Scheurwijdte - Doorbuiging - Nodige wapening Staal en hout: - Controle sterkte - Controle stabiliteit - Sectie optimaliseren	Stappen: 1) Balk aanmaken 2) Sectie + materiaal kiezen 3) Steunpunten toevoegen 4) Lastengroepen aanmaken 5) Belastingen plaatsen 6) Combinaties aanmaken 7) Berekeningen uitvoeren	+ Goede visuele ondersteuning + Volledig 3D + Overzichtelijke resultaten	Duidelijke weergave van de resultaten	Prijs op aanvraag op basis van gekozen modules of studenten-licentie
Bachelorproef Gielkens, Jans, Sanen	- Interne krachten - Spanningen - Doorbuiging - Wapening	Stappen: 1) Dimensies specificeren 2) Belasting aanbrengen 3) Wapening uitrekenen	+ Eenvoudig + Vanzelfsprekend - Niet professioneel	Posities wapening in combinatie met interne krachten	Niet van toepassing
Masterproef Bloemen, Jehaes	Analyse: - Interne krachten - Doorbuiging Beton: - Wapening - Scheurwijdte Staal: - Optimaal profiel	Stappen: 1) Overspanningen specificeren 2) Belastingen plaatsen 3) Berekeningen uitvoeren	+ Vanzelfsprekend - Niet professioneel	Geen	Niet van toepassing

2.3 Conclusie

De literatuurstudie illustreert duidelijk de aanzienlijke diversiteit in de prestaties van de onderzochte softwareprogramma's. Elk van deze programma's vertoont specifieke sterke en zwakke punten, waaronder variaties in gebruikersinterfaces. Over het geheel genomen is er ruimte voor verbetering op het gebied van educatieve waarde, wat begrijpelijk is gezien het professionele karakter van deze softwaretoepassingen. Een niet onverwacht overeenkomst is dat alle programma's, met uitzondering van die uit de bachelorproef van 2022-2023 en die uit de masterproef van 2016-2017, gepaard gaan met financiële kosten.

Voor het berekenen van betonnen liggers is ConCrete (Plus) de meest geschikte keuze, aangezien het programma een snelle modellering van betonnen balken mogelijk maakt en direct de benodigde wapening voorstelt die aan alle controles voldoet. Daarnaast genereert het programma gelijktijdig een wapeningsplan en een overzichtelijk borderel, een unieke functie die andere software mist.

Voor het berekenen van stalen en houten liggers worden Diamonds, SCIA Engineer, en AxisVM aanbevolen vanwege hun mogelijkheid om de profielen te optimaliseren.

Tekla Tedds onderscheidt zich op het gebied van educatieve waarde, met name door de mogelijkheid om een helder en gedetailleerd verslag te genereren waarin de volledige berekening is uitgewerkt. Het rapport van SCIA doet dit ook voor een gedeelte maar geeft enkel verwijzingen naar de plaats van de formules in de norm in plaats van de formule zelf te geven.

Hoofdstuk 3

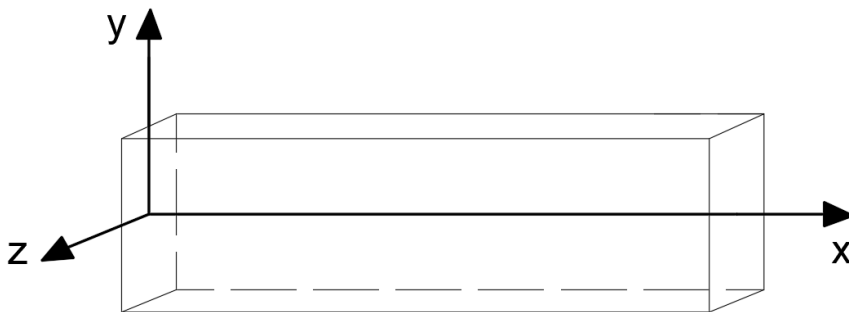
Achterliggende formules

Dit hoofdstuk legt alle onderliggende formules en controles van het programma uit die worden toegepast voor het analyseren en het ontwerpen van betonnen, stalen en houten balken. De formules en ontwerpeisen komen uit de Eurocodes, waarbij indien nodig de bijhorende Belgische nationale bijlage wordt gebruikt.

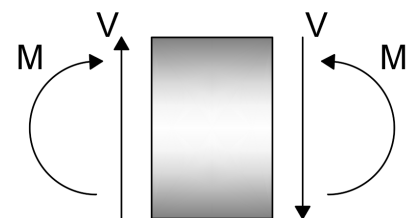
3.1 Analyse balk

Het analyseren van een balk omvat het berekenen van vervormingen, reactiekrachten en interne krachten. In deze masterproef wordt dit uitgevoerd met behulp van de eindige-elementenmethode (EEM). Deze sectie begint met een algemene inleiding over termen en theorieën volgens Eurocode 0. Vervolgens wordt de eindige-elementenmethode toegelicht en tot slot wordt een voorbeeld uitgewerkt om het geheel te verduidelijken.

Om consistentie te waarborgen in deze masterproef, is er gekozen voor een vast assenstelsel en tekenconventie. Het assenstelsel is als volgt gedefinieerd: de lengte van de balk loopt langs de x-as, de breedte langs de z-as, en de hoogte langs de y-as. Figuur 3.1 illustreert dit assenstelsel. In de tekenconventie geldt dat een dwarskracht positief is als deze in wijzerzin werkt, en een moment is positief als deze bovenaan druk en onderaan trek veroorzaakt. Figuur 3.2 verduidelijkt de tekenconventie.



Figuur 3.1: Aangenomen assenstelsel



Figuur 3.2: Tekonventie voor dwarskracht en moment

Zoals weergegeven in Figuur 3.1, zijn de assen gecentreerd in de dwarsdoorsnede van de balk. De keuze om de y-as verticaal te plaatsen is gebaseerd op de overweging dat in de meeste gevallen

de sterkte van de dwarsdoorsnede in de richting van de y-as groter is, terwijl de sterkte in de richting van de z-as meestal als de zwakkere wordt beschouwd.

3.1.1 Inleiding Eurocode 0

Voor eenzelfde balk moeten de interne krachten, reactiekrachten en vervormingen meerdere keren worden berekend. Dit komt omdat Eurocode 0 voorschrijft dat er rekening moet worden gehouden met verschillende grenstoestanden, ontwerpsituaties en belastingscombinaties [8]. Om aan deze vereisten te voldoen wordt gebruikt gemaakt van partiële factoren en combinatiefactoren. Deze termen en factoren worden hieronder toegelicht.

Grenstoestanden

Volgens Eurocode 0 moeten constructies worden ontworpen op basis van verschillende grenstoestanden. Een grenstoestand wordt gedefinieerd als “toestanden waarbij de constructie niet langer voldoet aan de van toepassing zijnde ontwerp- en berekeningscriteria” [8, p. 22]. Er bestaan twee verschillende typen grenstoestanden. De eerste is de uiterste grenstoestand (UGT), die de toestanden vertegenwoordigt waarbij het falen van de constructie optreedt. De tweede is de bruikbaarheidsgrenstoestand (BGT), die diverse voorwaarden bevat voor het gebruik en uiterlijk van de constructie, zoals doorbuiging, trillingen en scheurvorming [8].

Ontwerpsituaties

Daarnaast schrijft [8] voor dat er rekening moet worden gehouden met verschillende ontwerpsituaties. Een ontwerpsituatie wordt gedefinieerd als het “samenstellen van fysische omstandigheden die de reële omstandigheden representeren die optreden tijdens een zekere tijdsperiode, waarvoor het ontwerp en de berekening aantonen dat de van toepassing zijnde grenstoestanden niet worden overschreden” [8, p. 21]. In deze masterproef worden enkel berekeningen uitgevoerd voor tijdelijke en blijvende ontwerpsituaties. Buitengewone ontwerpsituaties, zoals die bijvoorbeeld bij brand, en aardbevingsontwerpsituaties worden dus niet beschouwd.

Belastingen

Belastingen moeten worden ingedeeld op basis van hun variatie in de tijd, zoals voorgeschreven door Eurocode 0 [8]. Er worden drie typen belastingen onderscheiden. De eerste is de blijvende belasting (aangeduid met de letter G of g^1), die een verwaarloosbare variatie in de tijd vertoont, zoals bijvoorbeeld het eigengewicht van de balk. Ten tweede is er de veranderlijke belasting (aangeduid met de letter Q of q), die een niet verwaarloosbare variatie in de tijd heeft, zoals wind, sneeuw of de nuttige belasting. Deze veranderlijke belastingen hebben elk drie combinatiefactoren, aangeduid als ψ_0 , ψ_1 en ψ_2 , om verschillende combinaties te maken, zoals verder uitgelegd bij de belastingscombinaties. Ten slotte zijn er de buitengewone belastingen (aangeduid met de letter A), maar deze worden niet behandeld in deze masterproef [8].

In het programma zal de gebruiker zelf de belastingwaarden moeten invoeren², met uitzondering van het eigengewicht van de balk. Het eigengewicht wordt automatisch berekend op basis van het gekozen materiaal en de doorsnede van de balk. Bovendien maakt het programma gebruik van

¹ Hoofdletters worden gebruikt voor puntlasten terwijl kleine letters worden gebruikt voor verdeelde belastingen.

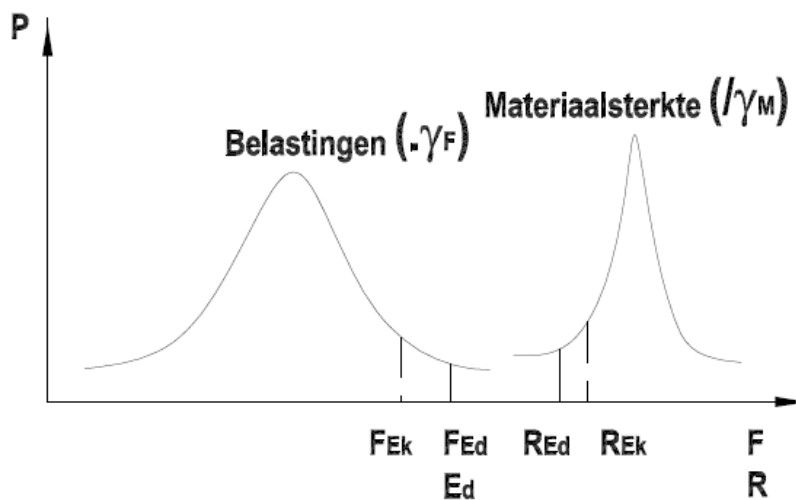
² De gebruiker kan slechts één waarde invoeren en dit dient de karakteristieke waarde te zijn.

belastingsgevallen als overkoepelende term voor verschillende soorten belastingen, zoals bijvoorbeeld sneeuw. Elk belastingsgeval kan diverse belastingen omvatten, zoals verdeelde belastingen, puntlasten en momenten, waarbij alle belastingen in één belastingsgeval dezelfde partiële en combinatiefactoren delen en steeds gelijktijdig optreden.

Methode van de partiële factoren

De methode van partiële factoren maakt gebruik van, zoals de naam al aangeeft, partiële factoren. Deze partiële factoren worden gebruikt om met onzekerheden om te gaan. Aanvankelijk worden karakteristieke waarden gebruikt voor belastingen en materiaaleigenschappen. De karakteristieke waarde, aangeduid met een kleine letter k als subscript, vertegenwoordigt de waarde waarbij er een vijf procent kans is dat deze wordt overschreden voor een belasting (F_k) en onderschreden voor een weerstand (R_k). Dit concept wordt geïllustreerd in Figuur 3.3 [26].

Bij de methode van partiële factoren worden deze karakteristieke waarden omgezet naar rekenwaarden, aangeduid met een kleine letter d als subscript. De rekenwaarde voor een belasting (F_d) wordt verkregen door de karakteristieke waarde van de belasting (F_{Ek}) te vermenigvuldigen met de partiële factor (γ_F) ervan. Voor een weerstand (R_d) wordt de rekenwaarde verkregen door de karakteristieke waarde van de weerstand (R_k) te delen door de partiële factor van het materiaal (γ_M). Aangezien de partiële factoren in de meeste gevallen groter zijn dan 1³, resulteert dit in een verhoging van de belastingswaarde en een verlaging van de weerstand, zoals geïllustreerd in Figuur 3.3 [26].



Figuur 3.3: Visuele voorstelling van de methode van de partiële factoren [26, p. 27]

De factor E_d , zoals weergegeven in Figuur 3.3, representeert het effect van de belastingen. Dit omvat het interne moment (M_d) of de interne dwarskracht (V_d) die optreedt binnen de constructie als gevolg van de belastingen. Deze methode beperkt de probabilistische kans dat de werkelijke waarde van de belasting groter is dan de werkelijke waarde van de weerstand. De grootte van deze kans wordt bepaald door de ontwerplevensduurcategorie en de gevolgklasse, zoals gespecificeerd in Eurocode 0 [8]. In deze masterproef wordt een ontwerplevensduurcategorie 4 (levensduur 50 jaar) en een gevolgklasse CC2 (residentiële gebouwen, kantoren, ...) aangenomen, dit komt overeen met een faalkans van 0,029% [27].

³ De waarden van de partiële factoren zijn vastgelegd in de Eurocodes. De partiële factoren voor de belastingen worden verder in deze sectie toegelicht, de partiële factoren voor de materialen worden besproken in de corresponderende secties van de materialen.

Belastingscombinaties

Om de interne krachten en vervormingen in een element of constructie te bepalen, wordt niet simpelweg de rekenwaarde voor elke mogelijke belasting genomen. Dit zou namelijk leiden tot aanzienlijke interne krachten en vervormingen die bovendien niet realistisch zouden zijn, omdat niet alle belastingen gelijktijdig optreden. In plaats daarvan worden belastingscombinaties gemaakt. De definitie van een belastingscombinatie volgens Eurocode 0 is: “Een samenstel van rekenwaarden gebruikt voor de toetsing van de constructieve betrouwbaarheid bij een grenstoestand waarbij verscheidene belastingen gelijktijdig werken” [8, p. 27]. Uit deze definitie volgt dat er combinaties moeten worden gemaakt voor elke grenstoestand.

In deze masterproef worden belastingscombinaties gemaakt voor de blijvende ontwerpsituatie (UGT-B) in de uiterste grenstoestand, de karakteristieke combinatie (BGT-K), de frequente combinatie (BGT-F), en de quasi-permanente combinatie (BGT-Q) voor de bruikbaarheidsgrenstoestand. UGT-B wordt gebruikt om de balk te ontwerpen en te controleren op basis van sterkte en stabiliteit. BGT-K wordt toegepast om onomkeerbare grenstoestanden te controleren, terwijl BGT-F bedoeld is voor het controleren van omkeerbare grenstoestanden. BGT-Q wordt ten slotte gebruikt voor langetermijneffecten en het uiterlijk van de constructie te controleren. De formules voor deze combinaties zijn gedefinieerd in Eurocode 0 en worden weergegeven in (3.1), (3.2), (3.3), en (3.4) [8].

$$\text{UGT-B} : \sum_{j \geq 1} \gamma_{Gj} G_{kj} + \gamma_P P_k + \gamma_{Q1} Q_{k1} + \sum_{i > 1} \gamma_{Qi} \psi_{0i} Q_{ki} \quad (3.1)$$

$$\text{BGT-K} : \sum_{j \geq 1} G_{kj} + P_k + Q_{k1} + \sum_{i > 1} \psi_{0i} Q_{ki} \quad (3.2)$$

$$\text{BGT-F} : \sum_{j \geq 1} G_{kj} + P_k + \psi_{11} Q_{k1} + \sum_{i > 1} \psi_{2i} Q_{ki} \quad (3.3)$$

$$\text{BGT-Q} : \sum_{j \geq 1} G_{kj} + P_k + \sum_{i > 1} \psi_{2i} Q_{ki} \quad (3.4)$$

De formules voor UGT-B, BGT-K en BGT-F bestaan elk uit vier termen. De eerste term omvat de som van alle rekenwaarden van de blijvende belastingen. De tweede term, die de voorspanning vertegenwoordigt, wordt in deze masterproef niet beschouwd en is dus gelijk aan nul. De derde term houdt rekening met de rekenwaarde van de overheersende veranderlijke belasting. Indien er slechts één veranderlijke belasting is die op een bepaald moment kan optreden, is deze automatisch de overheersende veranderlijke belasting. Als er meerdere gelijktijdige veranderlijke belastingen zijn, moeten verschillende combinaties worden gemaakt waarbij telkens een andere veranderlijke belasting als overheersend wordt beschouwd. De vierde term is alleen aanwezig indien er meer dan één veranderlijke belasting optreedt en bevat de som van alle andere gelijktijdige veranderlijke belastingen die in deze combinatie niet worden beschouwd als overheersend. Hierbij worden combinatiefactoren toegepast om de waarde van deze belastingen te verlagen.

Bij de formule voor BGT-Q wordt geen onderscheid gemaakt tussen overheersende en niet-overheersende veranderlijke belastingen. In deze formule staat dus, na het verwijderen van de factor voor voorspanning, één term voor de blijvende belastingen en één term voor alle veranderlijke belastingen.

De combinatiefactoren worden gebruikt om nieuwe waarden voor de veranderlijke belastingen te verkrijgen. Hiervoor worden de combinatiefactoren steeds vermenigvuldigd met de karakteristieke waarde van een veranderlijke belasting. Er zijn in totaal drie combinatiefactoren, waardoor drie nieuwe waarden voor veranderlijke belastingen worden verkregen. Ten eerste is er de combinatiewaarde $\psi_0 Q_k$, die ervoor zorgt dat niet voor elke gelijktijdige veranderlijke belasting de meest ongunstige waarde wordt verrekend. Ten tweede is er de frequente waarde $\psi_1 Q_k$, die is bepaald op basis van de duur waarin haar waarde wordt overschreden in verhouding tot een vooraf gekozen tijdspanne, of de frequentie van de overschrijding begrensd is tot een gegeven waarde. Ten slotte is er de quasi-blijvende waarde $\psi_2 Q_k$, die wordt bepaald op basis van het feit dat de waarde gedurende een aanzienlijk deel van een vooraf gedefinieerde periode wordt overschreden [8]. De waarden van de verschillende combinatiefactoren voor verschillende belastingen zoals wind, sneeuw, etc. zijn vastgelegd in Tabel A1.1 ANB in de Nationale bijlage van Eurocode 0 [28].

In de belastingscombinatie voor de uiterste grenstoestand, formule (3.1), moeten de partiële factoren voor belastingen in rekening worden gebracht. De waarden van deze partiële factoren zijn vastgelegd in Tabel A1.2 (B) ANB van de Nationale bijlage van Eurocode 0 [28]. De partiële factoren voor belastingen die van toepassing zijn in deze masterproef worden getoond in Tabel 3.1.

Tabel 3.1: Partiële factoren voor belastingen

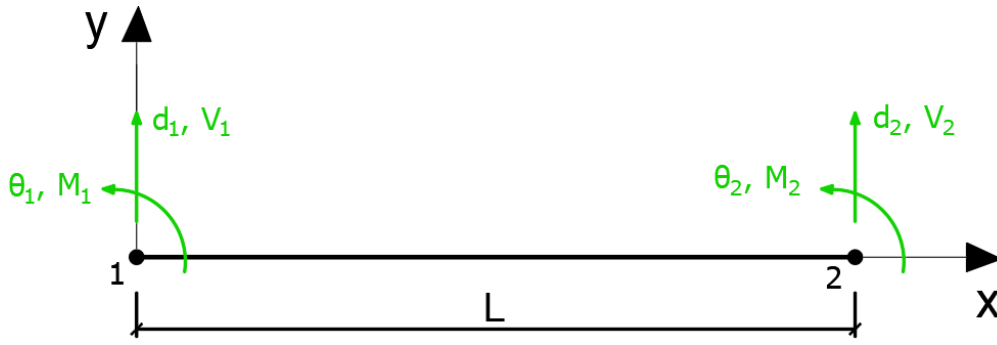
Belasting	Partiële factor indien gunstig [-]	Partiële factor indien ongunstig [-]
Blijvend	1	1,35
Veranderlijk	0	1,5

In Tabel 3.1 wordt een onderscheid gemaakt tussen de partiële factoren bij gunstige en ongunstige werking van de belastingen. Een belasting kan gunstig werken als deze in de tegengestelde richting werkt dan de overheersende veranderlijke belasting. Daarnaast kan een belasting gunstig werken in bepaalde gevallen als deze op een ander vlak optreedt dan de overheersende veranderlijke belasting en onafhankelijk is van deze veranderlijke belasting.

Het resultaat van deze belastingscombinaties is dat voor elke belasting die op de balk inwerkt, een specifieke rekenwaarde wordt bepaald. Deze waarden worden berekend voor verschillende belastingscombinaties, waarbij de rekenwaarde voor dezelfde belasting kan variëren afhankelijk van de combinatie. Deze rekenwaarden worden vervolgens gebruikt in de eindige-elementenmethode om de balk te analyseren.

3.1.2 Eindige-elementenmethode

De eindige-elementenmethode (EEM) verdeelt de balk in een eindig aantal kleinere elementen, waarbij elk element knooppunten heeft aan beide uiteinden. Met deze methode worden de vervorming, rotatie en interne krachten (snedekrachten) bij elk knooppunt bepaald. In deze masterproef worden alleen verticale belastingen (krachten in de y-richting) en momenten rond de z-as in beschouwing genomen, waardoor de interne krachten beperkt blijven tot de dwarskracht in verticale richting (aangeduid als V) en het moment rond de z-as (aangeduid als M). De EEM gebruikt de relatie tussen krachten en verplaatsingen, waardoor in deze masterproef de verticale verplaatsingen (aangeduid als d) en de rotatie rond de z-as (aangeduid als θ) worden berekend. Deze krachten en verplaatsingen zijn geïllustreerd in Figuur 3.4 [29].



Figuur 3.4: Tekenconventie eindige-elementenmethode

Zoals getoond in Figuur 3.4 zijn de dwarskracht en vervorming positief wanneer ze naar boven gericht zijn. Het moment en de rotatie zijn positief wanneer ze tegen de wijzers van de klok in werken. Merk op dat deze tekenconventie verschilt van die aan het begin van deze sectie. Om de tekenconventie van de eindige-elementenmethode om te zetten naar de aangenomen tekenconventie, moeten de tekens van het moment rond de eerste knoop en de dwarskracht op de tweede knoop worden omgekeerd bij de berekening van de interne krachten.

De eindige-elementenmethode maakt gebruik van een verband tussen een verplaatsingsvector $\vec{u}$, een krachtvector $\vec{F}$ en een globale stijfheidsmatrix K_{global} . Het verband tussen deze matrices wordt getoond door formule (3.5).

$$\vec{F} = K_{global} \cdot \vec{u} \quad (3.5)$$

De globale stijfheidsmatrix wordt opgesteld op basis van de eigenschappen van de balk, zoals het materiaal en de doorsnede. De krachtvector wordt samengesteld op basis van de steunpunten en de belastingen op de balk. De verplaatsingsvector wordt eveneens opgesteld op basis van de steunpunten. Vervolgens worden de verplaatsingen bepaald met behulp van het verband in formule (3.5). De waarden van de verplaatsingen in de verplaatsingsvector worden daarna gebruikt om de reactiekrachten en interne krachten te berekenen. Deze stappen worden in het volgende gedeelte nader toegelicht.

Globale stijfheidsmatrix

Elk element in de balk wordt gerepresenteerd door een matrix die de stijfheid ervan aangeeft. De stijfheidsmatrix van één element is een 4x4 matrix, waarbij elke rij en kolom een vrijheidsgraad van een knooppunt representeert. De eerste rij en kolom staan bijvoorbeeld voor de verticale vervorming van het eerste knooppunt. Formule (3.6) toont de stijfheidsmatrix van een element.

$$K_{element} = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{matrix} d_1 \\ \theta_1 \\ d_2 \\ \theta_2 \end{matrix} \quad (3.6)$$

waarin:

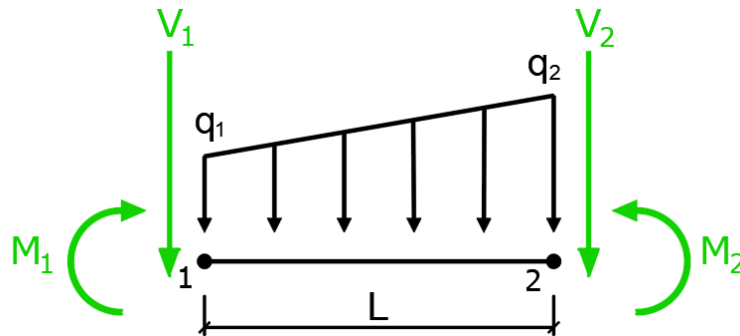
- L de lengte van het element is;
- E de elasticiteitsmodulus van het element is;
- I het traagheidsmoment van het element is.

De globale stijfheidsmatrix wordt gevonden door de waarden die overeenkomen met dezelfde vrijheidsgraden in de stijfheidsmatrix van elk element bij elkaar op te tellen. Dit resulteert in een vierkante matrix met een grootte van het totaal aantal vrijheidsgraden in de balk. Dit is gelijk aan tweemaal het aantal knopen in de balk.

Krachtvector

De krachtvector is een kolommatrix met een grootte gelijk aan tweemaal het aantal knopen in de balk. In deze matrix vertegenwoordigen de eerste twee waarden de eerste knoop, de volgende twee waarden de tweede knoop, enzovoort. De eerste waarde van elk paar stelt de verticale kracht op die knoop voor, terwijl de tweede waarde het moment rond die knoop weergeeft.

In het programma kunnen gebruikers een puntlast, een moment rond een punt, en een verdeelde belasting aanbrengen. Voor een puntlast en een moment moeten respectievelijk de eerste en tweede waarde van de corresponderende knoop worden ingevuld. Bij een verdeelde belasting moeten equivalente krachten op de knopen worden berekend, gelijk aan de verdeelde belasting op het element tussen deze knopen. Dit wordt geïllustreerd in Figuur 3.5.



Figuur 3.5: Equivalente knooppkrachten

De formules om deze krachten te berekenen, zijn weergegeven in formules (3.7)-(3.10). Bij het invoeren van de waarden in de krachtmatrix moet het correcte teken worden gebruikt, zodat deze overeenkomen met de tekenconventie van deze methode (zie Figuur 3.4).

$$V_1 = \frac{L \times (21 \times q_1 + 9 \times q_2)}{60} \quad (3.7)$$

$$M_1 = L \times \frac{L \times (3 \times q_1 + 2 \times q_2)}{60} \quad (3.8)$$

$$V_2 = \frac{L \times (9 \times q_1 + 21 \times q_2)}{60} \quad (3.9)$$

$$M_2 = L \times \frac{L \times (2 \times q_1 + 3 \times q_2)}{60} \quad (3.10)$$

De krachten in de krachtmatrix vertegenwoordigen de totale kracht op deze knoop of het totale moment rond deze knoop. Hierbij moet ook de reactiekracht van een eventueel steunpunt van deze knoop worden meegenomen. In het programma kan gekozen worden tussen een roloplegging, een scharnier en een inklemming. Bij een roloplegging en een scharnier wordt een onbekende toegevoegd op de plaats van dwarskracht van deze knoop. Bij een inklemming wordt zowel voor de dwarskracht als voor het moment een onbekende toegevoegd.

Verplaatsingsvector

De verplaatsingsvector is ook een kolommatrix met dezelfde grootte als de krachtmatrix. In de verplaatsingsvector wordt een knoop opnieuw vertegenwoordigd als een paar van twee waardes. Nu stelt de eerste waarde de verticale verplaatsing voor en de tweede waarde de hoekrotatie. Indien er een roloplegging of scharnier plaats vindt op een knoop wordt de eerste waarde gelijk aan nul gesteld. Indien er een inklemming plaats vindt worden beide waardes aan nul gelijk gesteld. De overige waardes worden als onbekendes ingevuld.

Berekening verplaatsingen

Om alle onbekende verplaatsingen en rotaties te berekenen, wordt gebruikgemaakt van het verband in formule (3.5). Hierbij wordt de krachtenmatrix $\vec{F}$ vermenigvuldigd met de inverse van de globale stijfheidsmatrix $K_{globaal}$. Voordat deze vermenigvuldiging plaatsvindt, moeten alle rijen en kolommen die overeenkomen met een beperking van de verplaatsing of rotatie (door een steunpunt) worden verwijderd uit beide matrices. Het resultaat van deze vermenigvuldiging is een matrix die alle onbekende verplaatsingen en rotaties bevat.

Berekening reactiekrachten

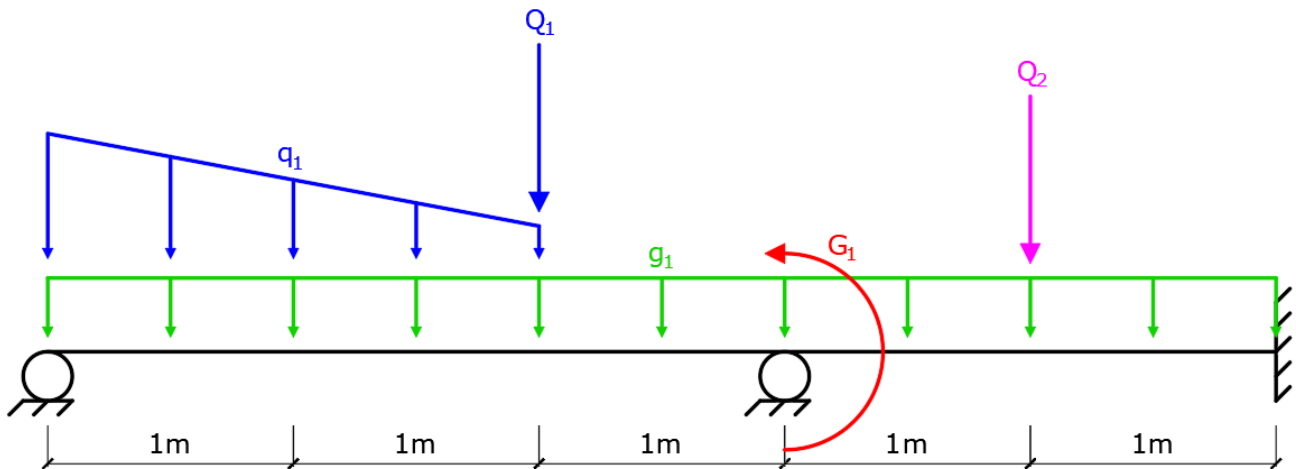
Om de reactiekrachten te bepalen, volgt eerst de berekening van de krachten in de krachtmatrix. Deze worden vervolgens gecorrigeerd met de overeenkomstige waarden van de belastingen in de krachtmatrix. Een kracht in de krachtmatrix wordt afgeleid door een rij in de globale stijfheidsmatrix te beschouwen als een rijvector. Deze rijvector wordt dan vermenigvuldigd met de kolomvector van verplaatsingen en rotaties om de kracht van deze rij te berekenen. Deze berekening wordt herhaald voor elke rij waarin een reactiekracht aanwezig is, waardoor na het corrigeren alle reactiekrachten worden gevonden.

Berekening interne krachten

Om de interne krachten te berekenen, wordt elk element afzonderlijk geanalyseerd. De dwarskracht en het moment bij beide knopen van een element worden bepaald door de elementstijfheidsmatrix te vermenigvuldigen met een matrix van de corresponderende verplaatsingen en rotaties. De verkregen waarden moeten vervolgens worden gecorrigeerd met de belastingen op het element en de knopen om de interne krachten te verkrijgen. Let hierbij op dat de tekens van het moment rond de eerste knoop en de dwarskracht op de tweede knoop worden omgedraaid.

3.1.3 Voorbeeld

Om de theorie te verduidelijken, wordt in deze sectie een fictief voorbeeld uitgewerkt. De configuratie van dit voorbeeld wordt weergegeven in Figuur 3.6, terwijl de belastingwaarden te vinden zijn in Tabel 3.2. Elke kleur van de belasting correspondeert met een specifiek belastingsgeval. Het is belangrijk te vermelden dat de belastingwaarden de karakteristieke waarden weergeven.



Figuur 3.6: Configuratie voorbeeld analyse

Tabel 3.2: Belastingenwaarden voorbeeld analyse

Belasting	Belastingsgeval	Karakteristieke waarde	ψ_0	ψ_1	ψ_2
Verdeelde belasting g_1	Eigengewicht	0,22 - 0,22 kN/m	-	-	-
Moment G_1	Blijvend	25 kNm	-	-	-
Verdeelde belasting q_1	Veranderlijke 1	40 - 10 kN/m	0,7	0,5	0,3
Puntlast Q_1	Veranderlijke 1	50 kN	0,7	0,5	0,3
Puntlast Q_2	Veranderlijke 2	10 kN	0,7	0,7	0,6

Zoals geïllustreerd in Figuur 3.6 betreft het een hyperstatische balk met een lengte van vijf meter. De balk is aan één uiteinde ingeklemd en heeft daarnaast twee rolopleggingen. Over de gehele lengte werkt een uniforme verdeelde belasting g_1 , die het eigengewicht van de balk voorstelt en daarmee een blijvende belasting is. Op drie meter van het begin van de balk is bovendien een blijvend moment G_2 aanwezig. Naast deze blijvende belastingen zijn er twee veranderlijke belastingsgevallen. In het eerste veranderlijke belastingsgeval werkt een niet-uniforme verdeelde belasting q_1 van het begin van de balk tot op twee meter, met op die positie ook een puntlast Q_1 . In het tweede veranderlijke belastingsgeval is er een puntlast Q_2 op vier meter van het begin van de balk.

In dit voorbeeld worden belastingscombinaties opgesteld voor UGT-B, BGT-K, BGT-F en BGT-Q. De combinaties voor UGT-B zijn weergegeven in Tabel 3.3, de combinaties voor BGT-K in Tabel 3.4, de combinaties voor BGT-F in Tabel 3.5, en de combinaties voor BGT-Q in Tabel 3.6. Elke combinatie wordt weergegeven als een som van belastingsgevallen, waarbij elk belastingsgeval wordt voorgesteld door twee factoren die met de karakteristieke waarde worden vermenigvuldigd. De eerste factor is de partiële factor en de tweede factor is één- of een ψ -factor.

Tabel 3.3: Combinaties voor UGT-B

Naam	Eigengewicht	Blijvend	Veranderlijke 1	Veranderlijke 2
UGT-B 1	$1,35 \times 1,00$	$1,35 \times 1,00$	$1,50 \times 1,00$	$1,50 \times 0,70$
UGT-B 2	$1,35 \times 1,00$	$1,35 \times 1,00$	$1,50 \times 0,70$	$1,50 \times 1,00$
UGT-B 3	$1,35 \times 1,00$	$1,35 \times 1,00$	$1,50 \times 1,00$	0,00
UGT-B 4	$1,35 \times 1,00$	$1,35 \times 1,00$	0,00	$1,50 \times 1,00$
UGT-B 5	$1,35 \times 1,00$	$1,35 \times 1,00$	0,00	0,00
UGT-B 6	$1,35 \times 1,00$	$1,00 \times 1,00$	$1,50 \times 1,00$	$1,50 \times 0,70$
UGT-B 7	$1,35 \times 1,00$	$1,00 \times 1,00$	$1,50 \times 0,70$	$1,50 \times 1,00$
UGT-B 8	$1,35 \times 1,00$	$1,00 \times 1,00$	$1,50 \times 1,00$	0,00
UGT-B 9	$1,35 \times 1,00$	$1,00 \times 1,00$	0,00	$1,50 \times 1,00$
UGT-B 10	$1,35 \times 1,00$	$1,00 \times 1,00$	0,00	0,00
UGT-B 11	$1,00 \times 1,00$	$1,35 \times 1,00$	$1,50 \times 1,00$	$1,50 \times 0,70$
UGT-B 12	$1,00 \times 1,00$	$1,35 \times 1,00$	$1,50 \times 0,70$	$1,50 \times 1,00$
UGT-B 13	$1,00 \times 1,00$	$1,35 \times 1,00$	$1,50 \times 1,00$	0,00
UGT-B 14	$1,00 \times 1,00$	$1,35 \times 1,00$	0,00	$1,50 \times 1,00$
UGT-B 15	$1,00 \times 1,00$	$1,35 \times 1,00$	0,00	0,00
UGT-B 16	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,50 \times 1,00$	$1,50 \times 0,70$
UGT-B 17	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,50 \times 0,70$	$1,50 \times 1,00$
UGT-B 18	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,50 \times 1,00$	0,00
UGT-B 19	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	$1,50 \times 1,00$
UGT-B 20	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	0,00

Tabel 3.4: Combinaties voor BGT-K

Naam	Eigengewicht	Blijvend	Veranderlijke 1	Veranderlijke 2
BGT-K 1	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 0,70$
BGT-K 2	$1,00 \times 1,00$	$1,00 \times 1,00$	$0,70 \times 1,00$	$1,00 \times 1,00$
BGT-K 3	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00
BGT-K 4	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	$1,00 \times 1,00$
BGT-K 5	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	0,00

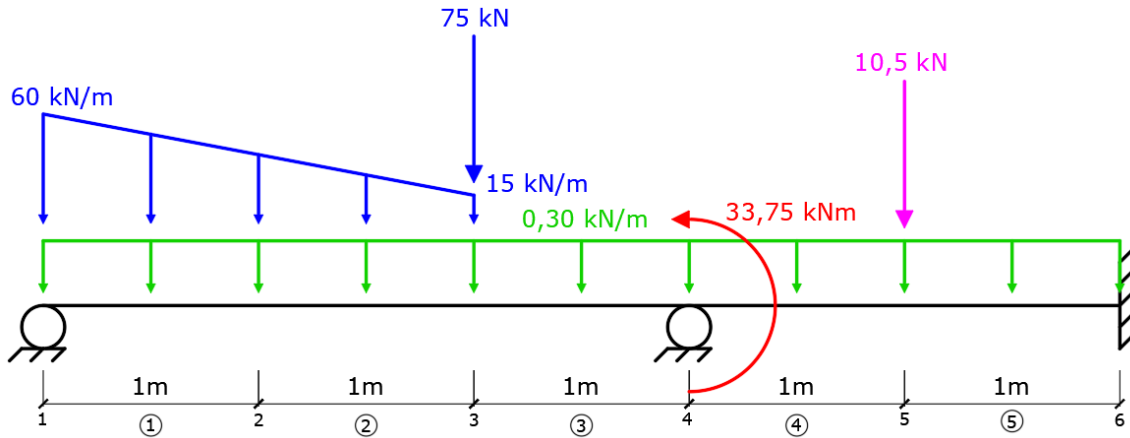
Tabel 3.5: Combinaties voor BGT-F

Naam	Eigengewicht	Blijvend	Veranderlijke 1	Veranderlijke 2
BGT-F 1	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 0,50$	$1,00 \times 0,60$
BGT-F 2	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 0,30$	$1,00 \times 0,70$
BGT-F 3	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 0,50$	0,00
BGT-F 4	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	$1,00 \times 0,70$
BGT-F 5	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	0,00

Tabel 3.6: Combinaties voor BGT-Q

Naam	Eigengewicht	Blijvend	Veranderlijke 1	Veranderlijke 2
BGT-Q 1	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 0,30$	$1,00 \times 0,60$
BGT-Q 2	$1,00 \times 1,00$	$1,00 \times 1,00$	$1,00 \times 0,30$	0,00
BGT-Q 3	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	$1,00 \times 0,60$
BGT-Q 4	$1,00 \times 1,00$	$1,00 \times 1,00$	0,00	0,00

Er zijn vier keer zoveel combinaties voor de uiterste grenstoestand dan voor de bruikbaarheids-grenstoestanden in dit voorbeeld. Dit komt doordat de partiële factor verschillende waarden aanneemt, afhankelijk van gunstige of ongunstige omstandigheden bij de uiterste grenstoestand, terwijl deze gelijk blijft bij de bruikbaarheidsgrenstoestand. Voor verdere uitwerking van dit voorbeeld wordt de belastingscombinatie “UGT-B 1” gebruikt. Een schematische weergave van dit voorbeeld, inclusief ingevulde belastingswaarden, wordt getoond in Figuur 3.7.



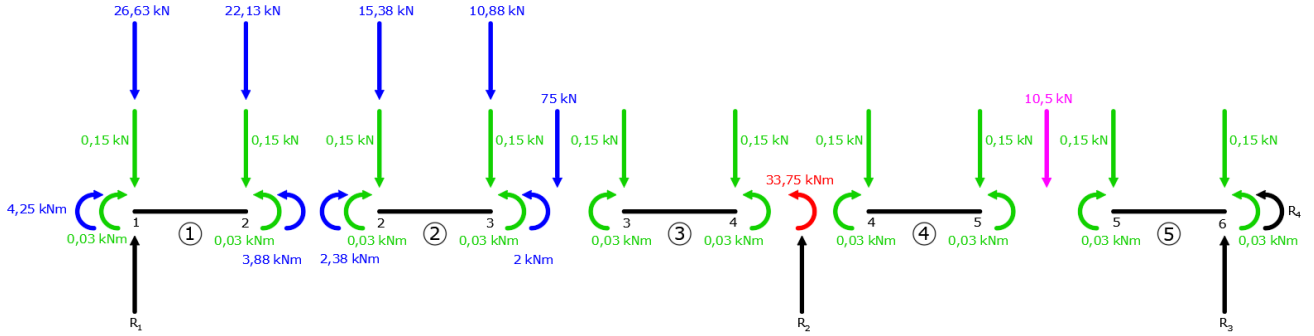
Figuur 3.7: Configuratie voorbeeld analyse met ingevulde waarden

Zoals te zien is in Figuur 3.7, is de balk opgedeeld in vijf elementen en zes knooppunten, waarbij elk element een lengte van één meter heeft. Hoewel in dit voorbeeld alle elementen even lang zijn, wordt in de applicatie de lengte van de elementen bepaald door de gegenereerde mesh, zoals wordt beschreven in Sectie 4.2.3. In dit voorbeeld is een stalen IPE200-ligger gebruikt met een elasticiteitsmodulus van $E = 210 \times 10^6 \text{ kN/m}^2$ en een traagheidsmoment van $I = 1943 \times 10^{-8} \text{ m}^4$ rond de z-as. Aangezien alle elementen dezelfde afmetingen en eigenschappen hebben, zijn hun stijfheidsmatrices identiek. De stijfheidsmatrix van één element wordt weergegeven in formule (3.11).

$$\begin{aligned}
 \frac{12EI}{L^3} &= \frac{12 \times 210 \times 10^6 \text{ kN/m}^2 \times 1943 \times 10^{-8} \text{ m}^4}{(1 \text{ m})^3} = 48964 \text{ kN/m} \\
 \frac{6EI}{L^2} &= \frac{6 \times 210 \times 10^6 \text{ kN/m}^2 \times 1943 \times 10^{-8} \text{ m}^4}{(1 \text{ m})^2} = 24482 \text{ kN} \\
 \frac{4EI}{L} &= \frac{4 \times 210 \times 10^6 \text{ kN/m}^2 \times 1943 \times 10^{-8} \text{ m}^4}{1 \text{ m}} = 16321 \text{ kNm} \\
 \frac{2EI}{L} &= \frac{2 \times 210 \times 10^6 \text{ kN/m}^2 \times 1943 \times 10^{-8} \text{ m}^4}{1 \text{ m}} = 8161 \text{ kNm}
 \end{aligned}$$

$$K_{\text{element},i} = \begin{bmatrix} 48964 & 24482 & -48964 & 24482 \\ 24482 & 16321 & -24482 & 8161 \\ -48964 & -24482 & 48964 & -24482 \\ 24482 & 8161 & -24482 & 16321 \end{bmatrix} \quad (3.11)$$

De globale stijfheidsmatrix K_{globaal} wordt weergegeven in formule (3.13) op pagina 67. Deze matrix heeft een dimensie van twaalf bij twaalf, omdat er zes knooppunten zijn met elk twee vrijheidsgraden. De volgende stap betreft het opstellen van de krachtvector $\vec{F}$. Hierbij moeten alle belastingen worden omgerekend naar equivalente krachten op de knooppunten. De resulterende knooppuntenkrachten in dit voorbeeld worden weergegeven in Figuur 3.8.



Figuur 3.8: Knooppuntenkrachten voorbeeld

Zoals te zien is in Figuur 3.8, representeren de kleuren van de krachten hetzelfde belastinggeval als de kleuren in Figuur 3.6. Daarnaast stellen de vier zwarte krachten in Figuur 3.8 de reactiekrachten in de steunpunten voor. Omdat deze nog niet gekend zijn, worden deze voorgesteld met de letter R . Om de krachtmatrix op te stellen, moet voor elke knoop de som van alle verticale krachten op deze knoop en de som van alle momenten rond deze knoop berekend worden. De krachtmatrix wordt weergegeven in formule (3.12).

$$\vec{F} = \begin{bmatrix} F_{K1} \\ M_{K1} \\ F_{K2} \\ M_{K2} \\ F_{K3} \\ M_{K3} \\ F_{K4} \\ M_{K4} \\ F_{K5} \\ M_{K5} \\ F_{K6} \\ M_{K6} \end{bmatrix} = \begin{bmatrix} R_1 - 26,78 \text{ kN} \\ -4,28 \text{ kNm} \\ -37,8 \text{ kN} \\ 1,5 \text{ kNm} \\ -86,18 \text{ kN} \\ 2 \text{ kNm} \\ R_2 - 0,3 \text{ kN} \\ 33,75 \text{ kNm} \\ -10,8 \text{ kN} \\ 0 \text{ kNm} \\ R_3 - 0,15 \text{ kN} \\ R_4 + 0,03 \text{ kNm} \end{bmatrix} \quad (3.12)$$

$$K_{\text{global}} = \begin{bmatrix} 48964 & 24482 & -48964 & 24482 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 24482 & 16321 & -24482 & 8161 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -48964 & -24482 & 97927 & 0 & -48964 & 24482 & 0 & 0 & 0 & 0 & 0 & 0 \\ 24482 & 8161 & 0 & 32642 & -24482 & 8161 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -48964 & -24482 & 97927 & 0 & -48964 & 24482 & 0 & 0 & 0 & 0 \\ 0 & 0 & 24482 & 8161 & 0 & 32642 & -24482 & 8161 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -48964 & -24482 & 97927 & 0 & -48964 & 24482 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24482 & 8161 & 0 & 32642 & -24482 & 8161 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -48964 & -24482 & 97927 & 0 & -48964 & 24482 \\ 0 & 0 & 0 & 0 & 0 & 0 & 24482 & 8161 & 0 & 32642 & -24482 & 8161 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -48964 & -24482 & 48964 & -24482 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24482 & 8161 & -24482 & 16321 \end{bmatrix} \quad (3.13)$$

$$\vec{u} = \begin{bmatrix} 16321 & -24482 & 8161 & 0 & 0 & 0 & 0 & 0 \\ -24482 & 97927 & 0 & -48964 & 24482 & 0 & 0 & 0 \\ 8161 & 0 & 32642 & -24482 & 8161 & 0 & 0 & 0 \\ 0 & -48964 & -24482 & 97927 & 0 & 24482 & 0 & 0 \\ 0 & 24482 & 8161 & 0 & 32642 & 8161 & 0 & 0 \\ 0 & 0 & 0 & 24482 & 8161 & 32642 & -24482 & 8161 \\ 0 & 0 & 0 & 0 & 0 & -24482 & 97927 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8161 & 0 & 32642 \end{bmatrix}^{-1} \begin{bmatrix} -4,28 \\ -37,8 \\ 1,5 \\ -86,18 \\ 2 \\ 33,75 \\ -10,8 \\ 0 \end{bmatrix} = \begin{bmatrix} -0,01151 \\ -0,00926 \\ -0,00528 \\ -0,00882 \\ 0,00633 \\ 0,00798 \\ 0,00186 \\ -0,0020 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ d_2 \\ \theta_2 \\ d_3 \\ \theta_3 \\ \theta_4 \\ d_5 \\ \theta_5 \end{bmatrix} \quad (3.14)$$

De verplaatsingsvector $\vec{u}$ wordt samengesteld door de waarden op de posities waar de verplaatsingen en rotaties worden beperkt door de steunpunten op nul te zetten. De overige waarden worden aangeduid als onbekenden. Deze verplaatsingsvector is weergegeven in formule (3.15).

$$\vec{u} = \begin{bmatrix} 0 \\ \theta_1 \\ d_2 \\ \theta_2 \\ d_3 \\ \theta_3 \\ 0 \\ \theta_4 \\ d_5 \\ \theta_5 \\ 0 \\ 0 \end{bmatrix} \quad (3.15)$$

Om de verplaatsingen en rotaties te berekenen worden eerst de rijen en kolommen die overeenkomen met een nul in de verplaatsingsvector verwijderd uit de globale stijfheidsmatrix en krachtenmatrix. Nadat dit gebeurd is, worden de verplaatsingen en rotaties berekend door de inverse van $K_{globaal}$ te vermenigvuldigen met $\vec{F}$. Deze berekening wordt getoond in formule (3.14).

De reactiekrachten in de steunpunten worden nu berekend zoals uitgelegd in Sectie 3.1.2. De berekening voor de reactiekracht van de roloplegging in de eerste knoop wordt getoond in formules (3.16) en (3.17). In Tabel 3.7 worden alle reactiekrachten weergegeven.

$$F_{K1} = 24482 \times (-0,01151) - 48964 \times (-0,00926) + 24482 \times (-0,00528) = 42,35 \text{ kN} \quad (3.16)$$

$$R_1 = F_{K1} + 26,78 \text{ kN} = 42,35 \text{ kN} + 26,78 \text{ kN} = 69,13 \text{ kN} \quad (3.17)$$

Tabel 3.7: Reactiekrachten voorbeeld

Steunpunt	Positie [m]	Reactiekracht verticaal [kN]	Reactiekracht moment [kNm]
Roloplegging	0	69,13	-
Roloplegging	3	136,22	-
Inklemming	5	-43,30	29,84

De interne krachten worden berekend door de element stijfheidsmatrices te vermenigvuldigen met de overeenkomstige verplaatsingen en rotaties. Vervolgens wordt de verkregen waarde gecorrigeerd met de equivalente kracht op het knooppunt die alleen door de belasting op dat specifieke element wordt geïntroduceerd. Dit proces wordt geïllustreerd in de vergelijkingen (3.18) en (3.19) voor het eerste element. Voor knooppunten die aan twee elementen zijn verbonden, moeten de interne krachten voor elk element afzonderlijk worden bepaald, omdat de waarden wanneer er een verticale kracht of een moment op het knooppunt werkt. Een overzicht van alle interne krachten voor elk knooppunt is te vinden in Tabel 3.8.

$$\begin{bmatrix} F_{K1} \\ M_{K1} \\ F_{K2} \\ M_{K2} \end{bmatrix} = \begin{bmatrix} 48964 & 24482 & -48964 & 24482 \\ 24482 & 16321 & -24482 & 8161 \\ -48964 & -24482 & 48964 & -24482 \\ 24482 & 8161 & -24482 & 16321 \end{bmatrix} \begin{bmatrix} 0 \\ -0,01151 \\ -0,00926 \\ -0,00528 \end{bmatrix} = \begin{bmatrix} 42,35 \\ -4,28 \\ -42,7 \\ 42,68 \end{bmatrix} \quad (3.18)$$

$$\begin{bmatrix} V_1 \\ M_1 \\ V_2 \\ M_2 \end{bmatrix} = \begin{bmatrix} 42,35 \\ -4,28 \\ -42,31 \\ 46,70 \end{bmatrix} - \begin{bmatrix} -26,78 \\ -4,28 \\ -22,28 \\ 3,91 \end{bmatrix} = \begin{bmatrix} 69,13 \text{ kN} \\ 0 \text{ kNm} \\ -20,03 \text{ kN} \\ 42,79 \text{ kNm} \end{bmatrix} \quad (3.19)$$

Tabel 3.8: Interne krachten voorbeeld

Positie [m]	Dwarskracht 1 [kN]	Dwarskracht 2 [kN]	Moment 1 [kNm]	Moment 2 [kNm]
0	69,13	69,13	0	0
1	20,03	20,03	42,79	42,79
2	-6,52	-81,52	47,56	47,56
3	-81,82	54,40	-34,11	-67,86
4	54,1	43,6	-13,61	-13,61
5	43,3	43,3	29,84	29,84

Merk op dat het teken van de eerste waarde van de dwarskracht op een positie van één meter omgekeerd is ten opzichte van de waarde in formule (3.19). Dit is gedaan om de tekenconventie van de eindige-elementenmethode om te zetten naar de tekenconventie die in deze masterproef wordt gehanteerd, zoals uitgelegd in Sectie 3.1.2.

3.2 Beton

In deze sectie worden de formules en controles voor het dimensioneren van gewapende betonnen balken uiteengezet. Deze methoden zijn gebaseerd op de ontwerpeisen en principes van Eurocode 2. Allereerst worden de aannames en beperkingen besproken die gelden voor het ontwerpen van betonnen balken in deze masterproef. Vervolgens wordt algemene informatie over beton als materiaal toegelicht, wat nodig is voor de daaropvolgende berekeningen. Daarna wordt beschreven hoe de langs- en dwarskrachtwapening in de uiterste grenstoestand wordt berekend. Ten slotte worden de controles voor spanningen, scheuren en doorbuiging in de bruikbaarheidsgrenstoestanden behandeld.

3.2.1 Aannames en beperkingen

In het programma kunnen alleen eenvoudige gewapende betonnen balken worden ontworpen. Dit houdt in dat het gaat om niet-voorgespannen, in-situ gestorte betonbalken met slechts één laag niet-gebundelde wapeningsstaven. Daarnaast hebben de balken over hun volledige lengte een constante rechthoekige doorsnede en is C50/60 de hoogst mogelijke betonkwaliteit in deze masterproef. De wapening bestaat altijd uit staalkwaliteit BE500S.

Het programma kan geen berekeningen uitvoeren voor wringing, vermoeiing, gedrongen liggers, omsloten beton en tweede-orde effecten. Bovendien wordt er ook geen controle uitgevoerd op het kippen van de balk.

Voor de wapeningsconfiguratie wordt aangenomen dat een wapeningsstaaf oneindig lang is. Dit betekent dat er geen rekening wordt gehouden met overlappingsen en de hierbij nodige overlappingslengte. Deze vereenvoudiging betekent dat het ontwerp niet volledig voldoet aan de detailleringsregels van Eurocode 2. Bijlage C bevat een opsomming van alle beginselen en toepassingsregels van Eurocode 2 die door deze aannames en beperkingen niet van toepassing zijn.

3.2.2 Algemene eigenschappen

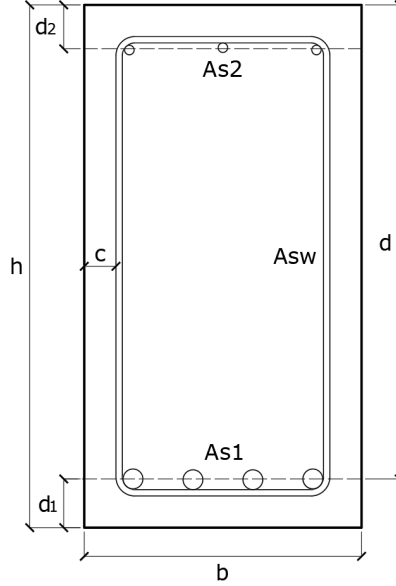
Beton bestaat uit cement, aggregaat (zand en grind), water en toevoegingsmiddelen. Wanneer cement met water reageert, ontstaat een chemisch proces dat hydratatie heet, waardoor het mengsel uithardt en het aggregaat stevig in het verharde cement wordt ingesloten. De sterkte van beton wordt beïnvloed door de samenstelling van deze componenten. Verschillende mengverhoudingen resulteren in verschillende sterkteklassen van beton, aangeduid met CXX/YY. In deze aanduiding representeert XX de karakteristieke druksterkte op een cilinder en YY de karakteristieke druksterkte op een kubus⁴, bijvoorbeeld C25/30 [30]. De sterkte- en vervormingseigenschappen van alle sterkteklassen van beton zijn terug te vinden in Tabel 3.1 van NBN EN 1992-1-1 [1].

Beton is een materiaal dat door het aggregaat en het verharde cement een hoge weerstand tegen drukkrachten heeft. Hoewel beton uitstekend bestand is tegen druk, is het relatief zwak in trek en bijzonder bros. Om deze zwakte te compenseren en het plotseling falen van een betonnen balk te voorkomen, worden stalen staven in de trekzone van het beton aangebracht. Deze stalen staven, ook wel wapening genoemd, absorberen de trekkrachten en verhogen daardoor de treksterkte van het beton.

⁴ De afmetingen van de cilinder en de kubus die gebruikt worden voor het bepalen van de druksterkte zijn gestandaardiseerd en te vinden in EN 206-1.

Grootheden doorsnede

In een betonnen balk bevinden zich langswapening en dwarskrachtwapening. De langswapening bestaat uit wapeningsstaven die in de lengterichting van de balk worden geplaatst, zowel aan de onder- als bovenkant van de doorsnede. De dwarskrachtwapening bestaat uit beugels die met een bepaalde tussenafstand van elkaar worden geplaatst. Figuur 3.9 toont een schets van de verschillende soorten wapening in een betonnen balk.



Figuur 3.9: Schets wapening doorsnede

Zoals te zien is in Figuur 3.9, wordt de dwarskrachtwapening aangeduid met A_{sw} . Deze beugels bevinden zich aan de buitenkant van de langswapening. De langswapening wordt aangeduid als A_{s1} of A_{s2} . De langswapening A_{s1} bevindt zich in de trekzone en wordt ook wel de primaire wapening genoemd, deze is onderaan in Figuur 3.9 te zien. De langswapening A_{s2} wordt de secundaire wapening genoemd en bevindt zich bovenaan in de figuur. Het verschil tussen primaire en secundaire langswapening wordt verder uitgelegd in Sectie 3.2.3.

De betondekking, aangeduid met de letter c , is de afstand tussen het oppervlak van de wapening en het dichtstbijzijnde betonoppervlak. De betondekking beschermt de wapening tegen verschillende aantastingen zoals bijvoorbeeld corrosie. De nuttige hoogte, aangeduid met de letter d , is gelijk aan de hoogte h verminderd met de afstand d_1 . De afstanden d_1 en d_2 worden berekend met formule (3.20) [31].

$$d_x = \frac{\sum s_i A_i}{\sum A_i} \quad (3.20)$$

waarbij:

- A_i de hoeveelheid langswapening in één laag is;
- s_i het zwaartepunt van de langswapening in één laag is.

Het zwaartepunt van een laag langswapening wordt berekend met formule (3.21):

$$s_i = c + \phi_b + \frac{\phi_s}{2} \quad (3.21)$$

waarbij:

- c de betondekking is;
- ϕ_b de diameter van de beugel is;
- ϕ_s de diameter van de wapeningsstaaf is.

Uiterste grenstoestand vs. bruikbaarheidsgrenstoestand

Bij het dimensioneren van betonbalken worden diverse berekeningen en controles uitgevoerd zowel in de uiterste grenstoestand als in de bruikbaarheidsgrenstoestand. De bepaling van de langs- en dwarskrachtwapening op basis van de sterkte vindt plaats in de UGT. In de BGT worden daarentegen controles uitgevoerd om de functionaliteit, duurzaamheid en het uiterlijk van de betonbalk te waarborgen. Dit omvat onder andere de controle van de spanning in het beton en de wapening, de scheurwijdte en de doorbuiging.

In deze sectie worden de principes en aannames voor de berekeningen in zowel de uiterste grenstoestand als de bruikbaarheidsgrenstoestand uitgelegd, zonder in te gaan op specifieke berekeningen of controles.

Bij de berekeningen voor de uiterste grenstoestand wordt aangenomen dat een vlakke doorsnede voor belasting ook vlak blijft na belasting. Verder wordt verondersteld dat er een perfecte aanhechting is tussen de wapening en het beton en dat de trekspanning in het beton verwaarloosd kan worden. Specifieke spanning-rek diagrammen voor beton en staal worden gebruikt in deze toestand. Daarnaast moeten de karakteristieke waarden van de betondruksterkte en staaltreksterkte worden omgezet in rekenwaarden. Formule (3.22) geeft de rekenwaarde van de betondruksterkte f_{cd} en formule (3.23) geeft de rekenwaarde van de staaltreksterkte f_{yd} weer [1][31].

$$f_{cd} = \alpha_{cc} \times \frac{f_{ck}}{\gamma_c} \quad (3.22)$$

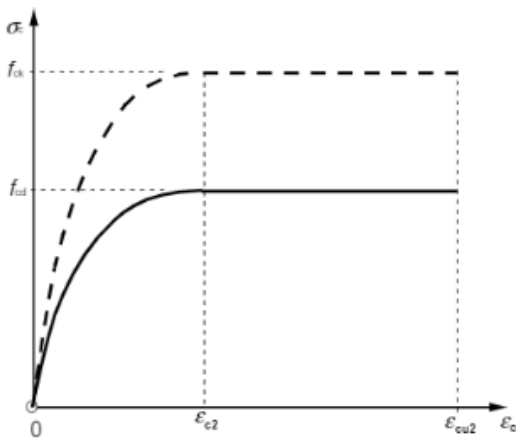
$$f_{yd} = \frac{f_{yk}}{\gamma_s} \quad (3.23)$$

waarbij:

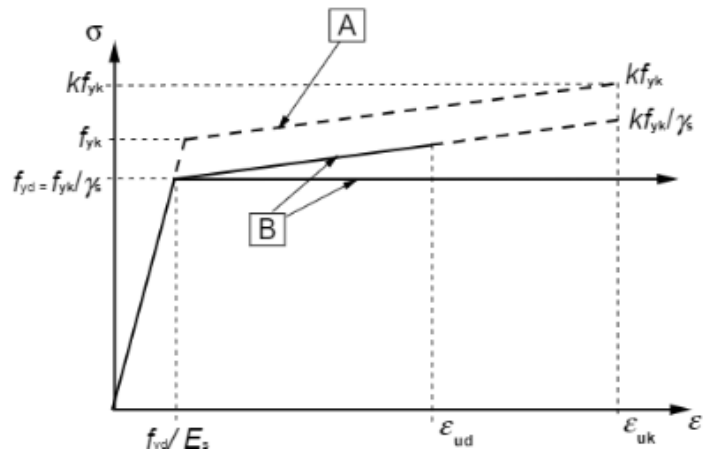
- α_{cc} een factor is die rekening houdt met de manier waarop de belasting werkt en de mogelijke langetermijneffecten op de druksterkte;
- f_{ck} de karakteristieke druksterkte van het beton is;
- γ_c de partiële factor voor beton is;
- f_{yk} de karakteristieke treksterkte van het staal is;
- γ_s de partiële factor voor staal is.

De factor α_{cc} is vastgelegd in de Nationale Bijlage van Eurocode 2 en bedraagt 0,85 voor berekeningen met betrekking tot buiging en 1,00 voor alle andere berekeningen [32]. De partiële factor voor beton γ_c bedraagt 1,5 en de partiële factor voor staal γ_s bedraagt 1,15 [1].

Zoals eerder gezegd zijn er specifieke spanning-rek diagrammen voor beton en staal die gebruikt worden voor de berekeningen in UGT. De spanning-rek diagram voor beton wordt getoond in Figuur 3.10 en de spanning-rek diagram voor betonstaal wordt getoond in Figuur 3.11.



Figuur 3.10: Parabool-rechthoekdiagram voor beton onder druk [1, p. 39]



Figuur 3.11: Bi-lineair spanning-rekdiagram voor betonstaal [1, p. 46]

In Figuur 3.10 worden twee spanning-rek relaties weergegeven. Het verschil tussen beide lijnen ligt in de maximale waarde van de spanning. Bij de stippellijn is de maximale spanning in het beton gelijk aan de karakteristieke druksterkte, terwijl bij de doorgetrokken lijn de maximale spanning gelijk is aan de rekenwaarde van de druksterkte. In deze masterproef wordt het spanning-rek diagram van de doorgetrokken lijn gehanteerd.

Zoals zichtbaar in Figuur 3.10, volgt de spanning aanvankelijk een paraboolvormig verloop totdat de rek van het beton ε_c een waarde van ε_{c2} bereikt. Wanneer de rek groter is dan ε_{c2} , blijft de spanning constant gelijk aan de rekenwaarde van de druksterkte van het beton, tot de maximale rek ε_{cu2} wordt bereikt. Volgens Eurocode 2 is ε_{c2} gelijk aan 2 promille en ε_{cu2} gelijk aan 3,5 promille voor beton met sterkteklassen tot en met C50/60. Als de rek in het beton kleiner is dan 2 promille, wordt de drukspanning in het beton berekend met formule (3.24) [1].

$$\sigma_c = f_{cd} \left[1 - \left(1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^2 \right] \quad (3.24)$$

Figuur 3.11 toont drie spanning-rek relaties, waarvan in deze masterproef de onderste wordt aangenomen. In deze relatie is de spanning in het betonstaal gelijk aan het product van de rek van het betonstaal ε_s en de elasticiteitsmodulus van het betonstaal E_s tot deze de rekenwaarde van de treksterkte bereikt. Daarna blijft de spanning in de wapening constant en gelijk aan de rekenwaarde van de treksterkte van het betonstaal. Dit overgangspunt van de rek wordt berekend met de volgende formule [1]:

$$\varepsilon_{s1} = \frac{f_{yd}}{E_s} = \frac{435 \text{ MPa}}{200000 \text{ MPa}} = 2,17\text{‰}$$

De rek van staal kent, net als die van beton, een maximum. In deze masterproef wordt deze maximale rek van het staal, ε_{ud} , vastgesteld op 10 promille.

Bij de berekeningen voor de bruikbaarheidsgrenstoestanden wordt aangenomen dat een vlakke doorsnede vóór belasting ook vlak blijft na belasting en dat er een perfecte hechting is tussen de wapening en het beton. In tegenstelling tot bij UGT wordt bij BGT aangenomen dat beton en staal perfect elastisch zijn en dus niet meer de spanning-rek diagrammen volgen zoals bij UGT. Dit betekent dat de wet van Hooke wordt gebruikt om het verband tussen spanning en rek te beschrijven ($\sigma = E \times \varepsilon$).

Daarnaast wordt er onderscheid gemaakt tussen een gescheurde en een niet-gescheurde doorsnede. Bij een niet-gescheurde doorsnede wordt aangenomen dat beton trekspanningen kan opnemen. Indien de doorsnede gescheurd is, wordt aangenomen dat beton, net zoals bij UGT, geen trekspanningen kan opnemen. Dit verschil heeft grote invloed op de betondrukhoogte x en het traagheidsmoment I van de doorsnede. De betondrukhoogte is de hoogte van de betonddoorsnede waarin drukspanningen optreden.

Om te bepalen of een doorsnede gescheurd is of niet, wordt een zogenaamd kritisch moment M_{cr} berekend. Dit kritisch moment wordt berekend op basis van een niet-gescheurde doorsnede. Indien het optredende moment in de doorsnede groter of gelijk is aan dit kritisch moment, moeten de daaropvolgende berekeningen gebruik maken van de betondrukhoogte en het traagheidsmoment van een gescheurde doorsnede. Indien het optredende moment kleiner is, worden deze grootheden voor een niet-gescheurde doorsnede gebruikt.

De berekeningen voor de betondrukhoogte en het traagheidsmoment van een niet-gescheurde doorsnede worden gegeven in respectievelijk formules (3.25) en (3.26).

$$x_{\text{niet gescheurd}} = \frac{\frac{b \times h^2}{2} + \alpha \times (A_{s1} \times d + A_{s2} \times d_2)}{A_{s1} \times \alpha + A_{s2} \times \alpha + b \times h} \quad (3.25)$$

$$I_{\text{niet gescheurd}} = \frac{b \times x^3}{3} + \frac{b \times (h - x)^3}{3} + \alpha \times A_{s1} \times (d - x)^2 + \alpha \times A_{s2} \times (x - d_2)^2 \quad (3.26)$$

waarin:

- b de breedte van de doorsnede is;
- h de hoogte van de doorsnede is;
- α de verhouding van de elasticiteitsmodulus van staal op die van beton is;
- A_{s1} de oppervlakte van de primaire wapening is;
- A_{s2} de oppervlakte van de secundaire wapening is;
- d de nuttige hoogte is;
- d_2 de afstand tot het zwaartepunt van de secundaire wapening is.

Op basis van de nuttige hoogte en het traagheidsmoment van de niet-gescheurde doorsnede kan het kritisch moment worden berekend met formule (3.27).

$$M_{cr} = \frac{I_{\text{niet gescheurd}}}{h - x_{\text{niet gescheurd}}} \times f_{ctm} \quad (3.27)$$

In formule (3.27) is f_{ctm} de gemiddelde treksterkte van het beton volgens Tabel 3.1 van NBN EN 1992-1-1 [1].

Formules (3.28) en (3.29) geven respectievelijk de formule voor de nuttige hoogte en het traagheidsmoment van een gescheurde doorsnede weer.

$$x_{\text{gescheurd}} = \frac{-\alpha \times (A_{s1} + A_{s2})}{b} + \sqrt{\left(\frac{\alpha \times (A_{s1} + A_{s2})}{b}\right)^2 + \frac{2 \times \alpha \times (A_{s1} \times d + A_{s2} \times d_2)}{b}} \quad (3.28)$$

$$I_{\text{gescheurd}} = \frac{b \times x^3}{3} + \alpha \times A_{s1} \times (d - x)^2 + \alpha \times A_{s2} \times (x - d_2)^2 \quad (3.29)$$

Duurzaamheid

Betonnen balken moeten voldoen aan duurzaamheidscriteria. Gedurende hun ontwerplevensduur moeten ze aan de eisen van bruikbaarheid, sterkte en stabiliteit voldoen, zonder significant verlies aan functionaliteit en zonder buitengewoon onderhoud. De duurzaamheid van een betonnen balk komt in het gedrang wanneer de wapeningsstaven corroderen. Om dit te voorkomen, moet er een voldoende dikke betondekking worden voorzien. Daarnaast moet de scheurwijdte beperkt worden gehouden en kan het nodig zijn om een beton met een hogere sterkteklasse te kiezen [1]. De controle op scheurwijdte wordt verder uitgelegd in Sectie 3.2.6. De berekening van de benodigde betondekking en de vereiste sterkteklasse worden hieronder toegelicht.

Volgens Eurocode 2 moet rekening worden gehouden met milieuklassen, die de mogelijke blootstelling aan chemische en fysische omstandigheden omvatten. In totaal zijn er achttien milieuklassen, die onderling zijn onderverdeeld op basis van de oorzaak van aantasting, zoals bijvoorbeeld door chloriden of carbonatatie. Deze milieuklassen worden beschreven in Tabel 4.1 van Eurocode 2 [1] en specifiek voor België in Tabel 1 van NBN B 15-001 [33].

De Belgische nationale aanvulling van de norm voor beton (NBN B 15-001) maakt gebruik van betontypes. Een betontype is een combinatie van de maximale water-cementfactor, het minimale cementgehalte, de minimale druksterkteklasse en eventueel het minimale luchtgehalte. Een betontype wordt aangeduid met de maximaal toelaatbare water-cementfactor voorafgegaan door de letter T en eventueel gevolgd door de letter A indien er een opgelegd minimum luchtgehalte is⁵. Tabel 3.9 geeft de verschillende betontypes weer [33].

Tabel 3.9: Betontypes [33, p. 46]

Aanduiding	T(1,50)	T(1,00)	T(0,65)	T(0,60)	T(0,55)	T(0,50)	T(0,45)
Maximale water-cementfactor	1,50	1,00	0,65	0,60	0,55	0,50	0,45
Minimaal cementgehalte	-	-	260	280	300	320	340
Minimale druksterkteklasse	C8/10	C12/15	C16/20	C20/25	C25/30	C30/37	C35/45

⁵ Betontypes met een opgelegd minimum luchtgehalte worden niet beschouwd in deze masterproef omdat de minimale betonsterkte van deze betontypes steeds lager is dan het corresponderende betontype zonder een minimum luchtgehalte (zie Tabel F.2-ANB van NBN B 15-001).

Deze nationale bijlage bevat ook duurzaamheidseisen waarin deze betontypes gekoppeld zijn aan milieuklassen. De minimale betonsterkteklasse van een betonnen balk is gelijk aan de minimale betonsterkteklasse van het bijhorende betontype van een milieuklasse. In deze masterproef wordt geen rekening gehouden met uitzonderingen op deze duurzaamheidseisen zoals gespecificeerd in Tabel F.2-ANB van NBN B 15-001. De duurzaamheidseisen worden getoond in Tabel 3.10 [33].

Tabel 3.10: Duurzaamheidseisen

Milieuklasse	X0	XC1	XC2	XC3	XC4	XD1	XD2	XD3	XS1
Betontype	T(1,00)	T(0,65)	T(0,60)	T(0,55)	T(0,50)	T(0,50)	T(0,50)	T(0,45)	T(0,50)
Milieuklasse	XS2	XS3	XF1	XF2	XF3	XF4	XA1	XA2	XA3
Betontype	T(0,45)	T(0,45)	T(0,55)	T(0,50)	T(0,50)	T(0,45)	T(0,55)	T(0,50)	T(0,45)

De betondekking wordt berekend door een minimale betondekking $c_{\min}$ te vermeerderen met een uitvoeringstolerantie Δc_{dev} . De formule voor de betondekking wordt getoond in formule (3.30) [1].

$$c = c_{\min} + \Delta c_{\text{dev}} \quad (3.30)$$

De minimale betondekking wordt berekend met formule (3.31).

$$c_{\min} = \max(c_{\min,b}; c_{\min,\text{dur}}; 10 \text{ mm}). \quad (3.31)$$

waarin:

- $c_{\min,b}$ de minimumdekking op basis van de aanhechtingseisen is;
- $c_{\min,\text{dur}}$ de minimumdekking op basis van de milieu-omstandigheden is.

De minimumdekking op basis van de aanhechtingseisen is gelijk aan de diameter van de wapeningsstaaf met de grootste diameter, mits de nominale maximale korrelafmeting niet groter is dan 32 mm. Als de korrelafmeting wel groter is, moet deze waarde met 5 mm worden verhoogd. De minimumdekking op basis van de milieuomstandigheden wordt volgens Eurocode 2 bepaald door de constructieklasse en de milieuklasse. In deze masterproef wordt uitgegaan van constructieklasse S4 (ontwerplevensduur 50 jaar). De constructieklasse wordt niet aangepast, zoals uitgelegd in Tabel 4.3N van NBN EN 1992-1-1. De waarden voor $c_{\min,\text{dur}}$ in functie van de milieuklasse voor constructieklasse S4 zijn weergegeven in Tabel 3.11 [1].

Tabel 3.11: Waarden voor $c_{\min,\text{dur}}$ voor constructieklasse S4

Milieuklasse	X0	XC1	XC2 / XC3	XC4	XD1 / XS1	XD2 / XS2	XD3 / XS3
$c_{\min,\text{dur}}$ [mm]	10	15	25	30	35	40	45

De uitvoeringstolerantie is standaard tien millimeter. Indien bij de uitvoering een kwaliteitsborgingssysteem wordt gehanteerd waarin de betondekking wordt gemeten, bedraagt de uitvoeringstolerantie slechts vijf millimeter.

Kruip

Beton is een materiaal dat gevoelig is voor kruip. Kruip is de extra vervorming die optreedt bij langdurige belasting van het materiaal. Deze extra vervorming zorgt ook voor hogere spanningen

in het beton waardoor de kans op scheuren toeneemt. Omdat kruip een langetermijneffect is, moet dit effect in rekening worden gebracht bij de quasi-blijvende combinatie voor de bruikbaarheidsgrenstoestand (BGT-Q). Dit gebeurt door de elasticiteitsmodulus van beton E_{cm} voor deze combinatie te verlagen met een kruipcoëfficiënt $\varphi_{(t,t_0)}$. Formule (3.32) laat zien hoe deze verlaagde elasticiteitsmodulus E_c wordt berekend [34].

$$E_c = \frac{E_{cm}}{1 + \varphi_{(t,t_0)}} \quad (3.32)$$

De kruipcoëfficiënt wordt berekend met behulp van de formules in bijlage B van NBN EN 1992-1-1. Bij deze berekening wordt aangenomen dat de ouderdom van het beton op het beschouwde tijdstip t gelijk is aan de ouderdom van het beton bij belasten t_0 of met andere woorden wordt de theoretische kruipcoëfficiënt φ_0 berekend. De formule van deze theoretische kruipcoëfficiënt wordt weergegeven in formule (3.33) [1].

$$\varphi_0 = \varphi_{RH} \times \beta(f_{cm}) \times \beta(t_0) \quad (3.33)$$

waarin:

- φ_{RH} een factor is die rekening houdt met de relatieve vochtigheid;
- $\beta(f_{cm})$ een factor is die rekening houdt met het effect van de betonsterkte;
- $\beta(t_0)$ een factor is die rekening houdt met de ouderdom van het beton op het tijdstip van belasten.

De factor φ_{RH} is afhankelijk van de gemiddelde druksterkte van het beton op 28 dagen f_{cm} . De berekening gebeurt met formule (3.34) als de druksterkte kleiner of gelijk is aan 35 MPa, en anders met formule (3.35) [1].

$$\varphi_{RH} = 1 + \frac{1 - RH/100}{0,1 \times \sqrt[3]{h_0}} \quad (3.34)$$

$$\varphi_{RH} = \left[1 + \frac{1 - RH/100}{0,1 \times \sqrt[3]{h_0}} \times a_1 \right] \times a_2 \quad (3.35)$$

waarin:

- RH de relatieve vochtigheidsgraad van de omgeving in procent is;
- h_0 de theoretische dikte van het element is;
- a_1 en a_2 coëfficiënten zijn waarmee de invloed van de betonsterkte in rekening wordt gebracht.

De theoretische dikte h_0 wordt berekend met formule (3.36) [1].

$$h_0 = \frac{2 \times A_c}{u} \quad (3.36)$$

waarin:

- A_c de oppervlakte van de betondoorsnede is;
- u de omtrek van de doorsnede dat in contact komt met de buitenlucht is.

De coëfficiënten a_1 en a_2 worden berekend met respectievelijk formule (3.37) en (3.38) [1].

$$a_1 = \left[\frac{35}{f_{cm}} \right]^{0,7} \quad (3.37)$$

$$a_2 = \left[\frac{35}{f_{cm}} \right]^{0,2} \quad (3.38)$$

Ten slotte worden de factoren voor de betonsterkte $\beta(f_{cm})$ en voor de ouderdom van het beton bij belasten $\beta(t_0)$ berekend met respectievelijk formule (3.39) en (3.40) [1].

$$\beta(f_{cm}) = \frac{16,8}{\sqrt{f_{cm}}} \quad (3.39)$$

$$\beta(t_0) = \frac{1}{(0,1 + t_0^{0,20})} \quad (3.40)$$

3.2.3 Langswapening

De langswapening in een betonnen balk bestaat uit wapeningsstaven die in de lengterichting van de balk worden geplaatst om de trekkrachten in deze richting op te vangen. Deze trekkrachten ontstaan door het optredende moment. De langswapening wordt zowel aan de onder- als bovenkant van de doorsnede aangebracht, waarbij de hoeveelheid wapening varieert afhankelijk van het teken en de grootte van het moment.

De meeste wapening bevindt zich altijd in de trekzone van de balk. Volgens de in deze masterproef gehanteerde tekencoventie is de trekzone bij een positief moment onderaan en bij een negatief moment bovenaan. De wapening in de trekzone wordt de primaire wapening genoemd, terwijl de wapening in de drukzone de secundaire wapening is. Deze worden respectievelijk aangeduid met de cijfers 1 en 2.

Minimum langswapening

Zowel de hoeveelheid primaire als secundaire wapening moet volgens Eurocode 2 groter zijn dan een minimum hoeveelheid. Deze minimum hoeveelheid wapening wordt bepaald door het maximum van de volgende drie waarden: de absolute minimum hoeveelheid langswapening, de minimum hoeveelheid langswapening om een brosse breuk te voorkomen, en de minimum hoeveelheid langswapening om scheuren in het beton te beperken.

De absolute minimum hoeveelheid langswapening is afhankelijk van de doorsnede van de balk en wordt berekend met de volgende formule [1]:

$$A_{s,min,abs} = 0,0013 \times b \times d \quad (3.41)$$

waarbij:

- b de breedte van de doorsnede is;
- d de nuttige hoogte van de doorsnede is.

De minimum hoeveelheid langswapening om een brosse breuk te vermijden wordt berekend met de formule (3.42) [1].

$$A_{s,\min,bros} = 0,26 \times \frac{f_{ctm}}{f_{yk}} \times b \times d \quad (3.42)$$

waarbij:

- f_{ctm} de gemiddelde treksterkte van beton is;
- f_{yk} de treksterkte van de wapening is.

De minimum hoeveelheid langswapening voor scheurbeheersing wordt berekend met de volgende formule [1]:

$$A_{s,\min,scheurbeheersing} = \frac{k_c \times k \times f_{ct,eff} \times A_{ct}}{\sigma_s} \quad (3.43)$$

waarbij:

- k_c een coëfficiënt is die rekening houdt met de spanningsverdeling in de doorsnede onmiddellijk voor het scheuren en met de verandering van de hefboomsarm;
- k een coëfficiënt is die rekening houdt met het effect van niet-gelijkmatige eigenspanningen, die leiden tot een reductie van de krachten ten gevolge van verhinderde vervormingen;
- $f_{ct,eff}$ de gemiddelde treksterkte van beton is op het tijdstip waarop voor het eerst scheuren kunnen worden verwacht;
- A_{ct} de oppervlakte van het beton in de trekzone is;
- σ_s de absolute waarde van de maximaal toelaatbare spanning in de wapening onmiddellijk na het scheuren is.

De coëfficiënt k_c is altijd gelijk aan 0,4 omdat er geen normaalkrachten in de balk kunnen optreden. De gemiddelde treksterkte van het beton op het tijdstip waarop scheuren voor het eerst kunnen worden verwacht, is gelijk aan de waarde voor de gemiddelde treksterkte van het beton uit Tabel 3.1 van NBN EN 1992-1-1. De oppervlakte A_{ct} is gelijk aan de halve oppervlakte van de volledige doorsnede, en de maximaal toelaatbare spanning σ_s is gelijk aan de treksterkte van de wapening. Voor deze maximale spanning wordt geen gereduceerde waarde gebruikt om te voldoen aan de grens van de scheurwijdte, omdat de maximale scheurwijdte volledig wordt berekend zoals uitgelegd in Sectie 3.2.6. De coëfficiënt k wordt berekend met formule (3.44) [1].

$$k = 1,21 - 0,7 \times \frac{h}{1000} \quad (3.44)$$

De coëfficiënt k is enkel in functie van de hoogte van de doorsnede. De waarde van deze coëfficiënt heeft echter een maximum van 1 (hoogte kleiner of gelijk 300 mm) en een minimum van 0,65 (hoogte groter of gelijk aan 800 mm).

Benodigde langswapening

De benodigde langswapening wordt berekend voor de uiterste grenstoestand. Dit betekent dat bij deze berekeningen de rekenwaarden van de betondruksterkte en de staaltreksterkte moeten worden gebruikt. Om de benodigde hoeveelheid langswapening te bepalen, moet eerst een gereduceerd moment μ_d worden berekend. De formule voor dit gereduceerde moment wordt gegeven door formule (3.45) [31].

$$\mu_d = \frac{M_d}{b \times d^2 \times f_{cd}} \quad (3.45)$$

waarbij:

- M_d de rekenwaarde van het optredend moment in UGT-B is;
- b de breedte van de doorsnede is;
- d de nuttige hoogte is;
- f_{cd} de rekenwaarde van de druksterkte van het beton is.

Dit gereduceerde moment wordt gebruikt om de spanning in de wapening σ_s te bepalen en om de verhouding tussen de hefboomsarm van de druk- en trekzone in een doorsnede en de nuttige hoogte te vinden. Deze verhouding wordt aangeduid met de Griekse letter ζ . De factor ζ en de spanning σ_s worden gevonden met behulp van Tabel 4.2.3.7a. In deze tabel staan voor verschillende waarden van het gereduceerde moment de bijbehorende waarde van de factor ζ en de waarde van de spanning σ_s . Deze tabel is opgenomen in bijlage D. Als de exacte waarde van het gereduceerde moment niet in de tabel voorkomt, moet er geïnterpoleerd worden tussen de eerstvolgende grotere en kleinere waarde. De benodigde hoeveelheid langswapening kan vervolgens worden berekend met formule (3.46) [31].

$$A_s = \frac{M_d}{\zeta \times d \times \sigma_s} \quad (3.46)$$

waarbij:

- M_d de rekenwaarde van het optredend moment in UGT-B is;
- ζ verhouding hefboomsarm in doorsnede en de nuttige hoogte;
- d de nuttige hoogte is;
- σ_s de spanning in de wapening is.

Detallering langswapening

Volgens Eurocode 2 moeten de langswapeningsstaven een minimale verankeringslengte l_{bd} hebben. Deze lengte is essentieel om de trekkrachten in de wapening veilig over te dragen aan het

beton. De benodigde verankeringslengte hangt af van verschillende factoren zoals de vorm van de wapeningsstaven, de betondekking, de opsluiting van de langswapening door de dwarskrachtwapening, de invloed van gelaste dwarsstaven en de drukspanning loodrecht op het verankeringsvlak [1]. In deze masterproef wordt een formule gebruikt die rekening houdt met de meest voorkomende waarden van deze parameters. Deze berekening wordt weergegeven in volgende formule [35]:

$$l_{bd} = \frac{345 \times \phi}{f_{ck}^{2/3}} \quad (3.47)$$

waarbij:

- ϕ de diameter van de wapeningsstaaf is;
- f_{ck} de karakteristieke druksterkte van het beton is.

Als er aan het uiteinde van de balk niet genoeg lengte beschikbaar is voor de verankering, mag de wapeningsstaaf verticaal omgebogen worden. De diameter van deze ombuiging moet voldoende groot zijn om beschadiging van de wapeningsstaaf te voorkomen. De minimale diameter van de ombuiging moet viermaal de diameter van de staaf zijn bij een staafdiameter van maximaal zestien millimeter. Bij een grotere staafdiameter moet de ombuigingsdiameter minimaal zevenmaal de diameter van de staaf zijn [1].

Verder moet gecontroleerd worden of de gekozen wapening in één laag in de doorsnede kan worden geplaatst. Deze controle wordt uitgevoerd met behulp van formule (3.48):

$$b \geq 2 \times c + 2 \times \phi_b + n \times \phi_s + 40 \times (n - 1) \quad (3.48)$$

waarbij:

- b de breedte van de doorsnede is;
- c de betondekking is;
- ϕ_b de diameter van de beugel is;
- ϕ_s de diameter van de wapeningsstaaf is;
- n het aantal wapeningsstaven is.

De waarde 40 in formule (3.48) is de minimale vrije ruimte tussen de wapeningsstaven. Dit is een praktische waarde die in deze masterproef wordt gehanteerd. Volgens Eurocode 2 is deze vrije ruimte afhankelijk van de maximale korreldiameter [1] [35].

3.2.4 Dwarskrachtwapening

De dwarskrachtwapening heeft als doel om de dwarskrachten op te vangen, zoals de naam al suggereert. Deze wapening bestaat uit beugels die op een bepaalde afstand van elkaar worden geplaatst. In deze masterproef wordt aangenomen dat er verticaal rechtopstaande beugels met twee benen worden gebruikt. De meeste dwarskrachtwapening is nodig nabij steunpunten en puntlasten. De beugels voor dwarskrachtwapening hebben doorgaans een diameter van zes, acht

of tien millimeter. De hoeveelheid geplaatste dwarskrachtwapening hangt af van de tussenafstand: hoe kleiner de tussenafstand, hoe groter de hoeveelheid geplaatste dwarskrachtwapening. De hoeveelheid dwarskrachtwapening wordt uitgedrukt als een oppervlakte wapening per eenheid afstand.

De berekening van de dwarskrachtwapening begint met het bepalen van de minimale hoeveelheid dwarskrachtwapening die, volgens Eurocode 2, over de volledige lengte van de balk aanwezig moet zijn. Hoewel de betonnen balk zelf enige weerstand tegen dwarskrachten biedt, wordt deze intrinsieke weerstand genegeerd omdat overal minimaal de voorgeschreven hoeveelheid dwarskrachtwapening moet worden geplaatst, ongeacht of de balk zelf de belasting kan dragen. Als de minimum dwarskrachtwapening niet voldoende is over de hele lengte van de balk, wordt de benodigde hoeveelheid dwarskrachtwapening berekend. Tot slot wordt gecontroleerd of de weerstand tegen dwarskrachten, zoals geleverd door de geplaatste wapening, niet groter is dan de maximaal toelaatbare weerstand [36].

Minimum dwarskrachtwapening

De formule voor het berekenen van de minimale hoeveelheid dwarskrachtwapening wordt weergegeven in formule (3.49) [1][36].

$$\left(\frac{A_{sw}}{s}\right)_{\min} = \frac{0,08 \times \sqrt{f_{ck}}}{f_{yk}} \times b \quad (3.49)$$

waarin:

- f_{ck} de karakteristieke druksterkte van het beton is;
- f_{yk} de karakteristieke treksterkte van de wapening is;
- b de breedte van de doorsnede is;

Deze formule is afgeleid van de minimale dwarskrachtwapeningsverhouding $\rho_{w,\min}$ waaraan een balk moet voldoen volgens Eurocode 2. Deze minimale verhouding wordt weergegeven in formule (3.50). De dwarskrachtwapeningsverhouding van de geplaatste wapening wordt berekend met formule (3.51). Formule (3.49) wordt verkregen door formule (3.50) in formule (3.51) in te vullen en de hoeveelheid dwarskracht per afstand af te zonderen [1].

$$\rho_{w,\min} = \frac{0,08 \times \sqrt{f_{ck}}}{f_{yk}} \quad (3.50)$$

$$\rho_w = \frac{A_{sw}}{s \times b} \quad (3.51)$$

Om de tussenafstand s tussen de beugels te berekenen, wordt eerst een diameter van de beugel aangenomen. Vervolgens wordt formule (3.52) gebruikt om de tussenafstand te bepalen.

$$s = \frac{\frac{\pi \times \phi_b^2}{4} \times n}{\frac{A_{sw}}{s}} \quad (3.52)$$

waarin:

- ϕ_b de diameter van de beugels is;
- n het aantal benen van de beugels is;

De tussenafstanden worden altijd zo gekozen dat ze deelbaar zijn door vijf. Indien de berekende tussenafstand niet aan deze eis voldoet, wordt afgerond naar het dichtstbijzijnde lagere getal.

De rekenwaarde van de weerstand tegen dwarskracht $V_{Rd,s}$ van de berekende dwarskrachtwapening wordt berekend met formule (3.53).

$$V_{Rd,s} = \frac{\pi \times \phi_b^2}{4} \times \frac{n}{s} \times 0,9 \times d \times f_{ywd} \times \cot \theta \quad (3.53)$$

waarin:

- d de nuttige hoogte is;
- f_{ywd} de rekenwaarde van de treksterkte van de wapening is;
- θ de schoorhoek is.

De schoorhoek moet volgens de Nationale bijlage van Eurocode 2 altijd groter zijn dan $26,6^\circ$ en kleiner dan 45° [32]. In deze masterproef wordt steeds aangenomen dat de schoorhoek gelijk is aan $26,6^\circ$.

Benodigde dwarskrachtwapening

Indien de weerstand van de minimum dwarskrachtwapening niet voldoet, moet de benodigde dwarskrachtwapening worden geplaatst. De rekenwaarde van de optredende dwarskracht waarvoor de wapening wordt berekend, is de maximale dwarskracht in een interval waarin de optredende dwarskracht de weerstand overschrijdt. Bij opleggingen hoeft de dwarskracht binnen een afstand gelijk aan de nuttige hoogte niet gecontroleerd te worden. De benodigde dwarskrachtwapening wordt berekend met formule (3.54).

$$\frac{A_{sw}}{s} = \frac{V_{Ed}}{0,9 \times d \times f_{ywd} \times \cot \theta} \quad (3.54)$$

De tussenafstand van de beugels wordt op dezelfde manier berekend als bij de minimum dwarskrachtwapening, door gebruik te maken van formule (3.52). Daarna wordt de weerstand van de berekende benodigde dwarskrachtwapening berekend met formule (3.53). Ten slotte moet worden gecontroleerd dat deze weerstand kleiner is dan de maximale weerstand $V_{Rd,max}$. Formule (3.55) geeft de formule voor de maximale weerstand weer [1].

$$V_{Rd,max} = \frac{\alpha_{cw} \times b \times 0,9 \times d \times \nu_1 \times f_{cd}}{\cot \theta + \tan \theta} \quad (3.55)$$

De factor α_{cw} is gelijk aan één voor niet-voorgespannen elementen. Daarnaast is de factor ν_1 gelijk aan 0,6 voor betonsterktes met een karakteristieke druksterkte lager dan 60 MPa.

Detaillering dwarskrachtwapening

Eurocode 2 bevat een maximale hart-op-hart afstand in de langsrichting van opeenvolgende beugels. De maximale hart-op-hart afstand is gelijk aan het minimum van 0,75 keer de nuttige hoogte van de balk of 600 millimeter [1].

Net zoals bij de langswapening moeten de ombuigingen van de beugels groter zijn dan een minimale buigdiameter. Hierbij gelden dezelfde regels als bij de langswapening (zie Sectie 3.2.3).

3.2.5 Spanningen

De spanningen in zowel het beton als de wapening worden berekend voor zowel de uiterste grenstoestand als de bruikbaarheidsgrenstoestand. De berekeningsmethoden voor deze spanningen verschillen echter per grenstoestand en dienen verschillende doelen.

Uiterste grenstoestand

De spanningen in de uiterste grenstoestand worden bepaald aan de hand van de spanning-rek diagrammen zoals beschreven in Sectie 3.2.2. De rekken in zowel het beton als de wapening worden, net zoals de factor ζ voor de berekening van langswapening, gevonden door het gereduceerde moment μ_d te berekenen. De bijbehorende rekken worden vervolgens verkregen door interpolatie in Tabel 4.2.3.7a (zie Sectie 3.2.3). De berekende spanningen in de UGT worden gebruikt als rekenwaarden: de spanning in het beton als de druksterkte van het beton en de spanning in het staal als de treksterkte van de wapening [27].

Bruikbaarheidsgrenstoestand

In de bruikbaarheidsgrenstoestand moeten de spanningen in zowel het beton als de wapening onder een bepaalde maximumwaarde blijven, afhankelijk van de sterkte van het beton of het betonstaal. Deze controle is noodzakelijk om scheuren te vermijden die de functionaliteit en duurzaamheid van de balk zouden kunnen aantasten. De maximale spanningswaarden voor verschillende combinaties in de bruikbaarheidsgrenstoestand zijn vastgelegd in de nationale bijlage van Eurocode 2 en worden weergegeven in Tabel 3.12 [1][32].

Tabel 3.12: Maximale spanningen in bruikbaarheidsgrenstoestand

	BGT-K	BGT-F	BGT-Q
Beton	$0,6 \times f_{ck}$	$0,5 \times f_{ck}^{(*)}$	$0,45 \times f_{ck}$
	$0,5 \times f_{ck}^{(*)}$		
Staal	$0,8 \times f_{yk}$		

Zoals te zien is in Tabel 3.12, is de maximale spanning in de wapening gelijk voor alle drie de combinaties in de bruikbaarheidsgrenstoestand. De maximale spanning in het beton is het strengst in de quasi-blijvende combinatie en het hoogst in de karakteristieke combinatie. De met een asterisk gemarkeerde maximale spanningen zijn alleen van toepassing als het betonelement zich in de milieuklassen XD, XS, en XF bevindt [1][32].

De spanning in het beton, de primaire wapening en de secundaire wapening wordt berekend met respectievelijk formules (3.56), (3.57), en (3.58).

$$\sigma_c = \frac{M_{Ed} \times y}{I} \quad (3.56)$$

$$\sigma_{s1} = \frac{d - x}{x} \times \sigma_c \times \alpha \quad (3.57)$$

$$\sigma_{s2} = \frac{x - d_2}{x} \times \sigma_c \times \alpha \quad (3.58)$$

waarbij:

- M_{Ed} de rekenwaarde van het optredend moment is;
- y de afstand is van het punt waar de spanning wordt berekend tot de neutrale lijn;
- I het traagheidsmoment van de doorsnede is;
- x de betondrukhoogte is;
- d de nuttige hoogte is;
- d_2 de afstand tot het zwaartepunt van de secundaire wapening is;
- α de verhouding van de elasticiteitsmodulus van staal op die van beton⁶ is.

In formules (3.56) - (3.58) wordt het traagheidsmoment en de betondrukhoogte van de doorsnede gebruikt. De gebruikte formules voor deze grootheden zijn afhankelijk van of de doorsnede gescheurd is of niet (zie Sectie 3.2.2).

3.2.6 Scheurwijdte

De scheurwijdte moet beperkt blijven om de functionaliteit en duurzaamheid van de constructie te waarborgen. Bovendien moet het uiterlijk van de balk acceptabel blijven. De maximale scheurwijdte w_{max} is vastgesteld op 0,3 mm voor alle milieuklassen, behalve voor X0 en XC1. Voor de milieuklassen X0 en XC1 geldt een maximale scheurwijdte van 0,4 mm als het uiterlijk acceptabel moet zijn. Als het uiterlijk geen rol speelt, is er geen limiet voor de scheurwijdte voor deze twee milieuklassen [1].

Voor het berekenen van de scheurwijdte moet eerst worden vastgesteld of de doorsnede daadwerkelijk gescheurd is door het optredend moment te vergelijken met het kritisch moment, zoals beschreven in Sectie 3.2.2. De berekening van de scheurwijdte vindt plaats voor BGT-Q, waarbij ook kruip moet worden meegenomen bij het bepalen van de elasticiteitsmodulus van beton (zie formule (3.32)).

Indien de doorsnede gescheurd is, wordt de scheurwijdte w_k berekend met formule (3.59).

$$w_k = S_{r,max} \times (\varepsilon_{sm} - \varepsilon_{cm}) \quad (3.59)$$

⁶ De elasticiteitsmodulus van beton is voor BGT-K en BGT-F gelijk aan de gemiddelde elasticiteitsmodulus van beton E_{cm} volgens Tabel 3.1 van NBN EN 1992-1-1. Bij BGT-Q wordt rekening gehouden met kruip en wordt de elasticiteitsmodulus van beton berekend met formule (3.32).

waarin:

- $S_{r,\max}$ de maximale scheurafstand is;
- ε_{sm} de gemiddelde rek van de wapening is;
- ε_{cm} de gemiddelde betonrek tussen de scheuren is.

De maximale scheurafstand wordt berekend met formule (3.60) [1].

$$S_{r,\max} = k_3 \times c + k_1 \times k_2 \times k_4 \times \frac{\phi}{\rho_{\text{eff}}} \quad (3.60)$$

waarin:

- k_1 een coëfficiënt is die rekening houdt met de aanhechteigenschappen van de wapening;
- k_2 een coëfficiënt is die rekening houdt met de rekverdeling;
- k_3 gelijk is aan 3,4 volgens NBN EN 1992-1-1 ANB [32];
- k_4 gelijk is aan 0,425 volgens NBN EN 1992-1-1 ANB [32];
- c de betondekking is;
- ϕ de staafdiameter van de primaire wapening is;
- ρ_{eff} de wapeningsverhouding is.

De coëfficiënt k_1 is 0,8 voor staven met goede aanhechting, en de coëfficiënt k_2 is 0,5 voor buiging. Als slechts één diameter van wapening in de doorsnede is gebruikt, is ϕ gelijk aan deze diameter. Indien er verschillende diameters in de doorsnede aanwezig zijn, moet een equivalente diameter ϕ_{eq} worden berekend met behulp van formule (3.61) [1].

$$\phi_{\text{eq}} = \frac{n_1 \times \phi_1^2 + n_2 \times \phi_2^2}{n_1 \times \phi_1 + n_2 \times \phi_2} \quad (3.61)$$

De wapeningsverdeling ρ_{eff} wordt berekend met formule (3.63).

$$\rho_{\text{eff}} = \frac{A_s}{A_{c,\text{eff}}} \quad (3.62)$$

met:

$$A_{c,\text{eff}} = b \times \min \left(\frac{h-x}{3}; 2,5 \times (h-d); \frac{h}{2} \right) \quad (3.63)$$

Indien de hart-op-hart afstand s tussen twee naastliggende staven groter is dan $5 \times (c + 2 \times \phi)$ is de maximale scheurafstand gelijk aan:

$$S_{r,\max} = 1,3 \times (h - x)$$

Het verschil tussen de gemiddelde rek van de wapening en de gemiddelde rek van het beton tussen de scheuren wordt berekend met formule (3.64) [1][35].

$$\varepsilon_{sm} - \varepsilon_{cm} = \frac{\sigma_s - k_t \times \frac{f_{ctm}}{\rho_{eff}} \times (1 + \alpha \times \rho_{eff})}{E_s} \geq 0,6 \times \frac{\sigma_s}{E_s} \quad (3.64)$$

waarin:

- σ_s de spanning in de trekwapening is;
- k_t een factor is die afhankelijk is van de belastingsduur. Deze factor is gelijk aan 0,6 voor een kortdurende belasting en gelijk aan 0,4 voor een langdurige belasting [1].
- f_{ctm} de gemiddelde treksterkte van het beton is;
- α de verhouding van de elasticiteitsmodulus van staal op die van beton is;
- E_s de elasticiteitsmodulus van staal is.

3.2.7 Doorbuiging

De doorbuiging van de balk moet worden gecontroleerd om zowel functionaliteit als het uiterlijk te waarborgen. Volgens Eurocode 2 mag de uiteindelijke doorbuiging van de balk onder de quasi-blijvende belastingen niet groter zijn dan de overspanning gedeeld door 250 [1].

De doorbuiging wordt berekend op lange termijn (BGT-Q), waarbij rekening wordt gehouden met kruip. Als de balk niet is gescheurd, is de doorbuiging gelijk aan de berekende doorbuiging van de balk. Indien er scheuren optreden, moet de uiteindelijke doorbuiging α worden berekend met behulp van formule (3.65) [1]:

$$\alpha = \zeta \times \alpha_{||} + (1 - \zeta) \times \alpha_{\perp} \quad (3.65)$$

waarin:

- ζ een verdelingsfactor is die rekening houdt met “tension stiffening” in een doorsnede;
- $\alpha_{||}$ de doorbuiging voor de gescheurde toestand is;
- $\alpha_{\perp}$ de doorbuiging voor de niet-gescheurde toestand is.

met:

$$\zeta = 1 - \beta \times \left(\frac{M_{cr}}{M_{Ed}} \right)^2 \quad (3.66)$$

waarin:

- β een coëfficiënt is die rekening houdt met de belastingsduur of herhaalde belasting op de gemiddelde rek. Deze is gelijk aan 1,0 voor kortdurende belastingen en 0,5 voor aanhoudende belastingen;
- M_{cr} het kritisch buigmoment is zoals besproken in Sectie 3.2.2.

3.3 Staal

In deze sectie worden de formules, ontwerpeisen en controles besproken die worden toegepast bij het dimensioneren van stalen balken volgens Eurocode 3 [2]. Allereerst worden de aannames en beperkingen uiteengezet die in deze masterproef gelden voor het ontwerp van stalen balken. Vervolgens wordt algemene informatie over het materiaal staal gegeven, noodzakelijk voor het begrijpen van de berekeningen. Daarna worden de formules, eisen en controles voor het ontwerp op basis van sterkte behandeld. Dit wordt gevolgd door een uitleg van de formules en controles tegen het kippen van de balk. Ten slotte wordt toegelicht hoe de doorbuiging van stalen balken wordt gecontroleerd.

3.3.1 Aannames en beperkingen

In het programma heeft de gebruiker alleen de mogelijkheid om te kiezen uit een aantal vooraf geprogrammeerde profielen. Specifieker zal de gebruiker alleen kunnen kiezen uit IPE- en HEB-profielen. Deze profielen hebben allemaal een doorsnedeklasse 1 of 2 bij buiging. Bovendien kan de gebruiker alleen kiezen uit de staalsoorten S235, S275, S355 en S460.

Bij de berekeningen houdt het programma geen rekening met gaten in de doorsnede of de stijfheid van de verbindingen. Het is niet mogelijk om een tweede-orde berekening uit te voeren of om balken te dimensioneren op basis van torsie, axiale trek of druk, en twee-assige buiging rond de balk. Daarnaast houdt het programma geen rekening met trillingen of de vereiste brandweerstand. Een opsomming van alle beginselen en toepassingsregels van Eurocode 3 die hierdoor niet meer van toepassing zijn, is te vinden in bijlage E.

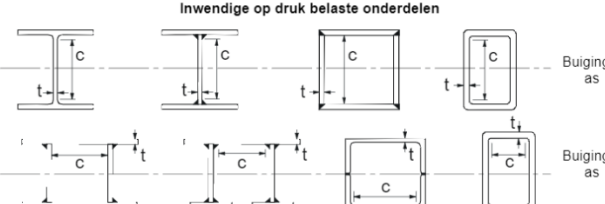
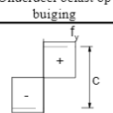
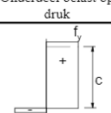
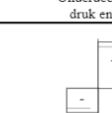
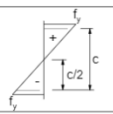
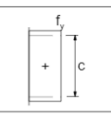
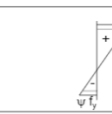
3.3.2 Algemeen

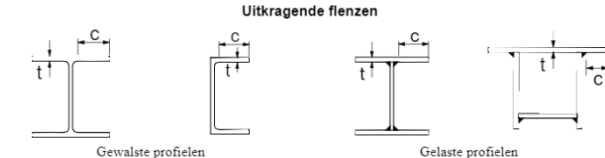
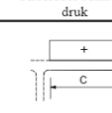
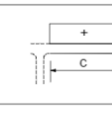
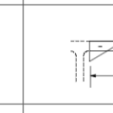
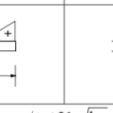
Staal is een materiaal dat op industriële wijze vervaardigd wordt. Het is een legering bestaande uit ijzer en koolstof met een maximaal koolstofgehalte van 2%. Het productieproces begint met ijzererts en cokes, die samen met toeslagmateriaal in een hoogoven worden geplaatst om vloeibaar ruwijzer te verkrijgen. Dit ruwijzer heeft echter nog een te hoog koolstofgehalte (ongeveer 4-5%) en bevat andere onzuiverheden, waardoor het nog nabehandeld moet worden. Na deze nabehandeling wordt vloeibaar staal verkregen. Dit vloeibare staal wordt vervolgens in mallen gegoten en daarna warm of koud gewalst tot de gewenste profielen [26].

Omdat stalen elementen aanvankelijk uit vloeibaar staal worden gevormd en daarna in de gewenste vorm worden gegoten, wordt staal vaak als een isotroop materiaal beschouwd. Dit betekent dat het in alle richtingen vergelijkbare eigenschappen heeft. Staal heeft bovendien een hoge weerstand tegen zowel druk als trek. Afhankelijk van de toegevoegde bestanddelen kunnen verschillende sterkteklassen van staal worden verkregen. De vloeigrens f_y en treksterkte f_u van de verschillende sterkteklassen zijn vastgelegd in Tabel 3.1 van NBN EN 1993-1-1. Alle sterkteklassen van staal hebben dezelfde elasticiteitsmodulus E en glijdingsmodulus G , respectievelijk 210 GPa en 81 GPa [2][26].

Classificatie van doorsneden

De doorsneden van stalen profielen worden op basis van hun plooiweerstand ingedeeld in vier klassen. Naarmate de klasse hoger is, wordt het profiel gevoeliger voor lokaal plooien, wat leidt tot een verminderde weerstand en rotatiecapaciteit van het materiaal. De classificatie van een profiel wordt bepaald door de verhouding van de breedte tot de dikte van zowel het lijf als de flenzen. Voor deze verhoudingen zijn specifieke grenzen vastgesteld om de klasse te bepalen. Elk op druk belast lijf en elke op druk belaste flens krijgt een klasse toegewezen. De klasse van het profiel wordt dan bepaald door de hoogste (meest nadelige) waarde van deze twee. De methoden voor het berekenen van de breedte en dikte van het lijf en de flenzen, evenals de grenswaarden voor deze verhoudingen, zijn vastgelegd in Tabel 5.2 van [2] en worden getoond in Figuur 3.12.

Inwendige op druk belaste onderdelen				
				
Klasse	Onderdeel belast op buiging	Onderdeel belast op druk	Onderdeel belast op druk en buiging	
Spanningsverdeling in de onderdelen (druk is positief)				
1	$c/t \leq 72\epsilon$	$c/t \leq 33\epsilon$	indien $\alpha > 0,5$: $c/t \leq \frac{396\epsilon}{13\alpha - 1}$ indien $\alpha \leq 0,5$: $c/t \leq \frac{36\epsilon}{\alpha}$	
2	$c/t \leq 83\epsilon$	$c/t \leq 38\epsilon$	indien $\alpha > 0,5$: $c/t \leq \frac{456\epsilon}{13\alpha - 1}$ indien $\alpha \leq 0,5$: $c/t \leq \frac{41,5\epsilon}{\alpha}$	
Spanningsverdeling in de onderdelen (druk is positief)				
3	$c/t \leq 124\epsilon$	$c/t \leq 42\epsilon$	indien $\psi > -1$: $c/t \leq \frac{42\epsilon}{0,67 + 0,33\psi}$ indien $\psi \leq -1$: $c/t \leq 62\epsilon(1 - \psi)\sqrt{-\psi}$	
$\epsilon = \sqrt{235/f_y}$	f_y	235	275	355
	ϵ	1,00	0,92	0,81

Uitkragende flenzen				
				
Klasse	Onderdeel belast op druk	Onderdeel belast op druk en buiging		
Spanningsverdeling in de onderdelen (druk is positief)		Uiterste vezel onder druk	Uiterste vezel onder trek	
1	$c/t \leq 9\epsilon$	$c/t \leq \frac{9\epsilon}{\alpha}$	$c/t \leq \frac{9\epsilon}{\alpha\sqrt{\alpha}}$	
2	$c/t \leq 10\epsilon$	$c/t \leq \frac{10\epsilon}{\alpha}$	$c/t \leq \frac{10\epsilon}{\alpha\sqrt{\alpha}}$	
Spanningsverdeling in de onderdelen (druk is positief)				
3	$c/t \leq 14\epsilon$	$c/t \leq 21\epsilon\sqrt{k_\sigma}$		
		Voor k_σ zie EN 1993-1-5		
$\epsilon = \sqrt{235/f_y}$	f_y	235	275	355
	ϵ	1,00	0,92	0,81

Figuur 3.12: Maximale breedte-dikte verhoudingen voor op druk belaste onderdelen [2, p. 48-49]

In Figuur 3.12 wordt een onderscheid gemaakt tussen onderdelen die belast worden op buiging, druk en een combinatie van druk en buiging. Omdat het in het programma niet mogelijk is om normaalkrachten aan te brengen, kunnen onderdelen niet alleen op druk of op druk en buiging worden belast. Daarom wordt in deze masterproef bij de classificatie van de profielen steeds de grenswaarden van de kolom voor onderdelen belast op buiging gebruikt. Deze classificatie bepaalt welke formules gebruikt worden voor de sterkteberekeningen, zoals verder uitgelegd wordt in Sectie 3.3.3.

3.3.3 Sterkte

De sterkte van een doorsnede van een stalen balk wordt gecontroleerd op basis van buiging, afschuiving en een combinatie van deze twee krachten. Bij deze controles wordt de weerstand van de doorsnede vergeleken met de optredende krachten. De weerstand van de doorsnede moet hierbij groter zijn dan de optredende krachten. De sterktecontroles worden altijd uitgevoerd voor de uiterste grenstoestand.

Rekenwaarden

Voordat een controle kan plaatsvinden, moeten de karakteristieke waarden van de weerstanden R_k worden omgezet in rekenwaarden van de weerstanden R_d . Formule (3.67) toont de formule voor deze omzetting.

$$R_d = \frac{R_k}{\gamma_{Mi}} \quad (3.67)$$

De factor γ_{Mi} is de partiële factor voor het materiaal staal. Volgens Eurocode 3 zijn er drie partiële factoren voor staal: een partiële factor voor de weerstand van de doorsnede γ_{M0} , een partiële factor voor de weerstand van elementen op het gebied van stabiliteit γ_{M1} , en een partiële factor voor de weerstand van doorsneden bij trek tot breuk γ_{M2} . Deze laatste wordt niet gebruikt in deze masterproef. Volgens Eurocode 3 en de nationale bijlage zijn de waarden van γ_{M0} en γ_{M1} beide gelijk aan 1 [2][37].

Buiging

In elke dwarsdoorsnede van de stalen balk moet volgens [2] aan de volgende voorwaarde worden voldaan:

$$\frac{M_{Ed}}{M_{c,Rd}} \leq 1,0 \quad (3.68)$$

waarin:

- M_{Ed} de rekenwaarde van het optredende moment is;
- $M_{c,Rd}$ de rekenwaarde van de weerstand van de doorsnede tegen buiging is.

Aangezien er alleen wordt gerekend met profielen in doorsnedeklasse 1 of 2, mag deze weerstand worden berekend op basis van het plastisch weerstandsmoment van de doorsnede. Dit is toegestaan omdat deze doorsneden weinig tot niet gevoelig zijn voor lokaal plooien, waardoor een plastische rekenmethode mag worden gebruikt. Formule (3.69) geeft de formule voor de weerstand van de doorsnede tegen buiging [2].

$$M_{c,Rd} = M_{pl,Rd} = \frac{W_{pl} \times f_y}{\gamma_{M0}} \quad (3.69)$$

waarin:

- W_{pl} het plastisch weerstandsmoment van de doorsnede is;
- f_y de vloeigrens van het staal is;
- γ_{M0} de partiële factor voor de weerstand van de doorsnede is.

De waarde van het plastisch weerstandsmoment van de doorsnede is afkomstig uit de profielen-catalogus van ArcelorMittal [38]. In deze catalogus wordt het plastisch weerstandsmoment rond de y-as $W_{pl,y}$ genomen, aangezien dit overeenkomt met de belaste as in deze masterproef.

Dwarskracht

Voor de controle op afschuiving moet de doorsnede op elk punt van de balk voldoen aan:

$$\frac{V_{Ed}}{V_{c,Rd}} \leq 1,0 \quad (3.70)$$

waarin:

- V_{Ed} de rekenwaarde van de optredende dwarskracht is;
- $V_{c,Rd}$ de rekenwaarde van de weerstand van de doorsnede tegen afschuiving is.

Volgens [2] moet voor de plastische rekenmethode formule (3.71) worden gebruikt om de rekenwaarde van de weerstand van de doorsnede tegen afschuiving te berekenen.

$$V_{c,Rd} = V_{pl,Rd} = \frac{A_v \times (f_y / \sqrt{3})}{\gamma_{M0}} \quad (3.71)$$

In formule (3.71) is A_v de oppervlakte van het werkzame afschuifoppervlak [2].

Combinatie van buiging en dwarskracht

Wanneer er zowel een buigmoment als een dwarskracht op een punt in de balk werken, moet een controle worden uitgevoerd die rekening houdt met deze combinatie. Bij deze controle moet de gereduceerde momentweerstand groter zijn dan het optredende buigmoment. Deze controle is echter alleen nodig als de optredende dwarskracht groter is dan de helft van de weerstand tegen dwarskracht bij vloeien ($V_{Ed} \geq 0,5 \times V_{pl,Rd}$) [2].

Indien de controle nodig is, wordt de gereduceerde momentweerstand berekend met formule (3.69) waarbij een gereduceerde vloeigrens wordt gebruikt. De gereduceerde vloeigrens wordt berekend met formule (3.72).

$$f_{y,gereduceerd} = (1 - \rho) \times f_y \quad (3.72)$$

met:

$$\rho = \left(\frac{2 \times V_{Ed}}{V_{pl,Rd}} - 1 \right)^2 \quad (3.73)$$

waarin:

- f_y de vloeigrens van het staal is;
- V_{Ed} de rekenwaarde van de optredende dwarskracht is;
- $V_{pl,Rd}$ de rekenwaarde van de plastische dwarskrachtweerstand van de doorsnede is.

De plastische dwarskrachtweerstand wordt berekend met formule (3.71).

3.3.4 Stabiliteit

Stalen balken die voornamelijk belast worden door buiging moeten gecontroleerd worden op het instabiliteitsverschijnsel genaamd kip. Kip treedt op wanneer de vrije gedrukte flens zijdelings uitbuigt, gecombineerd met een rotatie rond de lengteas van de balk [39]. Dit fenomeen kan optreden zelfs bij een buigmoment lager dan de theoretische weerstand tegen buiging en dient daarom vermeden te worden. Volgens Eurocode 3 moet de rekenwaarde van de kipweerstand $M_{b,Rd}$ van een balk voldoen aan [2]:

$$\frac{M_{Ed}}{M_{b,Rd}} \leq 1,0 \quad (3.74)$$

De rekenwaarde van de kipweerstand wordt berekend met formule (3.75) [2].

$$M_{b,Rd} = \chi_{LT} \times W_{pl,y} \times \frac{f_y}{\gamma_{M1}} \quad (3.75)$$

waarin:

- χ_{LT} de reductiefactor voor kip is;
- $W_{pl,y}$ het plastisch weerstandsmoment van de doorsnede is;
- f_y de vloeigrens van het staal is;
- γ_{M1} de partiële factor voor de stabiliteitsweerstand van elementen is.

Het plastisch weerstandsmoment $W_{pl,y}$ wordt gebruikt voor doorsnedeklassen 1 en 2. De reductiefactor voor kip wordt berekend met formule (3.76) [2].

$$\chi_{LT} = \frac{1}{\Phi_{LT} + \sqrt{\Phi_{LT}^2 - \bar{\lambda}_{LT}^2}} \leq 1,0 \quad (3.76)$$

waarin:

- $\Phi_{LT} = 0,5 \times [1 + \alpha_{LT} \times (\bar{\lambda}_{LT} - 0,2) + \bar{\lambda}_{LT}^2]$
- α_{LT} de imperfectiefactor is;
- $\bar{\lambda}_{LT} = \sqrt{\frac{W_{pl,y} \times f_y}{M_{cr}}}$
- M_{cr} het kritieke elastische kipmoment is.

De waarde van α_{LT} is vastgelegd in Tabel 3.13 [2].

Tabel 3.13: Aanbevolen waarden voor de imperfectiefactor α_{LT} [2, p. 67]

Doorsnede	Begrenzing	Kipkromme	Imperfectiefactor α_{LT} [-]
Gewalste I-profielen	$h/b \leq 2$	a	0,21
	$h/b > 2$	b	0,34
Gelaste I-profielen	$h/b \leq 2$	c	0,49
	$h/b > 2$	d	0,76
Andere doorsneden	-	d	0,76

Zoals blijkt uit Tabel 3.13, varieert de imperfectiefactor afhankelijk van de kipkromme. Deze kipkromme wordt beschreven in Tabel 6.4 van NBN EN 1993-1-1, terwijl de specifieke waarden voor α_{LT} zijn opgenomen in Tabel 6.3 van dezelfde norm. Beide tabellen zijn gecombineerd in Tabel 3.13 [2].

Het kritieke elastische weerstandsmoment wordt berekend volgens de Nationale bijlage van Eurocode 3 met formule (3.77) [37].

$$M_{cr} = C_1 \times \frac{\pi^2 \times E \times I_z}{(k \times L)^2} \times \left(\sqrt{\left(\frac{k}{k_w}\right)^2 \times \frac{I_w}{I_z} + \frac{(k \times L)^2 \times G \times I_t}{\pi^2 \times E \times I_z}} + (C_2 \times z_g)^2 - C_2 \times z_g \right) \quad (3.77)$$

waarin:

- E de elasticiteitsmodulus van staal is;
- G de glijdingsmodulus van staal is;
- I_z het traagheidsmoment om de z-As is;
- I_t het torsietraagheidsmoment is;
- I_w de krommingsconstante is;
- L de afstand tussen twee steunpunten is die laterale verplaatsing beperken;
- k en k_w effectieve lengte factoren zijn;
- z_g de afstand is tussen het aangrijpingspunt van de belasting en het afschuifmiddelpunt;
- C_1 en C_2 constanten zijn die de belasting en het steunpuntgedrag in rekening brengen.

De factoren k en k_w worden verondersteld gelijk aan 1 te zijn, aangezien er sprake is van normaal steunpuntgedrag. De factor z_g is positief wanneer de belasting bovenaan werkt en negatief wanneer de belasting onderaan aangrijpt. In deze masterproef wordt deze factor negatief genomen, omdat dit de meest voorkomende situatie is en tevens de meest conservatieve benadering vormt. De factoren C_1 en C_2 moeten worden berekend op basis van het moment bij de steunpunten en het moment veroorzaakt door de belasting op de balk. Deze factoren worden bepaald aan de hand van tabellen in de Belgische nationale bijlage van Eurocode 3. Aangezien voor deze tabellen geen wiskundige relaties beschikbaar zijn, is het niet mogelijk om met het programma de waarde van deze factoren te bepalen. Daarom wordt in deze masterproef de waarde van de factoren C_1 en C_2 gelijkgesteld aan de waarde voor een uniform verdeelde belasting op een opgelegde balk, namelijk respectievelijk 1,127 en 0,454 [37].

3.3.5 Doorbuiging

De doorbuiging moet worden gecontroleerd BGT-K, BGT-F en BGT-Q, elk met een andere maximale toelaatbare waarde. Eurocode 3 schrijft geen absolute maximale doorbuigingswaarden voor, echter kan deze door de gebruiker wel zelf worden opgelegd. De doorbuigingen worden verkregen uit de eindige-elementenmethode. Hierbij zijn geen extra factoren nodig om rekening te houden met verschijnselen zoals kruip of scheurvorming.

3.4 Hout

Dit hoofdstuk belicht de formules, ontwerpeisen en controleprocedures die worden toegepast bij het dimensioneren van houten balken. Deze formules, eisen en controles zijn gebaseerd op de Eurocode voor houtconstructies, met name Eurocode 5 [3]. Het ontwerp en de controle van houten balken zijn gebaseerd op de criteria van sterkte, stabiliteit en doorbuiging van de balk. Alvorens in te gaan op de specifieke berekeningen en controlestappen van deze criteria, wordt eerst een overzicht gegeven van de aannames en beperkingen die gelden voor houten balken binnen de context van deze masterproef. Dit wordt gevolgd door een sectie waarin de algemene eigenschappen van het materiaal worden toegelicht.

3.4.1 Aannames en beperkingen

In het programma kan de gebruiker uitsluitend houten balken modelleren met een constante rechthoekige doorsnede over de gehele lengte van de balk. Daarnaast wordt aangenomen dat het kruipgedrag van deze balk overal gelijk is en dat de vezels in de lengterichting liggen. De gebruiker kan kiezen uit houtsterkteklassen voor zowel gezaagd hout als gelijmd gelamineerd hout.

Het programma biedt geen mogelijkheden om een initiële kromming van de balk op te geven of belastingen met een excentriciteit te laten aangrijpen. Daarnaast kan de stijfheid van verbindingen niet worden aangegeven, waardoor het programma geen rekening kan houden met verschuivingen en bijkomende krachten als gevolg daarvan. Tenslotte kunnen gaten voor verbindingen in de dwarsdoorsnede niet worden meegenomen in de berekeningen. Een opsomming van alle beginselen en toepassingsregels van Eurocode 5 die hierdoor niet meer van toepassing zijn, is te vinden in bijlage F.

3.4.2 Algemene eigenschappen

Hout is een natuurlijk, anisotroop product dat verder kan worden bewerkt om andere eigenschappen te verkrijgen, op deze manier worden verschillende types hout gecreëerd. In deze masterproef is het mogelijk om houten balken te ontwerpen voor verschillende types hout, waaronder gezaagd loofhout (aangeduid als Dxx), gezaagd naaldhout (aangeduid als Cxx), gelijmd gelamineerd hout (aangeduid als GLxxh) en gecombineerd gelijmd gelamineerd hout (aangeduid als GLxxc). De laatste twee types bestaan uit lamellen die aan elkaar zijn gelijmd. Het verschil tussen gelijmd gelamineerd hout en gecombineerd gelijmd gelamineerd hout is dat bij het laatste type in het midden van de dwarsdoorsnede een lagere sterkteklasse hout wordt gebruikt dan aan de buitenkanten. De aanduiding 'xx' bij deze materialen verwijst naar de sterkteklasse van het hout, zoals bijvoorbeeld C24 en GL18h. Deze sterkteklassen zijn gestandaardiseerd en worden vermeld in NBN EN 338 [40] voor gezaagd hout en in NBN EN 14080 [41] voor gelijmd gelamineerd hout.

Voor het ontwerpen van houten balken is het noodzakelijk om rekening te houden met zowel de belastingsduurklassen als de klimaatklassen waarin de balk wordt geplaatst. De belastingsduurklassen beschouwen het effect van een constante belasting gedurende een bepaalde tijdsperiode op de constructie, zoals beschreven in Eurocode 5. De klimaatklassen nemen daarentegen de invloed van vocht op het hout in overweging. Er bestaan drie klimaatklassen, en de specifieke criteria om tot een bepaalde klimaatklasse te behoren zijn vastgelegd in 2.3.1.3 van NBN EN 1995-1-1 [3].

Daarnaast bevat de nationale bijlage van Eurocode 5 een toewijzing van klimaatklassen voor bepaalde ruimtes [42]. Het is echter aan de gebruiker om de klimaatklasse te specificeren in het programma.

Om deze effecten op zowel de sterkte als de vervormingen in overweging te nemen, worden respectievelijk de K_{mod} - en K_{def} -factor toegepast. De K_{mod} -factor wordt verder toegelicht in Sectie 3.4.3, terwijl de K_{def} -factor wordt behandeld in Sectie 3.4.5.

Tenslotte wordt er gebruik gemaakt van de methode van de partiële factoren om rekening te houden met ongunstige onzekerheden. De partiële factor γ_M voor hout is afhankelijk van het type hout en bedraagt 1,3 voor gezaagd hout en 1,25 voor gelijmd gelamineerd hout.

3.4.3 Sterkte

De sterkte van de doorsnede wordt gecontroleerd op basis van buiging en op basis van afschuiving. Dit gebeurt door de rekenwaardes voor de buigspanning en afschuifspanning in de doorsnede te vergelijken met de rekenwaardes voor de corresponderende sterkte-eigenschap van het materiaal. Om de rekenwaardes van de sterkte-eigenschap van het materiaal te verkrijgen, moet de karakteristieke waarde vermenigvuldigd worden met enkele factoren en gedeeld worden door de partiële factor. De formule voor het verkrijgen van de rekenwaarde voor een bepaalde sterkte-eigenschap wordt weergegeven door volgende formule:

$$X_d = k_h \times k_{mod} \times \frac{X_k}{\gamma_M} \quad (3.78)$$

waarin:

- X_d de rekenwaarde van de sterkte-eigenschap is;
- k_h de hoogtefactor is die rekening houdt met de afmetingen van het element;
- k_{mod} de modificatiefactor is voor de belastingsduur en het vochtgehalte;
- X_k de karakteristieke waarde van de sterkte-eigenschap is;
- γ_M de partiële factor van het materiaal is.

De waarde van de modificatiefactor, die het effect van de belastingsduur en het vochtgehalte in rekening brengt, kan worden gevonden in Tabel 3.1 van Eurocode 5. Deze waarden voor gezaagd hout en gelijmd gelamineerd hout worden weergegeven in Tabel 3.14 van deze masterproef. Volgens NBN EN 1995-1-1: 3.1.3 (2) moet bij het optreden van meerdere belastingsduurklassen de modificatiefactor voor de kortste duur gekozen worden [3].

De hoogtefactor k_h , die rekening houdt met de afmetingen van het element, wordt toegepast omdat voor het bepalen van de formules een referentiehoogte is gebruikt. Als het element kleiner is dan deze referentiehoogte, mag de sterkte-eigenschap worden verhoogd. De referentiehoogte voor gezaagd hout en gelijmd gelamineerd hout is echter niet gelijk. Voor gezaagd hout is de referentiehoogte 150 mm, terwijl deze voor gelijmd gelamineerd hout 600 mm bedraagt. De factor k_h voor gezaagd hout wordt gegeven door formule (3.79) en voor gelijmd gelamineerd hout door formule (3.80). Indien de hoogte van het element groter is dan de referentiehoogte, is de waarde van de hoogtefactor k_h gelijk aan 1 [3].

Tabel 3.14: Waarden voor k_{mod} [3, p. 34]

Materiaal	Norm	Klimaat- klassen	Belastingsduurklassen				
			Blijvend	Lang	Middellang	Kort	Zeer kort
Gezaagd hout	EN 14081-1	1	0,60	0,70	0,80	0,90	1,10
		2	0,60	0,70	0,80	0,90	1,10
		3	0,50	0,55	0,65	0,70	0,90
Gelijmd gelamineerd hout	EN 14080	1	0,60	0,70	0,80	0,90	1,10
		2	0,60	0,70	0,80	0,90	1,10
		3	0,50	0,55	0,65	0,70	0,90

$$k_{h,\text{gezaagd}} = \min \left\{ \left(\frac{150}{h} \right)^{0.2} \right. \\ \left. 1.3 \right. \quad (3.79)$$

$$k_{h,\text{gelijmd}} = \min \left\{ \left(\frac{600}{h} \right)^{0.1} \right. \\ \left. 1.1 \right. \quad (3.80)$$

De waarde van de partiële factor γ_M voor de verschillende types hout is gegeven in Sectie 3.4.2. De karakteristieke waarde van de sterkte-eigenschap X_k is afhankelijk van de sterkteklasse en het type hout zoals voorgeschreven in de corresponderende norm.

Buiging

De controle van de buigcapaciteit in de dwarsdoorsnede dient conform Eurocode 5 te gebeuren aan de hand van de formules (6.11) en (6.12) van deze Eurocode [3]. Deze formules zijn weergegeven in formules (3.81) en (3.82) in deze masterproef.

$$\frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \times \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (3.81)$$

$$k_m \times \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (3.82)$$

Waarin:

- $\sigma_{m,y,d}$ en $\sigma_{m,z,d}$ de rekenwaarden van de buigspanningen om de hoofdasen zijn;
- $f_{m,y,d}$ en $f_{m,z,d}$ de overeenkomstige rekenwaarden van de buigsterkten zijn;
- k_m een factor is die de herverdeling van de spanningen en de heterogeniteit van de dwarsdoorsnede in acht neemt.

Aangezien de belastingen slechts in de verticale richting kunnen worden aangebracht, zal er nooit een buigspanning om de zwakke hoofdas optreden, en zal $\sigma_{m,z,d}$ altijd nul zijn. Daarnaast is, in overeenstemming met NBN EN 1995-1-1: 6.1.6 (2), de factor k_m gelijk aan 0,7 voor een rechthoekige doorsnede [3]. Aangezien deze factor kleiner is dan één, is deze gunstig en kan de controle op buiging worden vereenvoudigd tot formule (3.83):

$$\sigma_{m,y,d} \leq f_{m,y,d} \quad (3.83)$$

De buigspanning $\sigma_{m,y,d}$ wordt berekend met formule (3.84):

$$\sigma_{m,y,d} = \frac{M_{y,Ed}}{W_y} \quad (3.84)$$

Met:

$$W_y = \frac{b \times h^2}{6} \quad (3.85)$$

Waarin:

- $M_{y,Ed}$ het buigmoment rond de sterke as is;
- W_y het weerstandsmoment⁷ rond de sterke as is;
- b de breedte van het element is;
- h de hoogte van het element is.

De rekenwaarde voor de buigsterkte wordt berekend door deze in te vullen in formule (3.78), samen met de correcte factoren. De dwarsdoorsnede wordt als voldoende beschouwd op basis van buiging indien de buigspanning kleiner is dan de buigsterkte.

Afschuiving

De controle op afschuiving in de doorsnede wordt uitgevoerd aan de hand van formule (6.13) van Eurocode 5 [3]. Deze formule voor de controle wordt weergegeven in formule (3.86):

$$\tau_d \leq f_{v,d} \quad (3.86)$$

Waarin:

- τ_d de rekenwaarde van de afschuifspanning is;
- $f_{v,d}$ de rekenwaarde van de afschuifsterkte is.

De rekenwaarde van de afschuifspanning voor hout wordt doorgaans bepaald aan de hand van formule (3.87). Deze formule kan worden toegepast aangezien het faalmechanisme bros is, wat impliceert dat er geen plastische vervorming optreedt. Aangezien er enkel rechthoekige doorsnedes voor hout mogelijk zijn, kan deze formule worden vereenvoudigd tot formule (3.88). In dit geval treedt de maximale schuifspanning in het midden van de doorsnede op.

$$\tau = \frac{V_{Ed} \times S}{I \times b} \quad (3.87)$$

$$\tau = \frac{3 \times V_{Ed}}{2 \times b \times h} \quad (3.88)$$

Waarin:

- V_{Ed} de rekenwaarde van de optredende dwarskracht is;

⁷ Formule (3.85) is enkel geldig voor rechthoekige doorsnedes

- S het statisch moment is;
- I het traagheidsmoment is;
- b de breedte van de doorsnede is;
- h de hoogte van de doorsnede is.

Volgens [43] dient rekening te worden gehouden met de invloed van scheuren op de afschuifweerstand. Dit wordt bereikt door de breedte te vermenigvuldigen met een factor k_{cr} . Deze factor is volgens dezelfde bron gelijk aan 0,67 voor zowel gezaagd als gelijmd gelamineerd hout.

De rekenwaarde van de afschuifsterkte wordt op vergelijkbare wijze bepaald als de rekenwaarde van de buigsterkte, door deze in formule (3.78) in te vullen. De dwarsdoorsnede wordt als adequaat beschouwd indien de optredende schuifspanning overal lager is dan de afschuifsterkte van de doorsnede.

3.4.4 Stabiliteit

Net zoals bij stalen balken moeten houten balken worden gecontroleerd op kip. De controle voor de weerstand tegen kip van een houten balk wordt weergegeven in formule (3.89) [3].

$$\sigma_{m,d} \leq k_{crit} \times f_{m,d} \quad (3.89)$$

waarin:

- $\sigma_{m,d}$ de rekenwaarde van de buigspanning is;
- $f_{m,d}$ de rekenwaarde van de buigsterkte is;
- k_{crit} een reductiefactor is die rekening houdt met de gevolgen van kip.

De reductiefactor wordt berekend met formule (3.90) [3].

$$k_{crit} = \begin{cases} 1 & \text{voor } \lambda_{rel,m} \leq 0,75 \\ 1,56 - 0,75 \times \lambda_{rel,m} & \text{voor } 0,75 < \lambda_{rel,m} \leq 1,4 \\ \frac{1}{\lambda_{rel,m}^2} & \text{voor } 1,4 < \lambda_{rel,m} \end{cases} \quad (3.90)$$

waarin $\lambda_{rel,m}$ de relatieve slankheid van de balk is. Zoals te zien is in formule (3.90) wordt de buigweerstand niet verlaagd indien de relatieve slankheid kleiner of gelijk is aan 0,75. Deze relatieve slankheid wordt berekend met formule (3.91) [3].

$$\lambda_{rel,m} = \sqrt{\frac{f_{m,k}}{\sigma_{m,crit}}} \quad (3.91)$$

met:

$$\sigma_{m,crit} = \frac{\pi \times \sqrt{E_{0,05} \times I_z \times G_{0,05} \times I_{tor}}}{l_{ef} \times W_y} \quad (3.92)$$

waarin:

- $f_{m,k}$ de karakteristieke waarde van de buigsterkte is;
- $\sigma_{m,crit}$ de kritische buigspanning is;
- $E_{0,05}$ de 5-percentielwaarde van de elasticiteitsmodulus evenwijdig aan de vezelrichting is;
- $G_{0,05}$ de 5-percentielwaarde van de afschuivingsmodulus evenwijdig aan de vezelrichting is;
- I_z het traagheidsmoment om de z-as is;
- I_{tor} het torsietraagheidsmoment is;
- l_{ef} de meewerkende balklengte is;
- W_y het weerstandsmoment om de y-as is.

De 5-percentielwaarde voor de elasticiteitsmodulus en afschuivingsmodulus voor gelijmd gelamineerd hout zijn te vinden in NBN EN 14080 [41]. De waarde van de 5-percentielwaarde voor de elasticiteitsmodulus van gezaagd hout is te vinden in NBN EN 338, maar deze norm bevat geen 5-percentielwaarden voor de afschuivingsmodulus [40]. Voor de berekening van deze afschuivingsmodulus worden formules uit de nieuwe Eurocode voor hout gebruikt. Formule (3.93) toont de formule voor hardhout en formule (3.94) voor zachthout [44].

$$G_{0,05,hardhout} = 0,67 \times G_{mean} \quad (3.93)$$

$$G_{0,05,zachthout} = 0,83 \times G_{mean} \quad (3.94)$$

Het traagheidsmoment rond de zwakke as wordt berekend met formule (3.95) en het torsietraagheidsmoment van een rechthoekige doorsnede met formule (3.96) [43].

$$I_z = \frac{h \times b^3}{12} \quad (3.95)$$

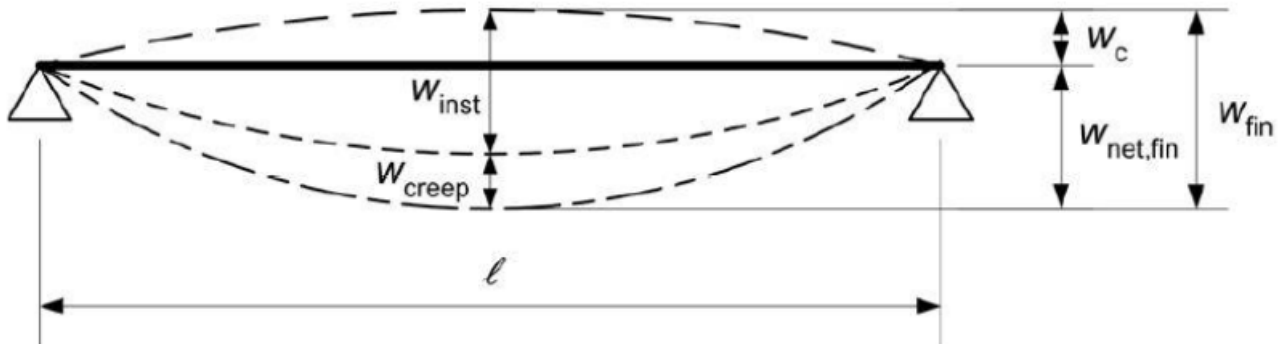
$$I_{tor} = \frac{h \times b^3}{3} \quad (3.96)$$

De meewerkende balklengte wordt doorgaans gelijkgesteld aan de lengte van de overspanning maar kan ook verkleind worden door steunen te plaatsen die het kippen van de balk verhinderen. Volgens Tabel 6.1 van [3] mag deze lengte worden vermenigvuldigd met een factor kleiner dan één, afhankelijk van de belasting. Aangezien in het programma verschillende belastingen op verschillende plaatsen kunnen aangrijpen, is het moeilijk te bepalen welke factor gehanteerd moet worden. Daarom wordt uit conservatieve overwegingen de volledige lengte in rekening gebracht. Daarnaast wordt deze lengte nog vermeerderd met tweemaal de hoogte van de balk, aangezien de krachten op de drukzijde aangrijpen. Dit wordt overal toegepast, zelfs als de belasting aan de trekzijde aangrijpt, omdat dit de meest conservatieve benadering is.

3.4.5 Doorbuiging

De doorbuiging van een houten balk wordt gecontroleerd door de uiteindelijke netto doorbuiging $W_{net,fin}$ te vergelijken met een maximale waarde. Deze maximale waarde wordt bepaald door

de gebruiker en wordt beschreven als de verhouding van de lengte van de overspanning tot een bepaalde factor, bijvoorbeeld $\frac{L}{300}$. Volgens Eurocode 5 zijn er verschillende bijdragen aan de doorbuiging van een houten balk die leiden tot de netto uiteindelijke doorbuiging $W_{\text{net,fin}}$. Deze diverse bijdragen worden geïllustreerd in Figuur 3.13, afkomstig uit Eurocode 5 [3].



Figuur 3.13: Verschillende bijdragen doorbuiging hout [3, p. 20]

Zoals te zien is in Figuur 3.13, is de netto uiteindelijke doorbuiging $W_{\text{net,fin}}$ het verschil tussen de uiteindelijke doorbuiging W_{fin} en de initiële opbuiging W_c . De uiteindeindelijke doorbuiging bestaat uit twee doorbuigingen. Ten eerste is er de onmiddellijke doorbuiging W_{inst} . Dit is de doorbuiging die direct optreedt bij het aanbrengen van de belasting op de balk. Ten tweede is er de doorbuiging door kruip W_{creep} . Dit is de bijkomende doorbuiging door het langdurig belasten van de balk. Volgens NBN EN 1995-1-1: 2.2.3 (5) mag voor elementen met hetzelfde kruipgedrag en in de veronderstelling dat er een lineair verband is tussen de belasting en de bijbehorende vervorming, de uiteindelijke doorbuiging worden berekend met Formule (3.97).

$$W_{\text{fin}} = W_{\text{fin,G}} + W_{\text{fin,Q}_1} + W_{\text{fin,Q}_i} \quad (3.97)$$

Waarin:

- W_{fin} de totale uiteindelijke doorbuiging is;
- $W_{\text{fin,G}}$ de uiteindelijke doorbuiging enkel door de blijvende belastingen is;
- $W_{\text{fin,Q}_1}$ de uiteindelijke doorbuiging enkel door de overheersende veranderlijke belasting is;
- $W_{\text{fin,Q}_i}$ de uiteindelijke doorbuiging enkel door de niet-overheersende veranderlijke belastingen is.

De uiteindelijke doorbuiging door de blijvende belastingen, de overheersende veranderlijke belasting en de niet-overheersende veranderlijke belastingen worden berekend door respectievelijk Formules (3.98), (3.99) en (3.100).

$$W_{\text{fin,G}} = W_{\text{inst,G}} (1 + k_{\text{def}}) \quad (3.98)$$

$$W_{\text{fin,Q}_1} = W_{\text{inst,Q}_1} (1 + \psi_{2,1} k_{\text{def}}) \quad (3.99)$$

$$W_{\text{fin,Q}_i} = W_{\text{inst,Q}_i} (\psi_{0,i} + \psi_{2,i} k_{\text{def}}) \quad (3.100)$$

Waarin:

- $W_{\text{inst,G}}, W_{\text{inst,Q}_1}, W_{\text{inst,Q}_i}$ de onmiddellijke doorbuiging van de respectievelijke belasting zijn;
- k_{def} factor is die rekening houdt met het vochtgehalte voor vervormingen;
- $\psi_{2,1}, \psi_{2,i}$ factor is voor de quasi-permante waarde van de veranderlijke belastingen;
- $\psi_{0,i}$ factor is voor de combinatie waarde van de veranderlijke belastingen.

Hierbij moeten de onmiddellijke doorbuigingen, volgens Eurocode 5, worden berekend voor de karakteristieke belastingscombinatie (BGT-K) [3]. Deze waarden van de onmiddellijke doorbuigingen worden rechtstreeks verkregen uit de eindige-elementenmethode. Bij deze methode wordt geen factor toegevoegd om doorbuiging door dwarskrachten in rekening te brengen, waardoor dit aspect in deze masterproef wordt verwaarloosd. De waarden van de factor k_{def} , die rekening houdt met het vochtgehalte voor vervormingen, zijn vastgelegd in Tabel 3.2 van Eurocode 5 [3]. De waarden van deze factor voor gezaagd en gelijmd gelamineerd hout, in functie van de klimaatklasse, worden weergegeven in Tabel 3.15 van deze masterproef.

Tabel 3.15: Waarden voor k_{def}

Materiaal	Norm	Klimaatklasse		
		1	2	3
Gezaagd hout	EN 14081-1	0,60	0,80	2,00
Gelijmd gelamineerd hout	EN 14080	0,60	0,80	2,00

3.5 Conclusie

Allereerst is het cruciaal dat de snedekrachten en vervormingen nauwkeurig worden bepaald volgens de richtlijnen van Eurocode 0 [8]. De eindige-elementenmethode wordt gebruikt om de balk te analyseren onder verschillende belastingscombinaties en in verschillende grenstoestanden. Deze snedekrachten en vervormingen vormen 50% van de vergelijking om te bepalen of een balk voldoet aan de eisen of niet.

De andere helft van de vergelijking hangt af van het materiaaltype en de doorsnede-eigenschappen. Voor elk materiaal, zoals beton, staal en hout, zijn er unieke formules voor het berekenen van de weerstanden. Ondanks de verschillen in berekeningen worden balken voornamelijk gecontroleerd op buiging, afschuiving en vervormingen, ongeacht het gebruikte materiaal.

Hoofdstuk 4

Back end van het programma

In dit hoofdstuk wordt de “*back end*” van het programma uitgelegd. De back end omvat alle geprogrammeerde functies die onder de motorkap van het programma zitten. Hierin zijn de formules uit Hoofdstuk 3 geïmplementeerd. Naast deze formules bevat de back end ook een database voor efficiënt en eenvoudig databeheer binnen het programma.

Eerst wordt besproken waarom en hoe een database geïmplementeerd is. Daarna wordt de analyse van een balk behandeld, gevolgd door de controle en optimalisatie van betonnen, stalen en houten balken, volgens dezelfde volgorde als in Hoofdstuk 3. Het is niet mogelijk om elke regel code te tonen en uit te leggen. In plaats daarvan worden de belangrijkste stukken code getoond en wordt de algemene gedachtengang en het proces van de berekeningen en controles uitgelegd. Hoewel de code in het Engels is geschreven, wordt in deze scriptie Nederlandstalige termen gebruikt. Indien nodig, worden de corresponderende Engelse termen in cursief vermeld.

4.1 Database

Een database is een structuur waarin informatie wordt opgeslagen, die kan worden toegevoegd en opgevraagd wanneer nodig. In deze masterproef is het gebruik van een database cruciaal vanwege de grote hoeveelheid data die het programma moet bijhouden. Bijvoorbeeld, op verschillende punten in de balk moeten dwarskrachten, momenten en vervorming worden opgeslagen voor alle verschillende belastingcombinaties voor alle grenstoestanden.

In dit gedeelte worden eerst de voordelen van het gebruik van een database binnen deze masterproef besproken. Vervolgens wordt uitgelegd welke verschillende soorten informatie door de database worden beheerd. Ten slotte wordt het ontwerp en gebruik van de database toegelicht.

4.1.1 Voordelen

Het gebruik van een database biedt drie grote voordelen. Ten eerste zorgt het voor een overzichtelijke opslagstructuur waardoor het voor de programmeur eenvoudiger en efficiënter is om het programma te ontwikkelen. Ten tweede hoeft niet alle data in het werkgeheugen te worden opgeslagen, wat de prestaties van het programma verbetert. Ten derde kan er bij het opvragen van data automatisch een filter worden toegepast. Dit beperkt de hoeveelheid data in het werkgeheugen en maakt het programma efficiënter omdat een externe filterfunctie niet nodig is.

4.1.2 Opgeslagen informatie

Een database bestaat uit verschillende tabellen die elk specifieke informatie bijhouden om alle gegevens en resultaten van het programma te beheren. In totaal worden 27 tabellen gebruikt, verdeeld in twee categorieën: tabellen met vooraf geprogrammeerde informatie en tabellen voor dynamische gegevensverwerking.

Vooraf geprogrammeerde tabellen

Deze tabellen bevatten essentiële informatie die niet verandert tijdens berekeningen en worden enkel gebruikt voor het opvragen van informatie. Deze tabellen omvatten:

- de sterkteklassen van beton;
- de milieuklassen van beton, inclusief bijhorende minimale dekking en maximale scheurwijdte;
- de staalprofielen;
- de waarden van de k_{mod} - en k_{def} -factoren voor hout, afhankelijk van het materiaal, de klimaatklasse en de belastingsduur.
- de sterkteklassen van houtsoorten.

Dynamische gegevensverwerking

Deze tabellen zijn initieel leeg en worden tijdens het gebruik van het programma gevuld met informatie afhankelijk van de berekening. Deze tabellen omvatten:

- de verschillende belastingsgevallen;
- de verschillende belastingscombinaties;
- de waarde van elk belastingsgeval in elke belastingscombinatie;
- de interne krachten en vervorming van de balk;
- de verschillende belastingen en bijhorende waarden;
- de steunpunten en bijhorende reactiekrachten;
- de waarden van de geplaatste en benodigde wapening bij beton;
- de waarden van de scheurwijdte bij beton;
- de waarden van de vervormingen zonder kruip bij beton;
- de waarden van de vervormingen met kruip bij beton;
- de waarden van de beton- en wapeningsspanningen bij beton;
- de waarden van de geplaatste wapening in een doorsnede bij beton;
- de waarden van de grootheden van een doorsnede bij beton;
- de diameter en coördinaten van doorlopende omgeplooid wapeningsstaven;
- de diameter en coördinaten van extra bijgelegde wapeningsstaven;
- de diameter, tussenafstand en coördinaten van beugels;

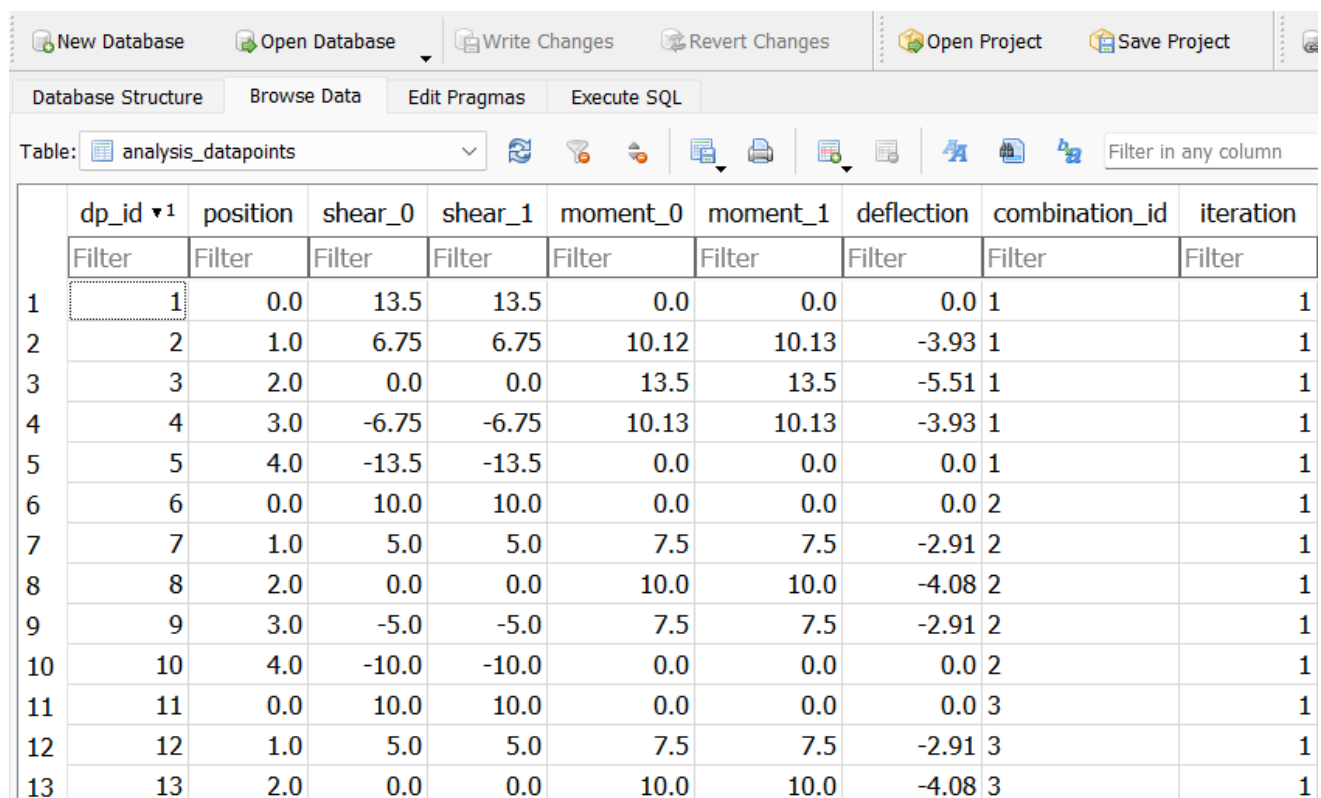
- de controle op basis van sterkte voor stalen balken;
- de controle op basis van stabiliteit voor stalen balken;
- de controle op basis van doorbuiging voor stalen balken;
- de controle op basis van sterkte voor houten balken;
- de controle op basis van stabiliteit voor houten balken;
- de waarden van onmiddellijke en finale doorbuiging voor houten balken;

4.1.3 Ontwerp

Voor het ontwerpen van de database wordt gebruik gemaakt van de applicatie DB Browser. DB Browser is een gratis en gebruiksvriendelijke applicatie waarmee databases voor SQLite kunnen worden gecreëerd, beheerd en aangepast [11]. SQLite is een databasemanagementsysteem gebaseerd op SQL (Structured Query Language), een programmeertaal die het ontwerpen en beheren van databases mogelijk maakt [45][46].

Opbouw tabel

Zoals eerder besproken, bestaat een database uit verschillende tabellen. Elke kolom van een tabel wordt geïdentificeerd met een sleutel zoals bijvoorbeeld “naam”, wat gebruikt zou kunnen worden in een tabel die informatie over personen opslaat. Elke rij in zo een kolom bevat dan een naam. In de ontworpen database bevat elke tabel specifieke sleutels afhankelijk van wat er opgeslagen moet worden. Figuur 4.1 illustreert een voorbeeld van zo een tabel, specifiek de tabel die de interne krachten en vervormingen beheert.



	dp_id ▼ ¹	position	shear_0	shear_1	moment_0	moment_1	deflection	combination_id	iteration
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	0.0	13.5	13.5	0.0	0.0	0.0	1	1
2	2	1.0	6.75	6.75	10.12	10.13	-3.93	1	1
3	3	2.0	0.0	0.0	13.5	13.5	-5.51	1	1
4	4	3.0	-6.75	-6.75	10.13	10.13	-3.93	1	1
5	5	4.0	-13.5	-13.5	0.0	0.0	0.0	1	1
6	6	0.0	10.0	10.0	0.0	0.0	0.0	2	1
7	7	1.0	5.0	5.0	7.5	7.5	-2.91	2	1
8	8	2.0	0.0	0.0	10.0	10.0	-4.08	2	1
9	9	3.0	-5.0	-5.0	7.5	7.5	-2.91	2	1
10	10	4.0	-10.0	-10.0	0.0	0.0	0.0	2	1
11	11	0.0	10.0	10.0	0.0	0.0	0.0	3	1
12	12	1.0	5.0	5.0	7.5	7.5	-2.91	3	1
13	13	2.0	0.0	0.0	10.0	10.0	-4.08	3	1

Figuur 4.1: Tabel voor interne krachten en vervormingen

De tabel in Figuur 4.1 bevat negen kolommen. De eerste kolom is bijzonder omdat deze een unieke waarde aan elke rij toekent. Deze unieke waarde kan worden gebruikt om verbindingen te maken tussen verschillende tabellen en wordt de primaire sleutel genoemd. Kolom 2 tot en met 7 bevatten specifieke resultaten voor de interne krachten en vervorming. De achtste kolom maakt een verbinding met een andere tabel die alle verschillende belastingscombinaties bijhoudt. De sleutel van deze kolom, de vreemde sleutel, verwijst naar de primaire sleutel van de tabel met belastingscombinaties. Hierdoor worden de interne krachten en vervorming op een specifiek punt gekoppeld aan een bepaalde belastingscombinatie. De laatste kolom houdt bij hoeveelste iteratie van de berekening het betreft. Bij sommige berekeningen is het nodig om deze meerdere keren uit te voeren en met deze kolom wordt de data aan een specifieke iteratie toegekend.

Opbouw algemene databasestructuur

De databasestructuur is eenvoudig. De vooraf geprogrammeerde tabellen zijn onafhankelijk en hebben geen verbindingen met andere tabellen. De dynamische tabellen, die informatie bijhouden over berekeningen voor verschillende belastingscombinaties, zijn verbonden met de primaire sleutel van de tabel voor belastingscombinaties. Hierdoor kunnen de waarden voor een specifieke combinatie worden opgevraagd. Deze dynamische tabellen omvatten berekeningen voor interne krachten, sterkte-, stabiliteit- en doorbuigingscontroles, evenals tabellen met informatie over belastingen en steunpunten omdat deze waarden per combinatie verschillen. Daarnaast hebben alle tabellen die uitkomsten van berekeningen bevatten een kolom die het iteratienummer bijhoudt. Een aantal tabellen worden besproken in de uitleg van de code bij de corresponderende berekening in Secties 4.2 tot en met 4.5.

4.1.4 Code

Het programma kan databasehandelingen uitvoeren met behulp van geprogrammeerde functies. Om dit te doen, moet het programma eerst verbinding maken met de database. De code die dit realiseert, wordt weergegeven in Code 4.1 en is afkomstig van Glenn van Vugt, doctoraatsonderzoeker binnen de onderzoeksgroep CERG van UHasselt.

```
1  # Code from Glenn van Vugt
2  def dtb_connect(dtb_name: str):
3      # Verify dtb_name
4      if dtb_name is None:
5          print("None dtb_name")
6          return
7
8      # Try to make a connection
9      try:
10         con = sql.connect(dtb_name)
11     except sql.OperationalError:
12         return None
13
14     # Return output
15     return con
```

Code 4.1: Functie database connectie maken

In Code 4.1 wordt er gecontroleerd of er een connectie gemaakt kan worden. Indien dit lukt wordt de connectie gegeven en anders wordt er niks gegeven. Deze code moet bij elke databasehandeling altijd als eerste worden opgeroepen omdat deze connectie nodig is om verdere opdrachten uit te voeren.

Tabel creëren

Alle tabellen voor dynamische gegevensverwerking worden bij het opstarten van het programma leeg gemaakt. Code 4.2 toont de code om dit te doen voor de tabel voor de interne krachten en vervormingen die te zien is in Figuur 4.1.

```
1 def dtb_create_datapoints():
2     # Connect to database
3     con = dtb_connect("database.db")
4
5     # Verify connection
6     if con is None:
7         raise Exception("Connection to database failed")
8
9     # Drop table if exists
10    query = """DROP TABLE IF EXISTS analysis_datapoints;"""
11    with con:
12        con.execute(query)
13
14    # Create table in database
15    query = """CREATE TABLE analysis_datapoints (
16        dp_id    INTEGER NOT NULL UNIQUE,
17        position      REAL NOT NULL,
18        shear_0 REAL NOT NULL,
19        shear_1 REAL NOT NULL,
20        moment_0      REAL NOT NULL,
21        moment_1      REAL NOT NULL,
22        deflection     REAL NOT NULL,
23        combination_id INTEGER REAL NOT NULL,
24        iteration INTEGER,
25        FOREIGN KEY(combination_id) REFERENCES analysis_combinations
26            (combination_id),
27        PRIMARY KEY(dp_id AUTOINCREMENT));"""
28
29    with con:
30        con.execute(query)
31        con.commit()
32
33    # Close connection
34    con.close()
```

Code 4.2: Functie tabel datapoints creëren

In Code 4.2 wordt eerst een connectie met de database gemaakt en gecontroleerd. Vervolgens wordt de eerste opdracht (*query*) uitgevoerd, die de bestaande tabel verwijdert als deze al bestaat. Elke opdracht wordt uitgevoerd via de databaseconnectie (*con.execute(query)*). Na het verwijderen van de bestaande tabel wordt in een tweede opdracht de nieuwe tabel aangemaakt. In deze opdracht wordt voor elke kolom de sleutel en het bijbehorende gegevenstype (zoals *INTEGER* voor gehele getallen, *REAL* voor getallen met decimalen, en *TEXT* voor tekst) gespecificeerd. Ook wordt de primaire sleutel (*PRIMARY KEY*) en eventuele vreemde sleutels (*FOREIGN KEY*) aangeduid. Naast *con.execute(query)* wordt bij de tweede opdracht ook *con.commit()* gebruikt om alle wijzigingen permanent in de database op te slaan. Ten slotte wordt de connectie gesloten zodat er een nieuwe connectie kan worden gemaakt met een andere tabel.

Dit proces wordt herhaald voor elke tabel die dynamisch gegevens verwerkt. Hierbij verandert alleen de tabelnaam in de eerste opdracht en de volledige tweede opdracht op basis van de kolommen in de nieuwe tabel.

Data opslaan

Nadat de tabel gemaakt is, kan informatie in de tabel worden opgeslagen. Hiervoor wordt opnieuw een nieuwe functie voor gebruikt. De functie voor data in de tabel van interne krachten en vervormingen op te slaan wordt getoond in Code 4.3.

```
1 def dtb_insert_datapoints(data: list):
2     con = dtb_connect("database.db")
3
4     # Verify connection
5     if con is None:
6         Exception("Connection to database failed")
7
8     # Insert data
9     query = """INSERT INTO analysis_datapoints (position, shear_0,
10 shear_1, moment_0, moment_1, deflection, combination_id,
11 iteration) VALUES (?,?,?,?,?,?,?,?);"""
12
13     with con:
14         con.executemany(query, data)
15         con.commit()
16
17     con.close()
```

Code 4.3: Functie datapoints opslaan

In de functie van Code 4.3 wordt een lijst van datapunten (*data*) in de database opgeslagen. Deze lijst bevat tupels¹ waarin de waarden in dezelfde volgorde staan als in de opdracht is gespecificeerd. De waarden die in de tabel worden opgeslagen, corresponderen met de kolomsleutels van deze tabel. Voor deze functie wordt *con.executemany(query, data)* gebruikt om de opdracht uit te voeren waardoor alle tupels in één keer kunnen worden opgeslagen. Als elke tupel apart zou worden opgeslagen, zou elke keer een connectie moeten worden gemaakt, geverifieerd en gesloten, wat het programma aanzienlijk zou vertragen.

¹ Een tupel wordt gebruikt om reeks vaste waarden in een vaste volgorde op te slaan.

Deze functie is specifiek voor het uploaden van datapunten naar de tabel voor datapunten. Voor het opslaan van andere informatie in een andere tabel is een andere functie gemaakt waarbij de waarden opnieuw overeenkomen met de kolomsleutels van die tabel.

Data opvragen

Tot slot worden er functies gebruikt om data uit de database op te vragen. Code 4.4 toont een voorbeeld van een functie die datapunten uit de tabel van datapunten ophaalt. Deze functie kan verschillende soorten informatie opvragen voor diverse combinaties en iteraties, en binnen bepaalde intervalposities in de balk.

```
1 def dtb_query_datapoints_between(data: str, combination_id: int,
2     iteration: int, lower_bound: float, upper_bound: float):
3
4     # Connect to database
5     con = dtb_connect("database.db")
6
7     # Verify connection
8     if con is None:
9         Exception("Connection to database failed")
10
11    # Query data
12    query = f"SELECT {data} FROM analysis_datapoints WHERE
13    combination_id = ? AND iteration = ? AND position BETWEEN ? AND
14    ?;"
15
16    with con:
17        cursor = con.cursor()
18        cursor.execute(query, (combination_id, iteration,
19    lower_bound, upper_bound))
20        data = cursor.fetchall()
21
22    # Close connection
23    con.close()
24
25    # Return output
26    return data
```

Code 4.4: Functie data opvragen

De functie in Code 4.4 heeft de parameters *data*, *combination_id*, *iteration*, *lower_bound* en *upper_bound* nodig. *data* bevat de sleutels van de kolommen die opgevraagd moeten worden, zoals posities, momenten, en vervorming in het ene geval of posities en dwarskrachten in een ander geval. Met de *WHERE*-clausule wordt aangeduid voor welke combinatie en iteratie de data moet worden opgevraagd en met *BETWEEN ... AND ...* wordt het interval voor de positie gespecificeerd. In de code wordt een cursor aangemaakt op basis van de databaseconnectie. Deze cursor is nodig om een opdracht die data opvraagt uit de database uit te voeren.

Naast de filteropties in bovenstaande code zijn er andere functies beschikbaar die gebruikmaken van bijvoorbeeld *CASE WHEN ... THEN ... ELSE ...* om voorwaarden toe te voegen, zoals het vinden van de hoogste waarde tussen twee waarden. Daarnaast is het mogelijk de opgevraagde data te sorteren met *ORDER BY ...*

4.2 Analyse

Bij het analyseren van een balk is het programma flexibel. De gebruiker kan zowel isostatische als hyperstatische balken analyseren met het programma. Het programma kan rolopleggingen, scharnieren en inklemmingen modelleren. Daarnaast kan de gebruiker zoveel belastingsgevallen maken als hij wilt met een onbeperkt aantal belastingen in elk belastingsgeval. Voor de belastingen kan de gebruiker kiezen uit: puntlasten, momenten rond één punt en uniforme en niet-uniforme verdeelde belastingen.

In deze sectie wordt eerst uitgelegd hoe het programma een model maakt van alle bovengenoemde elementen. Daarna wordt de code uitgelegd die alle verschillende belastingscombinaties genereert, gevolgd door gedeelte over hoe een mesh wordt genereert door het programma. Hierna wordt de implementatie van de eindige-elementenmethode toegelicht en als slot wordt uitgelegd hoe alle data wordt opgeslagen in de database.

4.2.1 Model balk

Voordat een balk geanalyseerd kan worden, moet er een model van de balk worden opgesteld. Dit model bevat informatie zoals de lengte, het traagheidsmoment en de elasticiteitsmodulus van de balk. Daarnaast omvat het model het type en de positie van de steunpunten, de verschillende belastingsgevallen en het type, de waarde en de positie van de belastingen.

Het programma maakt gebruik van klassen om de informatie van steunpunten, belastingsgevallen en belastingen initieel bij te houden. Deze informatie wordt in het werkgeheugen van het programma bewaard totdat de analyse is uitgevoerd en alle gegevens in de database zijn opgeslagen. Na het opslaan van de gegevens wordt deze informatie uit het werkgeheugen verwijderd om de prestaties van het programma niet te vertragen.

Klassen voor steunpunten

Code 4.5 toont de verschillende klassen voor de steunpunten.

```
1  @dataclass
2  class Support:
3      position: float # [m] from the left end of the beam
4      visible: bool = field(init=False, default=True) # True if the
5                  support is visible in the plot
6
7  @dataclass
8  class RollingSupport(Support):
9      y_reaction: float = field(init=False, default=0) # [kN]
10
```

```

11 @dataclass
12 class HingeSupport(Support):
13     y_reaction: float = field(init=False, default=0) # [kN]
14
15 @dataclass
16 class FixedSupport(Support):
17     y_reaction: float = field(init=False, default=0) # [kN]
18     moment_reaction: float = field(init=False, default=0) # [kNm]

```

Code 4.5: Klassen voor steunpunten

Zoals te zien is in Code 4.5, zijn er vier klassen voor de steunpunten. De eerste is een overkoepelende klasse waarvan de andere drie klassen erven. Deze overkoepelende klasse bevat de positie van het steunpunt en een booleaanse operator² die aangeeft of het steunpunt zichtbaar is in de interface.

De klassen voor de roloplegging (*RollingSupport*) en het scharnier (*HingeSupport*) zijn identiek omdat er geen reactiekracht in de x-richting optreedt. Deze klassen hebben alleen een reactiekracht in de y-richting. De klasse voor de inklemming (*FixedSupport*) heeft naast een reactiekracht in de y-richting ook een reactiekracht tegen het moment.

Klassen voor belastingsgevallen

Code 4.6 toont de klassen voor het blijvende belastingsgeval en voor het veranderlijke belastingsgeval.

```

1 class LoadGroup:
2     _object_teller = 0 # Class variable to count the number of
3     objects
4
5     def __init__(
6         self,
7         name: str,
8         favourable_ULS: float,
9         unfavourable_ULS: float,
10        favourable_SLS: float,
11        unfavourable_SLS: float,
12        psi_0: float,
13        psi_1: float,
14        psi_2: float,
15        duration: str,
16        loads: list[Load],
17        type: int
18    ):
19        LoadGroup._object_teller += 1
20        self.id = (LoadGroup._object_teller)
21        self.name = name
22        self.favourable_ULS = favourable_ULS

```

² Een booleaanse operator heeft slechts twee waarden, namelijk waar of niet-waar.

```

22         self.unfavourable_ULS = unfavourable_ULS
23         self.favourable_SLS = favourable_SLS
24         self.unfavourable_SLS = unfavourable_SLS
25         self.psi_0 = psi_0
26         self.psi_1 = psi_1
27         self.psi_2 = psi_2
28         self.duration = duration
29         self.loads = loads
30         self.type = type
31
32     class PermanentLoadGroup(LoadGroup):
33         def __init__(
34             self,
35             name: str,
36             favourable_ULS: float,
37             unfavourable_ULS: float,
38             favourable_SLS: float,
39             unfavourable_SLS: float,
40         ):
41             super().__init__(
42                 name,
43                 favourable_ULS,
44                 unfavourable_ULS,
45                 favourable_SLS,
46                 unfavourable_SLS,
47                 None,
48                 None,
49                 None,
50                 "permanent",
51                 [],
52                 1,
53             )
54
55     class VariableLoadGroup(LoadGroup):
56         def __init__(
57             self,
58             name: str,
59             favourable_ULS: float,
60             unfavourable_ULS: float,
61             favourable_SLS: float,
62             unfavourable_SLS: float,
63             psi_0: float,
64             psi_1: float,
65             psi_2: float,
66             duration: str,
67         ):
68             super().__init__(
69                 name,
70                 favourable_ULS,

```

```

71         unfavourable_ULS ,
72         favourable_SLS ,
73         unfavourable_SLS ,
74         psi_0 ,
75         psi_1 ,
76         psi_2 ,
77         duration ,
78         [] ,
79         2 ,
80     )

```

Code 4.6: Klassen voor belastingsgevallen

In Code 4.6 is er één overkoepelende klasse voor de belastingsgevallen (*LoadGroup*) en twee klassen die daarvan erven. In een belastingsgeval worden de naam, de gunstige en ongunstige partiële factoren voor zowel de uiterste grenstoestand als de bruikbaarheidsgrenstoestand bijgehouden. Daarnaast worden de verschillende combinatiefactoren, de belastingsduurklasse van het belastingsgeval en een lijst met alle belastingen in dit belastingsgeval vastgelegd.

Het verschil tussen de klasse voor blijvende belastingsgevallen (*PermanentLoadGroup*) en die voor veranderlijke belastingsgevallen (*VariableLoadGroup*) is dat de laatste combinatiefactoren heeft en verschillende belastingsduurklassen kan hebben. Het belangrijkste doel van dit onderscheid is om de berekening van de belastingscombinaties te vergemakkelijken.

Klassen voor belastingen

De klassen voor de belastingen worden weergegeven in Code 4.7.

```

1  @dataclass()
2  class Load:
3      load_group_id: int # id of the load group where the load belongs
4                          to
5  @dataclass()
6  class VerticalForce(Load):
7      position: float # [m] from the left end of the beam
8      value: float # [kN]
9
10 @dataclass()
11 class Moment(Load):
12     position: float # [m] from the left end of the beam
13     value: float # [kNm]
14
15 @dataclass()
16 class DistributedLoad(Load):
17     start_position: float # [m] from the left end of the beam
18     end_position: float # [m] from the left end of the beam
19     value_1: float # [kN/m]
20     value_2: float # [kN/m]

```

Code 4.7: Klassen voor belastingen

Code 4.7 bevat opnieuw een overkoepelende klasse (*Load*) en drie afgeleide klassen. De overkoepelende klasse houdt alleen het unieke identificatienummer van het belastingsgeval bij waarin de belasting zich bevindt.

De klasse voor een puntlast (*VerticalForce*) en de klasse voor een moment (*Moment*) bevatten dezelfde informatie maar zijn opnieuw opgesplitst om latere functies eenvoudiger te maken. Deze twee klassen bevatten allebei één positie en één waarde. De klasse voor een verdeelde belasting (*DistributedLoad*) heeft twee posities en twee waarden. De eerste positie geeft het beginpunt aan, terwijl de tweede positie het eindpunt markeert. De bijbehorende waarden vertegenwoordigen respectievelijk de start- en eindwaarde van de belasting.

4.2.2 Belastingscombinaties

Voordat wordt uitgelegd hoe het programma belastingscombinaties genereert, wordt de klasse van een belastingscombinatie getoond en welke informatie het programma nodig heeft om de eindige-elementenmethode uit te voeren. De klasse voor de belastingscombinatie wordt getoond in Code 4.8.

```
1 class LoadCombination:
2     _object_teller = 0
3
4     def __init__(
5         self,
6         name: str,
7         criteria: str,
8         factors: list,
9     ):
10         LoadCombination._object_teller += 1
11         self.id = (LoadCombination._object_teller)
12         self.name = name
13         self.criteria = criteria
14         self.factors = factors
```

Code 4.8: Klasse voor belastingscombinatie

De klasse in Code 4.8 bevat een uniek identiteitsnummer, een naam, een aanduiding voor welke combinatie de belastingscombinatie is gemaakt en een lijst met factoren. Deze lijst met factoren bepaalt de waarde van de belastingen in de EEM. In deze lijst wordt elk belastingsgeval gerepresenteerd door een tuple. Deze tuple bevat twee waarden: de waarde van de partiële factor en de waarde van de combinatiefactor voor dit specifieke belastingsgeval in de combinatie. Er wordt een tuple gebruikt omdat de waarden onveranderlijk zijn en omdat tupels minder werkgeheugen in beslag nemen dan lijsten, wat de prestaties bevordert.

Stel er is een situatie met drie belastingsgevallen: het eigengewicht en twee veranderlijke belastingsgevallen waarbij $\psi_0 = 0,7$ voor het eerste veranderlijke belastingsgeval en $\psi_0 = 0,5$ voor het tweede veranderlijke belastingsgeval. De partiële factoren van de belastingsgevallen zijn zoals beschreven in Sectie 3.1. Dan zou de lijst met factoren voor twee belastingscombinaties in UGT-B er als volgt uit kunnen zien:

belastingscombinatie 1 = [(1,35; 1,00), (1,50; 1,00), (1,50; 0,70)]

belastingscombinatie 2 = [(1,35; 1,00), (1,50; 0,60), (1,50; 1,00)]

Voor de eerste belastingscombinatie is het eerste belastingsgeval het overheersende belastingsgeval, terwijl voor de tweede belastingscombinatie het tweede belastingsgeval het overheersende belastingsgeval is. Stel er zouden vier belastinggevallen zijn, dan zouden er vier tupels in de lijst van factoren zitten in plaats van drie nu.

Het programma genereert belastingscombinaties voor UGT-B, BGT-K, BGT-F en BGT-Q. Voor elke situatie maakt het programma een belastingscombinatie met alleen eigengewicht en, indien er andere belastingsgevallen zijn, alle andere mogelijke combinaties zoals beschreven in Sectie 3.1. De functie die verantwoordelijk is voor het oproepen van de benodigde functies om deze combinaties te genereren, is weergegeven in Code 4.9.

```
1 def generate_combinations(load_groups: List, criteria_list: List):
2     combs = []
3     lg_in_comb = []
4     for criteria in criteria_list:
5         if criteria == "ULS-P":
6             combs, lg_in_comb = self_weight_combination(load_groups,
7 "ULS-P", combs, lg_in_comb)
8             if len(load_groups) > 1:
9                 combs, lg_in_comb = variable_ULS_P(load_groups,
10 combs, lg_in_comb)
11             elif criteria == "SLS-C":
12                 combs, lg_in_comb = self_weight_combination(load_groups,
13 "SLS-C", combs, lg_in_comb)
14                 if len(load_groups) > 1:
15                     combs, lg_in_comb = variable_SLS_C(load_groups,
16 combs, lg_in_comb)
17                 elif criteria == "SLS-F":
18                     combs, lg_in_comb = self_weight_combination(load_groups,
19 "SLS-F", combs, lg_in_comb)
20                     if len(load_groups) > 1:
21                         combs, lg_in_comb = variable_SLS_F(load_groups,
22 combs, lg_in_comb)
23                     elif criteria == "SLS-Q":
24                         combs, lg_in_comb = self_weight_combination(load_groups,
25 "SLS-Q", combs, lg_in_comb)
26                         if len(load_groups) > 1:
27                             combs, lg_in_comb = variable_SLS_Q(load_groups,
28 combs, lg_in_comb)
29                 dtb_insert_load_group_in_combination(lg_in_comb)
30                 dtb_insert_combinations(combs)
31     return combs
```

Code 4.9: Functie die belastingscombinatie genereert

De functie in Code 4.9 initialiseert eerst twee lege lijsten. De eerste lijst wordt gebruikt om alle combinaties op te slaan, terwijl de tweede lijst de factoren van elk belastingsgeval in een specifieke combinatie bevat. De functie laat combinaties genereren voor elke situatie die de gebruiker heeft meegegeven. Hierbij worden de lijsten doorgegeven aan elke functie die combinaties maakt, waarbij continu informatie wordt toegevoegd aan deze lijsten. Nadat alle combinaties zijn gegenereerd, wordt de informatie in de database opgeslagen en worden de combinaties geretourneerd om gebruikt te worden in eindige-elementenmethode.

Voordat de code voor het genereren van belastingscombinaties wordt uitgelegd, wordt eerst de achterliggende gedachte toegelicht. Hiervoor wordt gebruik gemaakt van de belastingscombinaties in Tabel 3.3 uit Sectie 3.1.3. Er wordt aangenomen dat de combinatiefactor voor blijvende belastingsgevallen gelijk aan één is en dat de partiële factor voor veranderlijke belastingsgevallen steeds gelijk aan de ongunstige waarde is. Deze aanname is theoretisch niet correct maar aangezien de gunstige waarde voor de partiële factor voor veranderlijke belastingsgevallen gelijk aan nul is, kan het zelfde resultaat verkregen worden door de combinatiefactor gelijk aan nul te stellen. Alle belastingscombinaties uit Tabel 3.3 kunnen dan worden opgebouwd uit combinaties van de volgende drie combinatiesets:

combinatieset 1 = [(1,35; 1,00), (1,35; 1,00), (1,00; 1,00), (0,70; 0,00)]
 combinatieset 2 = [(1,35; 1,00), (1,35; 1,00), (0,70; 0,00), (1,00; 1,00)]
 combinatieset 3 = [(1,35; 1,00), (1,35; 1,00), (0,00; 0,00), (0,00; 0,00)]

In deze combinatiesets vertegenwoordigt elk koppel een belastingsgeval. Voor blijvende belastingsgevallen (de eerste twee koppels) zijn de waarden gelijk aan de ongunstige en gunstige partiële factor en voor veranderlijke belastingsgevallen (de laatste twee koppels) zijn deze gelijk aan de mogelijke waarde van de combinatiefactor, inclusief nul. Het aantal combinatiesets hangt af van het aantal veranderlijke belastingsgevallen, in dit geval twee. Daarnaast moet er altijd een combinatieset zijn waarin alle veranderlijke belastingsgevallen gunstig zijn, oftewel gelijk aan 0, wat resulteert in drie combinatiesets in dit geval. De belastingscombinaties uit worden gevonden door voor elke combinatieset alle mogelijke unieke combinaties te maken waarbij telkens één waarde van elk koppel wordt genomen:

Lijst 1:

combinatie 1 = [1,35; 1,35; 1,00; 0,70]
 combinatie 2 = [1,35; 1,35; 1,00; 0,00]
 combinatie 3 = [1,35; 1,00; 1,00; 1,00]
 combinatie 4 = [1,35; 1,00; 1,00; 0,00]
 combinatie 5 = [1,00; 1,35; 1,00; 0,70]
 combinatie 6 = [1,00; 1,35; 1,00; 0,00]
 combinatie 7 = [1,00; 1,00; 1,00; 0,70]
 combinatie 8 = [1,00; 1,00; 1,00; 0,00]

Lijst 2:

- combinatie 9 = [1,35; 1,35; 0,70; 1,00]
 combinatie 10 = [1,35; 1,35; 0,00; 1,00]
 combinatie 11 = [1,35; 1,00; 0,70; 1,00]
 combinatie 12 = [1,35; 1,00; 0,00; 1,00]
 combinatie 13 = [1,00; 1,35; 0,70; 1,00]
 combinatie 14 = [1,00; 1,35; 0,00; 1,00]
 combinatie 15 = [1,00; 1,00; 0,70; 1,00]
 combinatie 16 = [1,00; 1,00; 0,00; 1,00]

Lijst 3:

- combinatie 17 = [1,35; 1,35; 0,00; 0,00]
 combinatie 18 = [1,35; 1,00; 0,00; 0,00]
 combinatie 19 = [1,00; 1,35; 0,00; 0,00]
 combinatie 20 = [1,00; 1,00; 0,00; 0,00]

De belastingscombinaties worden nu gevonden door de partiële factor van blijvende belastingsgevallen toe te voegen aan de waarden die deze belastingsgevallen vertegenwoordigen en de ongunstige partiële factor van veranderlijke belastingsgevallen toe te voegen aan de waarden die deze belastingsgevallen vertegenwoordigen in de combinatie:

- belastingscombinatie 1 = [(1,35; 1,00), (1,35; 1,00), (1,50; 1,00), (1,50; 0,70)]
 belastingscombinatie 2 = [(1,35; 1,00), (1,35; 1,00), (1,50; 1,00), (1,50; 0,00)]
 belastingscombinatie 3 = [(1,35; 1,00), (1,00; 1,00), (1,50; 1,00), (1,50; 0,70)]
 belastingscombinatie 4 = [(1,35; 1,00), (1,00; 1,00), (1,50; 1,00), (1,50; 0,00)]
 belastingscombinatie 5 = [(1,00; 1,00), (1,35; 1,00), (1,50; 1,00), (1,50; 0,70)]
 belastingscombinatie 6 = [(1,00; 1,00), (1,00; 1,00), (1,50; 1,00), (1,50; 0,00)]
 belastingscombinatie 7 = [(1,00; 1,00), (1,00; 1,00), (1,50; 1,00), (1,50; 0,70)]
 belastingscombinatie 8 = [(1,00; 1,00), (1,00; 1,00), (1,50; 1,00), (1,50; 0,00)]
 belastingscombinatie 9 = [(1,35; 1,00), (1,35; 1,00), (1,50; 0,70), (1,50; 1,00)]
 belastingscombinatie 10 = [(1,35; 1,00), (1,35; 1,00), (1,50; 0,00), (1,50; 1,00)]
 belastingscombinatie 11 = [(1,35; 1,00), (1,00; 1,00), (1,50; 0,70), (1,50; 1,00)]
 belastingscombinatie 12 = [(1,35; 1,00), (1,00; 1,00), (1,50; 0,00), (1,50; 1,00)]
 belastingscombinatie 13 = [(1,00; 1,00), (1,35; 1,00), (1,50; 0,70), (1,50; 1,00)]
 belastingscombinatie 14 = [(1,00; 1,00), (1,00; 1,00), (1,50; 0,00), (1,50; 1,00)]
 belastingscombinatie 15 = [(1,00; 1,00), (1,00; 1,00), (1,50; 0,70), (1,50; 1,00)]
 belastingscombinatie 16 = [(1,00; 1,00), (1,00; 1,00), (1,50; 0,00), (1,50; 1,00)]
 belastingscombinatie 17 = [(1,35; 1,00), (1,35; 1,00), (1,50; 0,00), (1,50; 0,00)]
 belastingscombinatie 18 = [(1,35; 1,00), (1,00; 1,00), (1,50; 0,00), (1,50; 0,00)]
 belastingscombinatie 19 = [(1,00; 1,00), (1,35; 1,00), (1,50; 0,00), (1,50; 0,00)]
 belastingscombinatie 20 = [(1,00; 1,00), (1,00; 1,00), (1,50; 0,00), (1,50; 0,00)]

In het programma is dit proces opgesplitst in twee functies. De eerste functies maakt de combinatiesets met de verschillende waarden voor de partiële en combinatiefactoren. De tweede functie maakt de combinaties en vult deze aan met de nodige partiële of combinatiefactor. De functie voor de eerste stap voor UGT-B wordt getoond in Code 4.10.

```

1  def create_combinationsets_ULS_P(load_groups: List):
2      combinationsets = []
3
4      # Create combinations with different leading load groups
5      for lg in load_groups:
6          if isinstance(lg, PermanentLoadGroup):
7              continue
8
9          leading_lg = lg
10         factors = []
11         # Assign for each load group a factor to the combination
12         for _lg in load_groups:
13             if isinstance(_lg, PermanentLoadGroup):
14                 factors.append((_lg.unfavourable_ULS, _lg.favourable_ULS))
15
16             elif isinstance(_lg, VariableLoadGroup):
17                 if _lg == leading_lg:
18                     factors.append((1, 1))
19                 else:
20                     factors.append((_lg.psi_0, 0))
21
22         combinationsets.append(factors)
23
24         # Combinations with only permanent load groups
25         factors = []
26         for _lg in load_groups:
27             if isinstance(_lg, PermanentLoadGroup):
28                 factors.append((_lg.unfavourable_ULS, _lg.favourable_ULS))
29
30             elif isinstance(_lg, VariableLoadGroup):
31                 factors.append((0, 0))
32         combinationsets.append(factors)
33
34         return combinationsets

```

Code 4.10: Functie combinatiesets voor UGT-B

De eerste stap in de functie van Code 4.10 is het aanmaken van een lege lijst waarin alle combinatiesets worden opgeslagen. Vervolgens wordt voor elk veranderlijk belastingsgeval een aparte combinatieset gemaakt. Het programma doorloopt alle belastingsgevallen en identificeert het eerste veranderlijke belastingsgeval. Daarna doorloopt het programma opnieuw alle belastingsgevallen, waarbij het elk geval een waarde toekent. Dit gebeurt op basis van het type belastingsgeval: blijvend of veranderlijk. Voor de veranderlijke belastingsgevallen maakt het programma een onderscheid of het belastingsgeval overheersend is in deze combinatieset of niet.

Nadat de eerste combinatieset is gemaakt, blijft het programma nieuwe combinatiesets genereren totdat alle veranderlijke belastingsgevallen als overheersend zijn beschouwd. Ten slotte doorloopt het programma alle belastingsgevallen nog een laatste keer, waarbij alleen de blijvende waarden worden toegekend en de veranderlijke waarden op nul worden gezet.

De functie die de belastingscombinaties maakt voor UGT-B wordt getoond in Code 4.11.

```

1  def variable_ULS_P(load_groups: List, combinations: List,
2      lg_in_combination: List):
3      combinationsets = create_combinationsets_ULS_P(load_groups)
4
5      unique_combinations = set()
6      for combinationset in combinationsets:
7          combination = itertools.product(*combinationset)
8          unique_combinations.update(combination)
9
10     unique_combinations = list(unique_combinations)
11
12     for i, combination in enumerate(unique_combinations):
13         lc = LoadCombination("ULS-P " + str(i+1), "ULS-P", [])
14         combinations.append(lc)
15         for j, factor in enumerate(combination):
16             if isinstance(load_groups[j], PermanentLoadGroup):
17                 lc.factors.append((factor, 1.0))
18                 lg_in_combination.append((factor, 1.0, load_groups[j]
19 ].id, lc.id))
20             elif isinstance(load_groups[j], VariableLoadGroup):
21                 lc.factors.append((load_groups[j].unfavourable_ULS,
22 factor))
23                 lg_in_combination.append((load_groups[j].
24 unfavourable_ULS, factor, load_groups[j].id, lc.id))
25
26     return combinations, lg_in_combination

```

Code 4.11: Functie belastingscombinaties maken voor UGT-B

De functie in Code 4.11 begint met het aanroepen van de functie die de combinatiesets genereert. Vervolgens wordt de *product* functie uit de ingebouwde bibliotheek *itertools* gebruikt om alle mogelijke combinaties binnen elke combinatieset te maken. Deze combinaties worden opgeslagen in een set³ genaamd *unique_combinations*.

³ Een set is vergelijkbaar met een lijst maar kan geen duplicaten bevatten.

Nadat alle combinaties zijn gegenereerd, wordt elke combinatie afzonderlijk doorlopen. Voor elke combinatie wordt een object aangemaakt van de klasse *LoadCombination*. Vervolgens wordt voor elke waarde in de combinatie een koppel gemaakt met de benodigde partiële of combinatiefactor. Ten slotte worden al deze objecten toegevoegd aan de overkoepelende lijst van combinaties en geretourneerd, zoals besproken in de functie in Code 4.9.

In Code 4.10 en 4.11 staan enkel de functies voor UGT-B. Voor BGT-K, BGT-F en BGT-Q zijn er gelijkaardige functies maar met andere partiële en combinatiefactoren die overeenkomen met de formules voor deze situaties.

4.2.3 Mesh

Het laatste dat moet worden bepaald voordat de eindige-elementenmethode kan worden uitgevoerd, zijn de knooppunten die de balk in elementen verdelen. Deze verzameling knooppunten wordt de mesh genoemd. De mesh bepaalt ook de posities waar alle controles bij het dimensioneren van een balk worden uitgevoerd. De grootte en kwaliteit van de mesh zijn cruciaal voor het programma. Een gedetailleerde mesh, met veel knooppunten, zorgt voor een grotere matrix en meer data, wat de nauwkeurigheid vergroot maar ook de berekeningen vertraagt. Aan de andere kant kan een te grove mesh onnauwkeurig zijn, waardoor belangrijke gegevens worden gemist en de berekeningen de werkelijkheid niet goed weergeven. Het is dus belangrijk om een balans te vinden tussen nauwkeurigheid en efficiëntie bij het bepalen van de mesh.

Bij het maken van de mesh houdt het programma rekening met twee aspecten. Ten eerste moeten bepaalde vaste posities altijd in de mesh worden opgenomen. Deze posities komen overeen met de locaties van belastingen en steunen tegen kip en, bij betonbalken, de posities waar de langswapening begint of eindigt. Ten tweede houdt het programma rekening met het aantal elementen waarin de gebruiker elke overspanning wil verdelen. De functie die de mesh bepaalt, wordt weergegeven in Code 4.12.

```
1 def determine_mesh(intervals: List, divisions: int, load_positions:
    List, stability_supports: List, reinforcement_positions: List):
2     mesh = set()
3
4     for interval in intervals: # interval is between to supports
5         mesh_interval = set()
6         fixed_positions = determine_fixed_positions(interval,
    stability_supports, load_positions, reinforcement_positions)
7         L = interval[1] - interval[0]
8         dx = L / divisions
9
10        # Iterate through pairs of fixed positions and determine
    positions between them
11        for pos1, pos2 in zip(fixed_positions, fixed_positions[1:]):
12            positions = determine_between_positions(pos1, pos2, dx)
13            mesh_interval.update(positions)
14        mesh.update(mesh_interval)
15    return sorted(mesh)
```

Code 4.12: Functie die mesh bepaalt

De functie in Code 4.12 heeft een lijst met tuples nodig, waarbij deze tuples de begin- en eindposities van een overspanning aangeven (*intervals*). Daarnaast heeft de functie het aantal elementen waarin elk interval moet worden verdeeld (*divisions*), de posities van de belastingen (*load_positions*), de posities van de steunen tegen kip (*stability_supports*) en de posities van de wapening (*reinforcement_positions*) nodig. De functie begint met het maken van een set waarin alle posities worden opgeslagen. Er wordt een set gebruikt zodat een positie niet dubbel in de mesh voorkomt. Vervolgens wordt voor elke overspanning een aparte mesh (*mesh_interval*) gecreëerd. Hiervoor worden eerst de vaste posities binnen dit interval bepaald, inclusief de start-, midden- en eindposities van het interval, evenals de posities van de belastingen, de steunpunten tegen kip en eventueel de wapening. Daarna wordt de lengte van elk element (*dx*) bepaald. Ten slotte worden er, indien nodig, extra posities tussen de vaste posities geplaatst als de afstand tussen deze posities groter is dan de lengte van een element. Dit gebeurt met de functie *determine_between_positions*. Uiteindelijk worden alle posities in de mesh geplaatst en wordt de mesh gesorteerd zodat de posities van klein naar groot staan.

De functie die de vaste posities in een interval bepaalt (*determine_fixed_positions*) is eenvoudig. Eerst worden de start-, midden- en eindposities in een set geplaatst. Vervolgens wordt voor elke waarde in de lijsten met posities van belastingen, steunpunten tegen kip en wapening gecontroleerd of deze binnen het interval vallen. Als dat het geval is, wordt de positie aan de set toegevoegd en anders niet. Daarna wordt de set gesorteerd en geretourneerd. De functie die de extra posities tussen twee vaste posities bepaalt is ingewikkelder en wordt getoond in Code 4.13.

```

1 def determine_between_positions(pos_1: float, pos_2: float, dx:
    float):
2     divisions = int((pos_2 - pos_1) // dx + 1)
3     dx = (pos_2 - pos_1) / divisions
4     positions = [round(pos_1 + i * dx, 2) for i in range(divisions)]
5     return positions + [pos_2]
```

Code 4.13: Functie die posities tussen vaste posities bepaalt

De functie in Code 4.13 heeft twee posities nodig (*pos_1* en *pos_2*) en de lengte van een element (*dx*). Eerst bepaalt de functie het aantal elementen waarin het interval tussen de twee posities kan worden verdeeld. Daarna berekent de functie een nieuwe lengte voor de elementen zodat het interval in gelijke stukken wordt verdeeld. Ten slotte worden alle posities binnen het interval bepaald op basis van het aantal elementen en de nieuwe lengte van elk element.

4.2.4 Eindige-elementenmethode

De code voor de eindige-elementenmethode voert het proces uit zoals beschreven in Sectie 3.1.2. Deze code is grotendeels geschreven door mijn begeleider, Glenn van Vugt, met enkele aanpassingen van mijn kant om deze geschikt te maken voor het programma en om een automatische koppeling met de database te realiseren.

De code bestaat uit vier fasen: In de eerste fase worden de mesh, belastingen en steunpunten verzameld en omgezet in het juiste formaat zodat deze geschikt zijn voor de analyse. In de tweede fase worden, afhankelijk van de combinaties, de juiste factoren toegepast op de belastingen. De derde fase voert de analyse uit, waarbij voornamelijk matrixbewerkingen worden uitgevoerd om

de interne krachten en doorbuigingen voor elke positie in de mesh te berekenen. In de laatste fase worden alle gegevens opgeslagen in de database.

Fase 1: Omzetting van informatie

De hoofdfunctie van de eerste fase wordt getoond in Code 4.14. Deze code ontvangt alle informatie die nodig is voor de EEM en zet deze in het juiste formaat. Dit juiste formaat kan een matrix zijn maar kan ook een lijst zijn waarbij de posities van steunpunten en belastingen overeenkomen met de posities in de mesh.

```
1 def calculate_internal_forces(mesh, E, I, supports, loads,
2   load_groups, combinations, iteration):
3     # Geometric properties - mesh
4     nodes = np.array([[x, 0] for x in mesh])
5     elements = np.array([[x, x + 1] for x in range(len(nodes) - 1)])
6
7     # Create beam
8     beam = Beam(E, I, nodes, elements)
9
10    # Turn loads into vector
11    loadings = adjust_loads(loadings, mesh)
12    # Adjust array for supports
13    adjust_supports(beam.supports, supports, mesh)
14    # Calculate loads
15    vec_loadings = calculate_loads(loadings, beam)
16
17    # Calculate combinations
18    calculate_combinations(vec_loadings, beam, combinations,
19                           load_groups, supports, iteration)
```

Code 4.14: Functie die alle data verzamelt

De functie in Code 4.14 creëert eerst twee matrices op basis van de mesh. De eerste matrix, *nodes*, bevat de posities van alle knooppunten in de mesh. De tweede matrix, *elements*, bevat per rij de twee knooppunten die elk element in de mesh vormen. Nadat deze matrices zijn gegenereerd, maakt de functie een object van de klasse *Beam* aan en roept vervolgens een aantal functies aan die de informatie omzetten naar het juiste formaat.

De functie zelf voert geen directe omzettingen uit. Deze omzettingen worden uitgevoerd door de functies die zij aanroept. Zo worden bijvoorbeeld alle benodigde matrices aangemaakt tijdens het aanmaken van een object van de klasse *Beam*. De initialisatiefunctie van deze klasse is weergegeven in Code 4.15.

```
1 class Beam:
2     def __init__(self, E, I, nodes, elements):
3         # Properties - to be defined
4         self.E = E
5         self.I = I
6         self.nodes = nodes.astype(float)
```

```

7         self.elements = elements.astype(int)
8
9         # Degrees of freedom per node
10        self.dof = 2
11
12        # Boundary conditions
13        self.supports = np.ones_like(nodes).astype(int)
14
15        # Loading
16        self.point_loads = np.zeros_like(nodes, float)
17        self.distributed_loads = np.zeros([len(elements), 2])
18        self.equivalent_loads = np.zeros_like(nodes)
19
20        # Results
21        self.forces = np.zeros([len(elements), 2 * self.dof])
22        self.displacements = np.zeros([len(elements), 2 * self.dof])

```

Code 4.15: Klasse *Beam*

De functie in Code 4.15 maakt gebruik van de functies *np.ones_like*, *np.zeros_like* en *np.zeros*. Deze functies creëren matrices van een bepaalde grootte, gevuld met waarden van één of nul, afhankelijk van hun toepassing. De afmetingen van de rijen en kolommen zijn afhankelijk van het beoogde gebruik van deze matrices. Deze matrices worden in latere functies verder bewerkt op basis van de belastingen en steunpunten.

De functie *adjust_loads*, die wordt aangeroepen in Code 4.14, bepaalt de positie van elke belasting in de mesh en creëert een tuple met informatie voor elk type belasting. Deze informatie omvat het type belasting (puntlast, moment of verdeelde belasting), de waarde van de belasting en de positie van het knooppunt in de lijst van knooppunten. Deze positie bepaalt welke rij in de matrix moet worden aangepast in de functie *calculate_loads*.

De functie *adjust_supports* past de matrix van steunpunten (*self.supports*) in het *Beam*-object aan op basis van de eigenschappen van de steunpunten van de balk. Deze matrix heeft oorspronkelijk twee kolommen en evenveel rijen als knooppunten. Wanneer een roloplegging of scharnier overeenkomt met een knooppunt, wordt de waarde in de eerste kolom van die rij op nul gezet, wat betekent dat de verticale verplaatsing wordt geblokkeerd. Bij een inklemming worden de waarden in beide kolommen op nul gezet, waardoor ook de rotatie wordt geblokkeerd.

De functie *calculate_loads* creëert per belasting twee matrices op basis van de informatie uit de functie *adjust_loads*. De eerste matrix heeft twee kolommen en evenveel rijen als knooppunten, en geeft de belasting op elk knooppunt weer. Als er een puntlast is, wordt de waarde in de eerste kolom gelijkgesteld aan de belasting en bij een moment wordt de waarde in de tweede kolom gelijkgesteld aan het moment. De tweede matrix heeft ook twee kolommen, maar evenveel rijen als er elementen zijn. Deze matrix representeert de verdeelde belastingen. Als er een verdeelde belasting over meerdere elementen plaatsvindt, wordt deze belasting opgesplitst, waarbij de eerste kolom de eerste waarde van de verdeelde belasting bevat en de tweede kolom de tweede waarde. Deze twee matrices worden op hun beurt in een rij gestopt en dan geretourneerd als *vec_loadings*.

Fase 2: Combinaties

Nadat alle matrices zijn aangemaakt en alle posities van de steunpunten en belastingen in deze matrices zijn gevonden, kunnen de combinaties worden uitgerekend. Hiervoor moeten de waarden van de belastingen worden aangepast op basis van de factoren van de combinatie. De functie die dit doet wordt getoond in Code 4.16.

```
1 def calculate_combinations(vec_loadings, beam, combinations,
2   load_groups, supports, iteration):
3
4     for i, combination in enumerate(combinations):
5         analysis = copy.deepcopy(beam)
6         k = 0 # Counter for vec_loadings
7         for j, [gamma, psi] in enumerate(combination.factors):
8             for load in range(len(load_groups[j].loads)):
9                 # Add factors to point loads
10                analysis.point_loads = np.add(analysis.point_loads,
11 gamma * psi * vec_loadings[k].point_loads)
12                # Add factors to distributed loads
13                analysis.distributed_loads = np.add(analysis.
14 distributed_loads, gamma * psi * vec_loadings[k].
15 distributed_loads)
16                k += 1
17
18            # Fase 3
19            analysis.run_analysis()
20            # Fase 4
21            insert_data(analysis, combination, load_groups, supports,
22 iteration)
```

Code 4.16: Functie dat combinaties uitrekent

De functie in Code 4.16 doorloopt elke belastingscombinatie. Voor elke belastingscombinatie wordt een object genaamd *analysis* gemaakt, wat een kopie is van het oorspronkelijke *Beam*-object. Hierdoor neemt het nieuwe object alle eigenschappen over zonder dat het oorspronkelijke object wordt aangepast tijdens de berekeningen. Bij elke combinatie wordt ook een variabele *k* op nul gezet. Vervolgens worden alle koppels van partiële en combinatiefactoren in een combinatie doorlopen, waarbij er net zoveel koppels zijn als belastingsgevallen. Voor elk koppel wordt elke belasting in het corresponderende belastingsgeval doorlopen en worden de factoren toegepast. De variabele *k* houdt bij hoeveel belastingen al zijn verwerkt, zodat steeds de juiste matrices van *vec.loadings* worden gebruikt. Zonder deze variabele zou bijvoorbeeld bij twee belastingsgevallen met elk twee belastingen de tweede belasting in het eerste belastingsgeval de factoren krijgen van het tweede belastingsgeval, en zouden beide belastingen in het tweede belastingsgeval niet eens worden meegerekend.

Fase 3: Analyse

Nadat alle belastingen met de juiste factor zijn toegekend aan het object *analysis* kan de eindigelementenmethode worden uitgevoerd. Deze methode wordt getoond in de functie van Code 4.17.

```

1 def run_analysis(self):
2     # Number of nodes
3     n_nodes = len(self.nodes)
4
5     # Number of files
6     n_elements = len(self.elements)
7
8     # Degrees of freedom
9     n_dof = n_nodes * self.dof
10
11     # Length
12     dl = self.nodes[self.elements[:, 1], :] - self.nodes[self.
elements[:, 0], :]
13     l = np.sqrt((dl ** 2).sum(axis=1))
14
15     # Allocate stiffness matrices
16     Kloc = np.zeros([n_elements, 2*self.dof, 2*self.dof])
17     Kglob = np.zeros([n_dof, n_dof])
18
19     for i in range(n_elements):
20         # DOF
21         aux = self.dof * self.elements[i, :]
22         idx = np.r_[aux[0]:aux[0] + self.dof, aux[1]:aux[1] +
self.dof]
23
24         # Local stiffness matrix
25         l_elem = l[i]
26         Kloc[i] = self.E * self.I / l_elem**3 * np.array([
27             [12,          6*l_elem, -        12,          6*l_elem],
28             [6*l_elem,   4*l_elem**2,   -6*l_elem,   2*l_elem
**2],
29             [-12,         -6*l_elem,    12,          -6*l_elem],
30             [6*l_elem,   2*l_elem**2,   -6*l_elem,   4*l_elem**2]
31         ])
32
33         # Assemble
34         Kglob[np.ix_(idx, idx)] += Kloc[i]
35
36     # Distributed loads
37     eq_load_elem = np.zeros([len(self.elements), 2 * self.dof])
38     for i in range(n_elements):
39         l_elem = l[i]
40         pi = self.distributed_loads[i, 0]
41         pf = self.distributed_loads[i, 1]
42
43         eq_load_elem[i, 0] = l_elem * (21 * pi + 9 * pf) / 60
44         eq_load_elem[i, 1] = l_elem * (l_elem * (3 * pi + 2 * pf
)) / 60

```

```

45         eq_load_elem[i, 2] = l_elem * (9 * pi + 21 * pf) / 60
46         eq_load_elem[i, 3] = l_elem * (l_elem * (-2 * pi - 3* pf
47     )) / 60
48
49     # Point loads
50     self.equivalent_loads = copy.deepcopy(self.point_loads)
51     for i in range(n_elements):
52         self.equivalent_loads[self.elements[i, 0], 0] +=
eq_load_elem[i, 0]
53         self.equivalent_loads[self.elements[i, 0], 1] +=
eq_load_elem[i, 1]
54         self.equivalent_loads[self.elements[i, 1], 0] +=
eq_load_elem[i, 2]
55         self.equivalent_loads[self.elements[i, 1], 1] +=
eq_load_elem[i, 3]
56
57     # Solution
58     free_dof = self.supports.flatten().nonzero()[0]
59     Ksolve = Kglob[np.ix_(free_dof, free_dof)]
60     Fsolve = self.equivalent_loads.flatten()[free_dof]
61     usolve = np.linalg.solve(Ksolve, Fsolve)
62
63     # Compute full matrices and vectors
64     u = self.supports.astype(float).flatten()
65     u[free_dof] = usolve
66     u = u.reshape(n_nodes, self.dof)
67
68     # Global vectors to calculate reaction forces
69     u_global = np.concatenate(u)
70     eq_loads_global = np.concatenate(self.equivalent_loads)
71     supports_global = np.concatenate(self.supports)
72
73     # Reaction forces
74     self.reaction_forces = []
75     for i in range(n_nodes * self.dof):
76         if supports_global[i] == 0:
77             self.reaction_forces.append(round(np.dot(Kglob[i],
u_global) - eq_loads_global[i], 2))
78
79     # Internal forces and displacements
80     u_elem = np.concatenate((u[self.elements[:, 0]], u[self.
elements[:, 1]]), axis=1)
81     for i in range(n_elements):
82         self.forces[i] = np.dot(Kloc[i], u_elem[i]) -
eq_load_elem[i]
83         self.displacements[i] = u_elem[i]

```

Code 4.17: Functie die de analyse uitvoert

De functie in Code 4.17 volgt hetzelfde proces als beschreven in Sectie 3.1.2. Eerst wordt op regel 24 de lokale stijfheidsmatrix voor elk element gegenereerd. Op regel 34 worden alle lokale matrices samengevoegd tot één globale matrix. Vervolgens wordt op de regels 37 tot en met 54 de krachtenmatrix opgesteld op basis van de (equivalente) krachten op de knooppunten. Na diverse matrixbewerkingen wordt de verplaatsingsvector berekend, waarna de reactiekrachten, interne krachten en verplaatsing van elk knooppunt worden bepaald.

Fase 4: Data opslag

Ten slotte wordt alle data opgeslagen in de database. Hierbij wordt op elke positie van de mesh de dwarskrachten, de momenten en de doorbuiging voor elke combinatie opgeslagen. Daarnaast ook de waarden van de belastingen en de reactiekrachten in elke combinatie opgeslagen.

4.3 Beton

Voor de berekeningen van beton in het programma moet de gebruiker de dimensies, de sterkteklassen van het beton, de betondekking en de kruipfactor invoeren. De laatste twee waarden kan de gebruiker zelf kiezen, maar hij kan ze ook laten bepalen zoals uitgelegd in Sectie 3.2.2. Daarnaast moet de gebruiker altijd een wapeningsconfiguratie specificeren, dit bevat onder-, boven- en dwarskrachtwapening. Met deze gegevens kan het programma de balk controleren, de wapening voor deze dimensies optimaliseren en zowel de wapening als de dimensies optimaliseren voor een zo efficiënt mogelijk materiaalgebruik.

In deze sectie wordt eerst uitgelegd hoe het programma de ingevoerde gegevens analyseert om de grootheden van de doorsnede te bepalen. Vervolgens wordt beschreven hoe de controle op basis van sterkte, spanningen, scheurwijdte en doorbuiging wordt uitgevoerd. Tot slot worden de optimalisaties van de wapening en doorsnede toegelicht.

4.3.1 Doorsnede analyse

Het eerste wat het programma doet bij het ontwerpen van betonnen balken is het analyseren van de doorsnede. Dit omvat het berekenen van de geplaatste wapening en het bepalen van de overgang van de primaire en secundaire wapening aan de onder- en bovenkant van de doorsnede, evenals het berekenen van de grootheden van de doorsnede.

Analyse wapening

Zoals eerder vermeld, moet de gebruiker altijd de onder-, boven- en dwarskrachtwapening specificeren. Het programma analyseert de wapening door op elke positie in de mesh de equivalente diameter, de afstand tot het zwaartepunt en de oppervlakte van de onder- en bovenwapening te bepalen. Voor de dwarskrachtwapening worden de diameter en de hoeveelheid wapening per tussenafstand bepaald. Al deze gegevens worden vervolgens opgeslagen in de database.

Figuur 4.2 toont de database waarin deze informatie is opgeslagen. De informatie in deze figuur is gebaseerd op een voorbeeld van een balk van vijf meter lang met een betondekking van 40 cm en een roloplegging in het midden. Over de volledige lengte van de balk zijn aan de onder- en bovenkant twee wapeningsstaven met een diameter van zestien millimeter geplaatst. In het midden van de balk is een extra wapeningsstaaf van achttien millimeter toegevoegd. Voor de

dwarskrachtwapening zijn beugels met een diameter van acht millimeter en een tussenafstand van 230 millimeter geplaatst.

Table: concrete_provided_reinforcement											
	provided_reinforcement_id	position	dia_down	d_down	As_prov_down	dia_up	d_up	As_prov_up	stirrup_dia	Asw_s_prov	iteration
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	0.0	16	56.0	402	16	56.0	402	8	0.437	1
2	2	0.42	16	56.0	402	16	56.0	402	8	0.437	1
3	3	0.83	16	56.0	402	16	56.0	402	8	0.437	1
4	4	1.25	16	56.0	402	16	56.0	402	8	0.437	1
5	5	1.5	16	56.0	402	16.72	56.39	656	8	0.437	1
6	6	2.0	16	56.0	402	16.72	56.39	656	8	0.437	1
7	7	2.5	16	56.0	402	16.72	56.39	656	8	0.437	1
8	8	3.0	16	56.0	402	16.72	56.39	656	8	0.437	1
9	9	3.5	16	56.0	402	16.72	56.39	656	8	0.437	1
10	10	3.75	16	56.0	402	16	56.0	402	8	0.437	1
11	11	4.17	16	56.0	402	16	56.0	402	8	0.437	1
12	12	4.58	16	56.0	402	16	56.0	402	8	0.437	1
13	13	5.0	16	56.0	402	16	56.0	402	8	0.437	1

Figuur 4.2: Tabel voor analyse wapening

Uit Figuur 4.2 blijkt duidelijk dat de wapening op verschillende posities is geanalyseerd. De informatie over de onderwapening (kolommen drie tot en met vijf) is constant omdat de wapening overall hetzelfde is. De informatie over de bovenwapening (kolommen zes tot en met acht) varieert tussen anderhalve meter en drieënhalve meter vanwege de extra wapeningsstaaf van achttien millimeter. De gegevens van de dwarskrachtwapening (kolommen negen en tien) blijven constant.

Grootheden doorsnede bepalen

Na de analyse van de wapening worden de grootheden van de doorsnede berekend. Dit omvat de nuttige hoogte, de hoeveelheid primaire en secundaire wapening, het kritische moment, en de betondrukhoogte en traagheidsmoment voor zowel de gescheurde als ongescheurde doorsnede. Deze berekeningen worden uitgevoerd voor alle posities in de mesh en voor elke belastingscombinatie, met behulp van de functie in Code 4.18.

```

1 def determine_section_parameters(
2     h: int,
3     b: int,
4     fctm: float,
5     Es: int,
6     Ec: int,
7     provided_reinforcement: dict,
8     internal_forces: dict,
9     iteration: int,
10 ):
11     database = []
12     for key, data in internal_forces.items():
13         for i in range(len(data)):
14             position = data[i][0]

```

```

15     Med = data[i][2]
16     reinforcement = provided_reinforcement[i]
17     if Med >= 0: # Tension below
18         d = h - reinforcement[3]
19         As1 = reinforcement[4]
20         d2 = reinforcement[6]
21         As2 = reinforcement[7]
22     else: # Tension above
23         d = h - reinforcement[6]
24         As1 = reinforcement[7]
25         d2 = reinforcement[3]
26         As2 = reinforcement[4]
27     alpha = calculate_alpha(Es, Ecm)
28     x_uncracked = int(calculate_x_uncracked(b, h, alpha, As1
29 , As2, d, d2))
30     I_uncracked = int(calculate_I_uncracked(b, h, alpha, As1
31 , As2, d, d2, x_uncracked))
32     Mcr = int(calculate_Mcr(h, fctm, I_uncracked,
33 x_uncracked))
34     x_cracked = int(calculate_x_cracked(b, alpha, As1, As2,
35 d, d2))
36     I_cracked = int(calculate_I_cracked(b, alpha, As1, As2,
37 d, d2, x_cracked))
38     database.append((position, h, b, d, d2, As1, As2,
39 x_uncracked, I_uncracked, Mcr, x_cracked, I_cracked, key,
40 iteration))
41     dtb_insert_concrete_section_parameters(database)

```

Code 4.18: Functie grootheden doorsnede bepalen

De functie in Code 4.18 doorloopt elke belastingscombinatie en elke positie in de mesh. Het teken van het optredende moment bepaalt of er trek boven of onder is, wat de plaatsing van de primaire en secundaire wapening bepaalt. Vervolgens worden voor elke positie de betondrukhoogtes, traagheidsmomenten en het kritische moment berekend met specifieke functies die de formules voor deze grootheden bevatten. Tot slot worden alle berekende gegevens opgeslagen in de database, zoals getoond in Figuur 4.3.

Table: concrete_section_parameters																
	concrete_section_parameters_id	position	h	b	d	d2	As1	As2	x_uncracked	I_uncracked	Mcr	x_cracked	I_cracked	combination_id	iteration	
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	
1		1	0.0	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1
2		2	0.42	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1
3		3	0.83	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1
4		4	1.25	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1
5		5	1.5	500	350	444.0	56.39	402	656	248	3886096201	44.0	69	381571879	1	1
6		6	2.0	500	350	443.61	56.0	656	402	251	3886146148	45.0	87	584766537	1	1
7		7	2.5	500	350	443.61	56.0	656	402	251	3886146148	45.0	87	584766537	1	1
8		8	3.0	500	350	443.61	56.0	656	402	251	3886146148	45.0	87	584766537	1	1
9		9	3.5	500	350	444.0	56.39	402	656	248	3886096201	44.0	69	381571879	1	1
10		10	3.75	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1
11		11	4.17	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1
12		12	4.58	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1
13		13	5.0	500	350	444.0	56.0	402	402	250	3829223296	44.0	70	381282993	1	1

Figuur 4.3: Tabel grootheden doorsnede

In Figuur 4.3 wordt alle informatie voor elke positie in de mesh bijgehouden. De breedte en hoogte van de doorsnede blijven constant over de volledige lengte van de balk. De andere grootheden variëren echter afhankelijk van de geplaatste wapening en het teken van het moment, wat duidelijk zichtbaar is tussen anderhalve en drieënhalve meter. Deze figuur toont enkel de grootheden van de eerste combinatie en iteratie. In werkelijkheid bevat de tabel de waarden voor elke combinatie en iteratie.

4.3.2 Controle doorsnede

Op elke positie van de mesh wordt de doorsnede gecontroleerd. Hierbij wordt er gecontroleerd op sterkte, toelaatbare spanningen, scheurwijdte en doorbuiging. Indien één van deze criteria niet voldoet, voldoet de volledige balk niet.

Sterkte

Het programma controleert de sterkte van een betonnen balk door de geplaatste onder- en bovenwapening en de dwarskrachtwapening te vergelijken met de corresponderende benodigde wapening. Deze controle vindt plaats op elke positie in de mesh en voor elke belastingscombinatie volgens UGT-B.

Vooraleer de sterkte wordt gecontroleerd, wordt de minimum langs- en dwarskrachtwapening berekend. Deze minimum wapening wordt eenmaal berekend voor de volledige balk en niet voor elke positie in de mesh. Voor de nuttige hoogte, die gebruikt wordt voor deze berekeningen, wordt de grootse waarde van de nuttige hoogte gebruikt aangezien dit de grootste minimum wapening vereist en dus het meest conservatieve is. Deze maximale nuttige hoogte wordt in de database gevonden met de functie getoond in Code 4.19.

```
1 def dtb_query_max_d(combination_ids: list, iteration: int):
2     con = dtb_connect("sources/database.db")
3     if con is None:
4         raise Exception("Connection to database failed")
5     placeholders = ", ".join(["?" for _ in range(len(combination_ids))])
6     query = f"""
7     SELECT d
8     FROM concrete_section_parameters
9     WHERE combination_id IN ({placeholders}) AND iteration = ?
10    ORDER BY d DESC
11    LIMIT 1
12    """
13    with con:
14        cur = con.cursor()
15        parameters = combination_ids + [iteration]
16        cur.execute(query, parameters)
17        result = cur.fetchone()
18    con.close()
19    return result
```

Code 4.19: Functie dat de maximale nuttige hoogte vindt

De functie in Code 4.19 maakt en controleert eerst de verbinding met de database. Vervolgens zet de functie een lijst van alle unieke combinatienummers (*combination_ids*) om in een formaat dat in de query kan worden gebruikt. Daarna selecteert de functie de grootste waarde van de nuttige hoogte en retourneert deze. Dit gebeurt door de waarden aflopend te sorteren (regel 10) en de eerste (hoogste) waarde te selecteren (regel 11).

Nadat de hoeveelheid minimum langs- en dwarskrachtwapening is bepaald, wordt elke positie gecontroleerd op sterkte. Dit gebeurt met de functie in Code 4.20.

```

1  def control_section_As(
2      interval: tuple,
3      h: int,
4      b: int,
5      fck: float,
6      fyk: int,
7      internal_forces: dict,
8      max_moments: dict,
9      provided_reinforcement: dict,
10     As_min: float,
11     Asw_s_min: float,
12     iteration: int,
13 ):
14     check_list_As = []
15     database = []
16     # Check for each combination
17     for key, internal_forces in internal_forces.items():
18         Med_max = abs(max_moments[key][1])
19         # Check for each position
20         for i in range(len(internal_forces)):
21             position = internal_forces[i][0]
22             Med = internal_forces[i][2]
23             Ved = abs(internal_forces[i][1])
24             if Med >= 0:
25                 d = h - provided_reinforcement[i][3]
26             else:
27                 d = h - provided_reinforcement[i][6]
28             Asw_s_prov = provided_reinforcement[i][9]
29             As_prov_down = provided_reinforcement[i][4]
30             As_prov_up = provided_reinforcement[i][7]
31
32             As_needed_down, As_needed_up, Asw_s_needed =
33             calculate_needed_As(interval, position, b, d, fck, fyk, Ved, Med,
34             Med_max, As_min, Asw_s_min)
35
36             check_Asw_s = Asw_s_prov >= Asw_s_needed
37             check_As_down = As_prov_down >= As_needed_down
38             check_As_up = As_prov_up >= As_needed_up
39             check_As = check_Asw_s and check_As_down and check_As_up
40             check_list_As.append(check_As)

```

```

39         database.append([position, As_prov_down, As_needed_down,
As_prov_up, As_needed_up, Asw_s_prov, Asw_s_needed, key,
iteration])
40         updated_database = update_Asw_s_needed(database, len(
internal_forces))
41         database.extend(updated_database)
42
43     # Check if all the combinations are safe
44     check_As = all(check_list_As)
45     dtb_insert_concrete_control_As(database)
46     return check_As

```

Code 4.20: Functie voor controle sterkte beton

De functie in Code 4.20 begint met het aanmaken van twee lege lijsten: één om alle controles bij te houden en één om de gegevens op te slaan die in de database moeten worden bewaard. Vervolgens doorloopt de functie elke belastingscombinatie en elke positie binnen deze combinatie. Hierbij worden de benodigde grootheden voor het berekenen van de vereiste wapening bepaald. Op regel 32 worden de benodigde hoeveelheden langswapening onderaan de doorsnede (*As_needed_down*), langswapening bovenaan de doorsnede (*As_needed_up*) en dwarskrachtwapening (*Asw_s_needed*) berekend. Vervolgens wordt gecontroleerd of de geplaatste wapening groter dan of gelijk aan de vereiste wapening is. Tot slot worden alle gegevens in de database opgeslagen en wordt geretourneerd of de wapening voldoet.

De functie die de benodigde wapening berekend op regel 32 van Code 4.20 wordt getoond in Code 4.21.

```

1  def calculate_needed_As(
2      interval: tuple,
3      position: float,
4      b: int,
5      d: float,
6      fck: int,
7      fyk: int,
8      Ved: float,
9      Med: float,
10     Med_max: float,
11     As_min: float,
12     Asw_s_min: float,
13 ):
14     sigma_s_assumption = round(fyk / 1.15, 0)
15     while True:
16         # Shear reinforcement
17         if position < interval[0] + d / 1000 or position > interval
[1] - d / 1000:
18             Asw_s_needed = 0
19         else:
20             Asw_s_needed = max(round(calculate_Asw_s(Ved, d, fyk),
3), Asw_s_min)

```

```

21     # Longitudinal reinforcement
22     extra_As = calculate_extra_As(Ved, abs(Med), Med_max, d, fyk
    )
23
24     if Med >= 0: # Tension reinforcement at the bottom
25         As, sigma_s = calculate_As(b, d, fck, Med)
26         As_needed_down = max(int(As + extra_As), As_min)
27         As_needed_up = As_min
28     else: # Tension reinforcement at the top
29         As, sigma_s = calculate_As(b, d, fck, abs(Med))
30         As_needed_down = As_min
31         As_needed_up = max(int(As + extra_As), As_min)
32     # Check assumption
33     if sigma_s_assumption == sigma_s:
34         break
35     else:
36         sigma_s_assumption = sigma_s
37
38     return As_needed_down, As_needed_up, Asw_s_needed

```

Code 4.21: Functie dat benodigde wapening bepaalt

De functie in Code 4.21 begint met het aannemen van een waarde voor de spanning in de primaire wapening, omdat deze afhankelijk is van de rek in het staal, wat niet vooraf bekend is. Deze waarde wordt aanvankelijk gelijkgesteld aan de rekenwaarde van de treksterkte van de wapening. Dit is waarom een While-lus wordt gebruikt, zodat deze aanname kan worden geverifieerd.

In de lus wordt eerst de benodigde dwarskrachtwapening berekend. Deze dwarskrachtwapening hoeft echter niet te worden bepaald voor een afstand gelijk aan de nuttige hoogte vanaf het dagvlak van een oplegging, wat de if-voorwaarde op regel 17 verklaart. Als de wapening niet binnen deze afstand valt, wordt het maximum van de berekende en de minimale dwarskrachtwapening genomen.

Vervolgens wordt de langswapening berekend. Eerst wordt de extra langswapening als gevolg van de dwarskracht berekend (regel 22). Daarna worden de onder- en bovenwapening berekend, waarbij de ene de primaire en de andere de secundaire wapening is, afhankelijk van het teken van het moment. De hoeveelheid primaire wapening is het maximum van de berekende benodigde wapening plus de extra door de dwarskracht en de minimale hoeveelheid langswapening. De secundaire wapening is altijd gelijk aan de minimale hoeveelheid langswapening.

Tot slot wordt gecontroleerd of de aangenomen spanning in de primaire wapening correct is. Als dit het geval is, wordt de While-lus beëindigd en worden de benodigde hoeveelheden wapening geretourneerd. Anders wordt de aangenomen spanning gelijk gesteld aan de berekende spanning en wordt de lus herhaald totdat een correcte waarde is gevonden.

De functie *update_Asw_s_needed* op regel 40 in Code 4.20 wijzigt de waarden die binnen de afstand van de nuttige hoogte van de rand van een steunpunt vallen, welke oorspronkelijk nul zijn. Deze functie past deze waarden aan naar de laatste waarde na of voor dit punt, afhankelijk van of het voor of na het steunpunt ligt. Als het voor het steunpunt is, worden alle waarden in het interval gelijkgesteld aan de laatste berekende waarde. Als het na het steunpunt is, worden alle waarden gelijkgesteld aan de eerstvolgende berekende waarde.

Spanning

De spanning in het beton, in de primaire wapening en in de secundaire wapening worden gecontroleerd voor BGT-K, BGT-F en BGT-Q. Code 4.22 toont de functie die deze controle uitvoert.

```
1 def control_section_stresses(  
2     fck: int,  
3     fyk: int,  
4     alpha: float,  
5     internal_forces: dict,  
6     section_parameters: dict,  
7     criteria: str,  
8     iteration: int,  
9 ):  
10     max_concrete, max_steel = max_stresses(fck, fyk, criteria)  
11     check_list_stresses = []  
12     database = []  
13     extra_As_needed = []  
14     for key, internal_forces in internal_forces.items():  
15         for i in range(len(internal_forces)):  
16             position = internal_forces[i][0]  
17             Med = internal_forces[i][2]  
18             Mcr = section_parameters[key][i][10]  
19             d = section_parameters[key][i][4]  
20             d2 = section_parameters[key][i][5]  
21             if abs(Med) < Mcr: # Uncracked section  
22                 x = section_parameters[key][i][8]  
23                 I = section_parameters[key][i][9]  
24             else: # Cracked section  
25                 x = section_parameters[key][i][11]  
26                 I = section_parameters[key][i][12]  
27             sigma_c, sigma_s1, sigma_s2 = calculate_stresses(d, d2,  
28 x, I, alpha, abs(Med))  
29             check_concrete = sigma_c <= max_concrete  
30             check_steel1 = sigma_s1 <= max_steel  
31             check_steel2 = sigma_s2 <= max_steel  
32             check_stresses = check_concrete and check_steel1 and  
33 check_steel2  
34             check_list_stresses.append(check_stresses)  
35             if not check_stresses:  
36                 extra_As_needed.append(position)  
37                 database.append([position, sigma_c, sigma_s1, sigma_s2,  
38 key, iteration])  
39  
40     check_stresses = all(check_list_stresses)  
41     dtb_insert_concrete_control_stresses(database)  
42     return check_stresses, extra_As_needed
```

Code 4.22: Functie voor controle spanningen beton

De functie in Code 4.22 bepaalt eerst de maximaal toelaatbare spanningen, afhankelijk van het type combinatie. Vervolgens worden drie lijsten aangemaakt: de eerste om alle controles op te slaan, de tweede voor de informatie die in de database moet worden opgeslagen, en de derde voor de optimalisatie van de wapening, die verder wordt besproken in Sectie 4.3.3. Daarna doorloopt de functie opnieuw alle posities in de mesh voor alle combinaties. Vervolgens wordt bepaald of de doorsnede op deze positie gescheurd is of niet. Afhankelijk hiervan worden de betondrukhoogte en het traagheidsmoment gekozen. Deze grootheden worden gebruikt om de spanningen te berekenen, die vervolgens worden vergeleken met de maximaal toelaatbare spanningen. Tot slot wordt de informatie in de database opgeslagen en wordt de uitkomst van de controle geretourneerd.

Scheurwijdte

De functie voor de controle van de scheurwijdte heeft dezelfde structuur als de functies in Code 4.20 en Code 4.22. Het verschil is dat in plaats van het berekenen van spanningen zoals in Code 4.22, bij de scheurwijdte de scheurwijdte wordt berekend en gecontroleerd, zoals uitgelegd in Sectie 3.2.6.

Doorbuiging

De functie voor de controle van de doorbuiging volgt ook dezelfde structuur als de functies in Code 4.20 en Code 4.22. Echter, voor de doorbuiging moeten zowel de gescheurde als niet-gescheurde toestanden worden berekend met behulp van de eindige-elementenmethode. Dit gebeurt tweemaal: één keer met het traagheidsmoment van de gescheurde toestand en één keer met het traagheidsmoment van de niet-gescheurde toestand. Nadat deze berekeningen zijn uitgevoerd, wordt de controle uitgevoerd aan de hand van de formules zoals uitgelegd in Sectie 3.2.7.

4.3.3 Optimalisatie wapening

De gebruiker heeft de flexibiliteit om de wapening te optimaliseren op basis van sterkte, spanning in BGT-K, BGT-F of BGT-Q, scheurwijdte, doorbuiging, of een combinatie van controles. Het programma genereert altijd één wapeningsconfiguratie die voldoet aan alle geselecteerde controles op elke positie in de mesh. De optimalisatiefunctie voor de wapening wordt getoond in Code 4.23.

```
1 def optimize_reinforcement(self, optimize_criteria: list, b: int, h:
  int, reinforcement: dict):
2     # Fase 1
3     iteration, Asw_s_min, As_min = self.optimize_analyse(b, h,
  reinforcement, self.criteria, optimize_criteria, self.combinations)
4     while True:
5         # Fase 2
6         needed_As = {}
7         start_needed_As = self.create_start_needed_As(b, h, As_min,
  Asw_s_min, False)
8         start_needed_As_SLS_Q = self.create_start_needed_As(b, h,
  As_min, Asw_s_min, True)
9         if "ULS-P" in optimize_criteria:
10             self.control_strength(b, h, iteration)
11             needed_As["ULS"] = dtb_query_As_needed(iteration)
```

```

12         if "STRESSES SLS-C" in optimize_criteria:
13             start_needed_As_stresses_SLS_C = copy.deepcopy(
start_needed_As)
14             needed_As["stresses_SLS_C"] = self.
optimize_reinforcement_stresses("SLS-C",
start_needed_As_stresses_SLS_C, iteration)
15         if "STRESSES SLS-F" in optimize_criteria:
16             start_needed_As_stresses = copy.deepcopy(start_needed_As
)
17             needed_As["stresses_SLS_F"] = self.
optimize_reinforcement_stresses("SLS-F", start_needed_As_stresses
, iteration)
18         if "STRESSES SLS-Q" in optimize_criteria:
19             start_needed_As_stresses = copy.deepcopy(
start_needed_As_SLS_Q)
20             needed_As["stresses_SLS_Q"] = self.
optimize_reinforcement_stresses("SLS-Q", start_needed_As_stresses
, iteration)
21         if "CRACK WIDTH SLS-Q" in optimize_criteria:
22             start_needed_As_crack_width = copy.deepcopy(
start_needed_As_SLS_Q)
23             needed_As["crack_width"] = self.
optimize_reinforcement_crack_width(start_needed_As_crack_width,
iteration)
24         if "DEFLECTION SLS-Q" in optimize_criteria:
25             start_needed_As_deflections = copy.deepcopy(
start_needed_As_SLS_Q)
26             needed_As["deflections"] = self.
optimize_reinforcement_deflections(start_needed_As_deflections,
iteration)
27         final_needed_As = self.determine_needed_As(needed_As, [0, 1,
2, 3]) # Indexes = [position, As_down_needed, As_up_needed,
Asw_s_needed]
28         # Fase 3
29         d = dtb_query_min_d(self.combination_ids["ULS-P"], iteration
)
30         reinforcement = optimize_reinforcement(self.L_beam, self.b,
self.c, d, self.material["fck"], self.As_min, final_needed_As,
self.Asw_s_min)
31         # Fase 4
32         iteration = self.optimize_analyse(b, h, reinforcement, self.
criteria, optimize_criteria, self.analysis.combinations)
33         check = self.control_section(optimize_criteria, iteration)
34         if check:
35             break
36         return reinforcement

```

Code 4.23: Functie dat wapening optimaliseert

Zoals te zien is in Code 4.23, bestaat de functie uit vier fasen. In de eerste fase worden alle voorbereidingen getroffen om de berekeningen uit te kunnen voeren. In de tweede fase wordt voor elke controle de benodigde wapening berekend voor elke positie in de mesh, en wordt de minimale hoeveelheid wapening bepaald op basis van de resultaten van elke controle. In fase drie wordt de wapening berekend op basis van deze waarden. Tot slot wordt in fase vier gecontroleerd of de wapening effectief voldoet. Deze vier fasen worden hierna meer in detail besproken.

Fase 1: Voorbereiding

In de eerste fase wordt enkel de functie *optimize_analyse* opgeroepen. Deze functie analyseert de balk op basis van de gegeven breedte, hoogte en wapening. Vervolgens analyseert de functie de wapening en berekent de functie voor elk soort combinatie de doorsnede-eigenschappen, zoals uitgelegd in Sectie 4.3.1. Daarna berekent de functie de minimum langs- en dwarskrachtwapening. Indien de wapening geoptimaliseerd moet worden voor doorbuiging, voert de functie analyses uit voor zowel de gescheurde als niet-gescheurde toestand. Tot slot retourneert de functie het iteratienummer waarop dit allemaal is gebeurd zodat deze informatie kan worden opgevraagd in de volgende fasen.

Fase 2: Bepaling hoeveelheid wapening

De twee fase start met het aanmaken van een woordenboek. In deze woordenboek zal voor elke controle een lijst van waarde van nodige wapening in functie van de positie worden bewaard. Na het aanmaken van de woordenboek wordt de functie *create_start_needed_As* tweemaal opgeroepen. Eenmaal voor SLS-Q en een andermaal voor alle andere soorten combinaties. Deze functie wordt getoond in Code 4.24.

```
1 def create_start_needed_As(self, b: int, h: int, As_min: int,
2   Asw_s_min: float, SLS_Q: bool):
3     needed_As = {}
4     data = []
5     alpha = self.alpha_SLS_Q if SLS_Q else self.alpha
6     for position in self.analysis.mesh:
7         h = h
8         b = b
9         d = h - (self.c + 10 + 32/2)
10        d2 = self.c + 10 + 12/2
11        As1 = As_min
12        As2 = As_min
13        x_uncracked, I_uncracked, Mcr, x_cracked, I_cracked =
14        calculate_section_parameters(b, h, alpha, As1, As2, d, d2, self.
15        material["fctm"])
16        data.append([Asw_s_min, position, h, b, d, d2, As1, As2,
17        x_uncracked, I_uncracked, Mcr, x_cracked, I_cracked])
18        needed_As["envelope"] = data
19    return needed_As
```

Code 4.24: Functie die startlijst voor wapening creëert

De functie in code 4.24 maakt een bibliotheek met daarin een lijst van sublijsten van waarden die worden aangenomen om de optimalisatie te starten. Elke sublijst vertegenwoordigt de doorsnede van een positie. In deze sublijst wordt er een waarde vastgesteld voor de nuttige hoogte (regel 8) en de afstand tot het zwaartepunt van de secundaire wapening (regel 9). Deze aannames hoeven niet exact te zijn, omdat deze waarden in fase vier tijdens de controle worden herberekend op basis van de gevonden wapening. Voor de primaire en secundaire wapening wordt de minimale langswapening aangenomen. Vervolgens worden de doorsnede-eigenschappen voor deze waarden berekend. Uiteindelijk worden alle waarden, inclusief de dwarskrachtwapening die gelijk is aan de minimale dwarskrachtwapening, per positie verzameld in de sublijst. De volgorde van deze sublijst (zie regel 13) is belangrijk voor de volgende functies. Deze bibliotheek wordt op het einde van de functie geretourneerd.

Hierna wordt voor elke controle waarvoor de optimalisatie moet plaatsvinden een aparte lijst met sublijsten gemaakt, zoals te zien is in de if-voorwaarden in Code 4.23. Elke sublijst bestaat uit vier elementen: [positie; wapening onder; wapening boven; dwarskrachtwapening]. De hoofdlijst bevat evenveel sublijsten als er posities zijn, waarbij de hoeveelheden wapening in elke sublijst overeenkomen met de benodigde wapening op die positie voor dat specifieke type wapening.

Als de optimalisatie gebaseerd is op sterkte, wordt deze lijst samengesteld door eerst de sterktecontrole uit te voeren (regel 10), waarbij de benodigde wapening in de database wordt opgeslagen. Vervolgens worden deze waarden uit de database gehaald (regel 11). Voor deze sterktecontrole wordt de bibliotheek uit code 4.24 niet gebruikt.

Voor de controles op spanningen, scheurwijdte en doorbuiging wordt deze bibliotheek wel gebruikt. Hierbij wordt eerst een kopie van de bibliotheek gemaakt om te voorkomen dat de originele waarden worden gewijzigd, zodat bij gebruik voor meerdere controles de tweede controle niet start met de waarden verkregen uit de eerste controle. De functie die wordt gebruikt om de wapening te bepalen op basis van spanningen (*optimize_reinforcement_stresses*) wordt getoond in Code 4.25.

```

1  def optimize_reinforcement_stresses(self, criteria: str,
2      needed_As_stresses: dict, iteration: int):
3      internal_forces = {}
4      internal_forces["envelope"] = dtb_query_internal_forces_envelope
5      (self.combination_ids[criteria], 0, self.L_beam, iteration)
6      alpha = self.alpha_SLS_Q if criteria == "SLS-Q" else self.alpha
7      while True:
8          check_stresses, extra_As_needed = control_section_stresses(
9              fck=self.material["fck"],
10             fyk=self.fyk,
11             alpha=alpha,
12             criteria=criteria,
13             internal_forces=internal_forces,
14             section_parameters=needed_As_stresses,
15             iteration=iteration,
16         )
17         # If the stresses are within the limits, break the loop
18         if check_stresses:
19             break

```

```

18         for key, data in needed_As_stresses.items():
19             for row in data:
20                 position = row[1]
21                 if position in extra_As_needed:
22                     As1 = row[6] + 100
23                     x_uncracked, I_uncracked, Mcr, x_cracked,
I_cracked = calculate_section_parameters(row[3], row[2], alpha,
As1, row[7], row[4], row[5], self.material["fctm"])
24                     # Update parameters
25                     row[6] = As1
26                     row[8] = x_uncracked
27                     row[9] = I_uncracked
28                     row[10] = Mcr
29                     row[11] = x_cracked
30                     row[12] = I_cracked
31
32         As_needed_stresses = self.determine_needed_As(needed_As_stresses
, [1, 6, 7, 0]) # Indexes = [position, As1_needed, As2_needed,
Asw_s_needed]
33         As_needed_stresses = self.convert_Asi_to_Asx(As_needed_stresses,
internal_forces) # Indexes = [position, As_down_needed,
As_up_needed, Asw_s_needed]
34         return As_needed_stresses

```

Code 4.25: Functie dat wapening bepaald op basis van spanning

De functie in Code 4.25 gebruikt de functie *dtb_query.internal_forces_envelope* om op elke positie in de database het grootste moment te vinden voor alle combinaties voor dit soort combinaties. Daarna start een while-lus die blijft lopen totdat de controle op spanningen is geslaagd. Als de controle niet slaagt, wordt de primaire wapening verhoogd met 100 mm² op elke positie waar de controle niet is geslaagd. Deze posities worden bepaald tijdens de spanningscontrole, zoals te zien is in Code 4.22. Als een positie faalt, wordt deze toegevoegd aan de lijst *extra_As_needed*, die aan het einde van de functie wordt geretourneerd.

Na het verhogen van de primaire wapening worden de doorsnede-eigenschappen opnieuw berekend met de nieuwe waarde van de primaire wapening. De lus herhaalt zich totdat elke positie voldoende wapening heeft en de controle slaagt. Vervolgens worden de positie, de hoeveelheid primaire wapening, de hoeveelheid secundaire wapening en de hoeveelheid dwarskrachtwapening uit de bibliotheek gehaald en omgezet naar het formaat [positie; primaire wapening; secundaire wapening; dwarskrachtwapening]. Hiervoor wordt de functie op regel 32 gebruikt. Het is belangrijk om de juiste indexen mee te geven, die overeenkomen met de indexen van de sublijsten in Code 4.24. Bijvoorbeeld, positie heeft index één⁴ en primaire wapening heeft index zes.

Tot slot worden alle verkregen sublijsten omgezet naar het formaat [positie, wapening onder, wapening boven, dwarskrachtwapening]. Dit gebeurt door de functie *convert_Asi_to_Asx* op regel 33. Deze functie doet deze omzetting op basis van het teken van het moment op elke positie.

⁴ Positie staat eigenlijk op de tweede plaats in de sublijst maar in programmeertaal begint men te tellen vanaf nul.

De functies die de wapening bepalen voor de scheurwijdte en doorbuiging hebben dezelfde opzet als die voor de spanningen. Nadat de benodigde wapening voor alle controles is berekend, wordt op regel 27 van Code 4.23 de lijst *final_needed_As* samengesteld. Deze lijst bevat opnieuw evenveel sublijsten als er posities zijn, waarbij elke sublijst de grootste benodigde hoeveelheid wapening voor elke controle bevat. Op deze manier wordt voor elke positie de hoeveelheid wapening bepaald die voldoet aan alle controles.

Fase 3: Berekening wapening

Nadat de benodigde wapening is berekend, wordt de daadwerkelijke wapening bepaald. Eerst wordt de kleinste nuttige hoogte in de huidige wapeningsconfiguratie gevonden (regel 29 van Code 4.23). Vervolgens wordt de wapening berekend (regel 30). De functie die deze berekening uitvoert, wordt getoond in Code 4.26.

```
1 def optimize_reinforcement(  
2     L_beam: float,  
3     b: int,  
4     c: int,  
5     d: float,  
6     fck: int,  
7     As_min: float,  
8     As_needed: list,  
9     Asw_s_min: float,  
10 ):  
11     stirrup_dia = 10  
12     while True:  
13         reinforcement_down = determine_reinforcement_longitudinal(  
14             L_beam, b, c, fck, stirrup_dia, As_min, As_needed, 1)  
15         reinforcement_up = determine_reinforcement_longitudinal(  
16             L_beam, b, c, fck, stirrup_dia, As_min, As_needed, 2)  
17         shear_reinforcement, stirrup_dia_used =  
18             determine_shear_reinforcement(L_beam, As_needed, Asw_s_min, d)  
19         if stirrup_dia_used == stirrup_dia:  
20             break  
21         stirrup_dia = stirrup_dia_used  
22  
23     reinforcement = {  
24         "longitudinal_down": reinforcement_down,  
25         "longitudinal_up": reinforcement_up,  
26         "shear": shear_reinforcement}  
27     return reinforcement
```

Code 4.26: Functie die wapening bepaald

De functie in Code 4.26 begint met een aanname voor de diameter van de beugels. Vervolgens start een while-lus waarin de onderwapening, bovenwapening en dwarskrachtwapening worden berekend. Als de berekende diameter van de beugels overeenkomt met de aanvankelijke aanname, wordt de lus beëindigd. Indien dit niet het geval is, wordt de lus herhaald met de nieuwe diameter. Uiteindelijk retourneert de functie een bibliotheek met de berekende wapening.

De langs- en dwarskrachtwapening worden in bovenstaande functie bepaald door een andere functies op te roepen. De functie die de langswapening bepaalt wordt getoond in Code 4.27.

```
1 def determine_reinforcement_longitudinal(  
2     L_beam: float,  
3     b: int,  
4     c: int,  
5     fck: int,  
6     stirrup_dia: int,  
7     As_min: float,  
8     needed_As: list,  
9     index: int  
10 ):  
11     reinforcement = []  
12     intervals = determine_intervals(L_beam, needed_As, As_min, index)  
13     max_As = max(needed_As, key=lambda x: x[index])[index]  
14     diameter, amount = find_diameter_and_amount(b, c, stirrup_dia,  
15     max_As, True)  
16     reinforcement.append((0, L_beam, 2, diameter, "continuous"))  
17     As_continuous = 2 * PI * (diameter / 2) ** 2  
18     reduced_b = b - (2 * diameter + 2 * 40)  
19     for interval in intervals:  
20         As_needed = interval[2]  
21         if As_needed <= As_continuous:  
22             continue  
23         As_needed_extra = As_needed - As_continuous  
24         diameter, amount = find_diameter_and_amount(reduced_b, c,  
25         stirrup_dia, As_needed_extra, False)  
26         anchorage_length = round(calculate_anchorage_length(diameter  
27         , fck)/1000, 2)  
28         length_interval = interval[1] - interval[0]  
29         start_bar, end_bar = determine_start_end_bar(interval,  
30         anchorage_length, L_beam)  
31         reinforcement.append((start_bar, end_bar, amount, diameter,  
32         "extra"))  
33     return reinforcement
```

Code 4.27: Functie die langswapening bepaald

De functie in Code 4.27 wordt gebruikt voor zowel de boven- als onderwapening. Voor onderwapening wordt *index* ingesteld op 1 en voor bovenwapening op 2. Dit komt overeen met de index die de verschillende soorten wapening hebben in de sublijst van *needed_As*.

De functie begint met het creëren van intervallen waarbij de waarden in een interval allemaal gelijk zijn aan de minimum langswapening of allemaal groter. Elk interval heeft het formaat: [begin positie; eind positie; wapening], waarbij de wapening de grootste benodigde hoeveelheid binnen dat interval is. Vervolgens wordt de continue langswapening bepaald, bestaande uit twee

staven die over de volledige lengte van de balk aanwezig zijn. Deze staven worden berekend op basis van de grootste benodigde wapening met de functie *find_diameter_and_amount*. Als er meer dan twee staven nodig zijn, wordt het aantal beperkt tot twee. Op deze manier kan de maximale benodigde wapening gehaald worden door extra toe staven toe voegen waar nodig, zonder dat de balk overall overgedimensioneerd is.

Nadat de diameter van deze staven is bepaald, wordt de oppervlakte van de wapening berekend en hoeveel breedte in de doorsnede nog beschikbaar is voor extra wapening. Voor elk interval wordt de benodigde wapening vergeleken met de reeds aanwezige continue wapening. Als de aanwezige wapening voldoende is, is er geen extra wapening nodig. Als extra wapening wel nodig is, wordt berekend hoeveel extra wapening nodig is (regel 23), en vervolgens worden de diameter en hoeveelheid van deze extra wapening bepaald. De minimale verankeringslengte voor deze extra wapening wordt ook berekend. Als het interval niet groot genoeg is, wordt extra lengte toegevoegd op basis van de benodigde verankeringslengte. Uiteindelijk retourneert de functie de berekende wapening.

De functie *find_diameter_and_amount* die de optimale diameter en aantal wapeningsstaven vindt wordt getoond in Code 4.28.

```
1 def find_diameter_and_amount(  
2     b: int,  
3     c: int,  
4     stirrup_dia: int,  
5     As_needed: float,  
6     continuous: bool,  
7 ):  
8     best_diameter = best_amount = 0  
9     smallest_difference_As = max(As_needed, 30)# Must be at least 30  
10    (As for 1 bar diameter 6 = 28.27)  
11    for diameter in BAR_DIAMETERS:  
12        # Calculate the minimum amount needed to exceed As for the  
13        current diameter  
14        min_amount = int(As_needed / (PI * diameter**2 / 4)) + 1  
15        if continuous:  
16            # There must be at least 2 bars  
17            min_amount = max(min_amount, 2)  
18        # loop through all the possible amounts, starting from  
19        min_amount  
20        for amount in range(min_amount, MAX_AMOUNT + 1):  
21            if not check_width(b, c, stirrup_dia, diameter, amount):  
22                break # if the reinforcement can't fit in the beam,  
23                stop the loop for that diameter  
24                As = amount * PI * diameter**2 / 4  
25                difference_As = As - As_needed  
26                # if the difference is smaller than the current smallest  
27                difference = better solution  
28                if difference_As < smallest_difference_As:  
29                    smallest_difference_As = difference_As  
30                    best_diameter = diameter
```

```

26         best_amount = amount
27         break
28     # if the best diameter is still 0, no reinforcement found that
fits in the beam
29     if best_diameter == 0:
30         raise ValueError("No reinforcement found that fits in the
beam")
31     return best_diameter, best_amount

```

Code 4.28: Functie die optimale diameter en aantal wapening vindt

De functie in Code 4.28 begint met het instellen van de beste diameter en het beste aantal op nul. Vervolgens zoekt de functie naar het kleinste verschil tussen de benodigde en de gevonden wapening, waarbij dit verschil altijd groter moet zijn dan 30 mm². Dit is omdat één staaf met een diameter van 6 mm, de kleinste diameter die kan worden geplaatst, een oppervlakte van 28,27 mm² heeft.

Daarna doorloopt de functie alle mogelijke diameters (6, 8, 10, 12, 14, 16, 20, 25 en 32 mm). Voor elke diameter berekent de functie hoeveel staven nodig zijn om de benodigde hoeveelheid wapening te overschrijden. Voor continue wapening moeten er altijd minimaal twee staven zijn om de beugels op hun plaats te houden.

Binnen elke diameterlus doorloopt de functie mogelijke aantallen staven. Het programma controleert of de wapening in de balk past. Als het niet past, wordt de lus voor die diameter vroegtijdig beëindigd.

Voor configuraties die wel passen, berekent de functie de totale oppervlakte van de wapening en bepaalt het het verschil tussen dit oppervlak en *As_needed*. Als dit verschil kleiner is dan het tot dan toe gevonden verschil, wordt dit de nieuwe beste combinatie.

Aan het einde controleert de functie of de beste diameter nog steeds nul is. Als dat het geval is, is er geen geschikte combinatie gevonden. Tot slot retourneert de functie de beste diameter en het beste aantal staven.

De functie die de dwarskrachtwapening bepaald is eenvoudiger dan deze van de langswapening en wordt getoond in Code 4.29.

```

1  def determine_shear_reinforcement(
2      L_beam: float,
3      As_needed: list,
4      Asw_s_min: float,
5      d: float
6  ):
7      i = 0
8      intervals = determine_intervals(L_beam, As_needed, Asw_s_min, 3)
9      while True:
10         try:
11             shear_reinforcements = []
12             stirrup_diameter = STIRRUP_DIAMETERS[i] # [10, 8, 6]
13             s_max = calculate_s_max(d)
14             for interval in intervals:

```

```

15         Asw_s_needed = interval[2]
16         s = calculate_s(Asw_s_needed, stirrup_diameter)
17         if s > s_max: # If the needed spacing is bigger than
the maximum spacing, take a lower stirrup diameter
18             raise ValueError("Spacing is too big, take a
lower stirrup diameter!")
19             shear_reinforcements.append((interval[0], interval
[1], stirrup_diameter, s))
20             break
21
22     except ValueError as e:
23         if str(e) == "Spacing is too big, take a lower stirrup
diameter!":
24             i += 1
25             continue
26     return shear_reinforcements, stirrup_diameter

```

Code 4.29: Functie die dwarskrachtwapening bepaald

De functie in Code 4.29 begint met het bepalen van vergelijkbare intervallen als die in de functie voor langswapening in Code 4.27. Hierbij worden intervallen gemaakt waarin de waarde ofwel gelijk is aan de minimale dwarskrachtwapening, ofwel groter is, waarbij de grootste waarde wordt opgeslagen.

Vervolgens start een while-lus die doorgaat totdat de juiste diameter en tussenafstand zijn gevonden. De functie begint met de grootst mogelijke diameter (10 mm). Daarna berekent de functie de maximale tussenafstand en de tussenafstand voor elk interval. Als de berekende tussenafstand groter is dan de maximale toegestane afstand, wordt een fout gegenereerd. Deze fout wordt opgevangen op regel 18, waarna de diameter van de beugel wordt verkleind, wat resulteert in een kleinere tussenafstand.

Als alle berekende tussenafstanden binnen de maximale limiet vallen, wordt de lus doorbroken en retourneert de functie de dwarskrachtwapening samen met de diameter. Deze diameter wordt vervolgens gecontroleerd in de functie van Code 4.26.

Fase 4: Controle

Tot slot wordt de balk opnieuw geanalyseerd en gecontroleerd met de nieuwe wapeningsconfiguratie. Deze controle richt zich uitsluitend op de criteria waarop de wapening is geoptimaliseerd. Als de wapening aan deze controle voldoet, wordt deze geretourneerd. Zo niet, dan wordt het hele proces herhaald totdat een geschikte wapening is gevonden.

4.3.4 Optimalisatie doorsnede en wapening

Als voor een bepaalde doorsnede geen geschikte wapening wordt gevonden, kan de gebruiker ervoor kiezen om de dimensies van de doorsnede te optimaliseren. Hierbij worden de hoogte en breedte aangepast om de materialen optimaal te benutten. Dit betekent een rek in het beton van drieënhalve promille en een rek in de primaire wapening van tien promille. De functie die deze berekeningen uitvoert, wordt getoond in Code 4.30.

```

1 def optimize_section(self, optimize_criteria: list, reinforcement:
    dict):
2     max_moment = dtb_query_max_moment(self.combination_ids["ULS-P"],
    0, self.L_beam, 1)[1]
3     L = self.L_beam * 1000
4     d = L / 15
5     h = d + self.c + 8 + 20 / 2
6     b = h / 2
7
8     fcd = 0.85 * self.material["fck"] / 1.5
9     d_opt = 2.31 * np.sqrt((abs(max_moment) * 10**6) / (b * fcd))
10    h_opt = (d_opt + self.c + 8 + 20 / 2) // 50 * 50
11    b_opt = best_h / 2
12    while True:
13        try:
14            reinforcement = self.optimize_reinforcement(
    optimize_criteria, b_opt, h_opt, reinforcement)
15            break
16
17        except ValueError as e:
18            if str(e) == "No reinforcement found that fits in the
    beam":
19                h_opt += 50
20                b_opt += 25
21                continue
22    return h_opt, b_opt, reinforcement

```

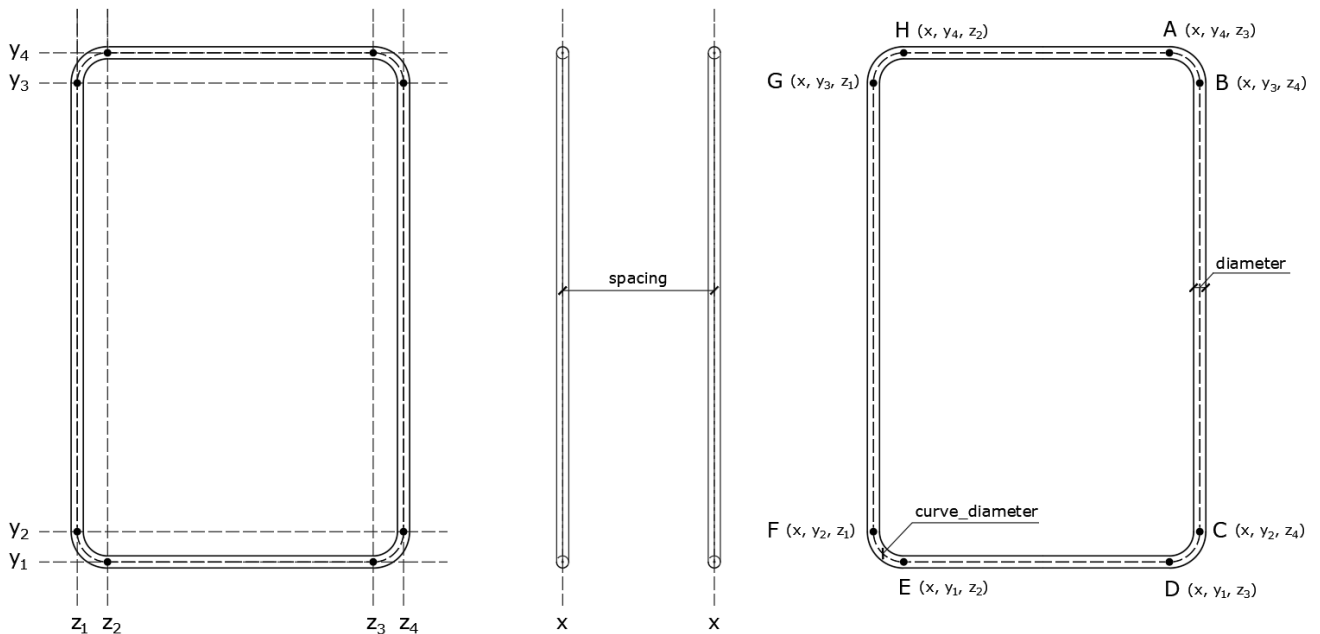
Code 4.30: Functie die doorsnede en wapening optimaliseert

In de functie van Code 4.30 wordt eerst het maximale optredende moment in UGT-B uit de database gehaald. Vervolgens wordt de nuttige hoogte geschat met een schattingsformule op basis van de slankheid (regel 4). Daarna wordt een eerste hoogte bepaald met aannames voor de diameter van de beugel en de primaire wapening, waarmee een eerste breedte wordt berekend. Deze breedte wordt dan gebruikt om de optimale nuttige hoogte te vinden (regel 9). De optimale hoogte wordt op dezelfde manier berekend en de optimale breedte is de helft van deze hoogte.

De wapening wordt vervolgens geoptimaliseerd voor deze dimensies. Als er nog steeds geen geschikte wapening wordt gevonden, worden de hoogte met 50 mm en de breedte met 25 mm vergroot. Deze lus herhaalt zich totdat een geschikte wapening is gevonden. Tot slot worden de hoogte, breedte en wapening geretourneerd.

4.3.5 Positionering wapening

Het programma kan de geplaatste of berekende wapening tekenen. Hierbij worden de contouren van de balk, de beugels en de wapeningsstaven aan de onder- en bovenkant weergegeven. Om dit te realiseren, beschikt het programma over functies die de berekende wapening omzetten in coördinaten. Het aantal en de positie van deze punten zijn afhankelijk van het type wapening. Figuur 4.4 toont het schema dat is gebruikt om een beugel te beschrijven.



Figuur 4.4: Schematisering beugel

Figuur 4.4 toont twee vooraanzichten en één zijaanzicht van een beugel. Hierop is te zien dat een beugel wordt beschreven door acht posities, waarbij elke positie drie coördinaten heeft (x , y en z). In het vooraanzicht aan de linkerkant van de figuur is te zien dat er voor alle posities vier verschillende y -coördinaten en vier verschillende z -coördinaten zijn. Het zijaanzicht in het midden van de figuur toont dat alle posities van de beugel dezelfde x -coördinaat delen. In het vooraanzicht aan de rechterkant van de figuur wordt getoond hoe de posities in de database worden opgeslagen. Elke positie krijgt een letter toegewezen. Naast de acht posities is het noodzakelijk om de diameter van de beugel, de diameter van de bocht van de aslijn en de tussenafstand op te slaan om de beugels correct te kunnen tekenen. Al deze informatie wordt opgeslagen in de database, zoals weergegeven in Figuur 4.5.

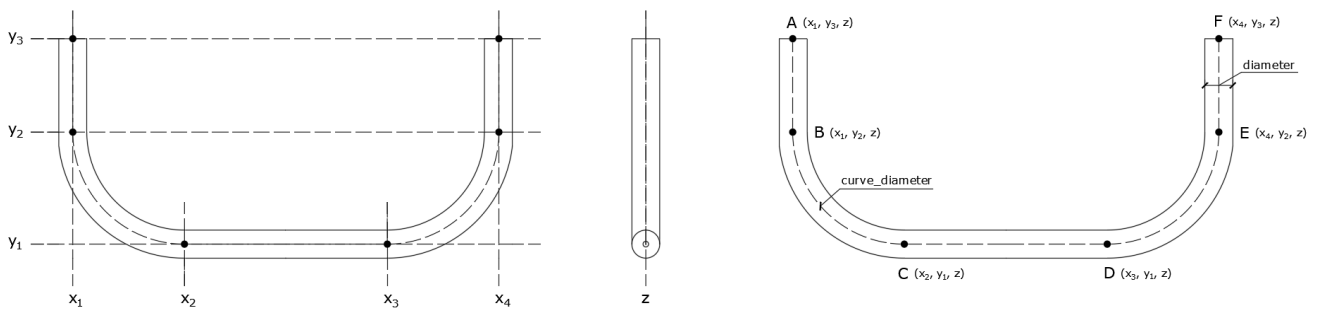
Table: reinforcement_stirrups		Filter in any column																											
	stirrup_id	diameter	curve_diameter	spacing	Ax	Ay	Az	Bx	By	Bz	Cx	Cy	Cz	Dx	Dy	Dz	Ex	Ey	Ez	Fx	Fy	Fz	Gx	Gy	Gz	Hx	Hy	Hz	▼
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
2	2	8	40	320	335.2	206.0	111.0	335.2	186.0	131.0	335.2	-186.0	131.0	335.2	-206.0	111.0	335.2	-206.0	-111.0	335.2	-186.0	-13...	335.2	186.0	-131.0	335.2	206.0	-111.0	
3	3	8	40	320	626.4	206.0	111.0	626.4	186.0	131.0	626.4	-186.0	131.0	626.4	-206.0	111.0	626.4	-206.0	-111.0	626.4	-186.0	-13...	626.4	186.0	-131.0	626.4	206.0	-111.0	
4	4	8	40	320	917.6	206.0	111.0	917.6	186.0	131.0	917.6	-186.0	131.0	917.6	-206.0	111.0	917.6	-206.0	-111.0	917.6	-186.0	-13...	917.6	186.0	-131.0	917.6	206.0	-111.0	
5	5	8	40	320	1208.8	206.0	111.0	1208.8	186.0	131.0	120...	-186.0	131.0	120...	-206.0	111.0	120...	-206.0	-111.0	120...	-186.0	-13...	120...	186.0	-131.0	120...	206.0	-111.0	
6	6	8	40	320	1500.0	206.0	111.0	1500.0	186.0	131.0	150...	-186.0	131.0	150...	-206.0	111.0	150...	-206.0	-111.0	150...	-186.0	-13...	150...	186.0	-131.0	150...	206.0	-111.0	
7	7	8	40	150	1642.86	206.0	111.0	164...	186.0	131.0	164...	-186.0	131.0	164...	-206.0	111.0	164...	-206.0	-111.0	164...	-186.0	-13...	164...	186.0	-131.0	164...	206.0	-111.0	
8	8	8	40	150	1785.71	206.0	111.0	178...	186.0	131.0	178...	-186.0	131.0	178...	-206.0	111.0	178...	-206.0	-111.0	178...	-186.0	-13...	178...	186.0	-131.0	178...	206.0	-111.0	
9	9	8	40	150	1928.57	206.0	111.0	192...	186.0	131.0	192...	-186.0	131.0	192...	-206.0	111.0	192...	-206.0	-111.0	192...	-186.0	-13...	192...	186.0	-131.0	192...	206.0	-111.0	
10	10	8	40	150	2071.43	206.0	111.0	207...	186.0	131.0	207...	-186.0	131.0	207...	-206.0	111.0	207...	-206.0	-111.0	207...	-186.0	-13...	207...	186.0	-131.0	207...	206.0	-111.0	

Figuur 4.5: Tabel voor beugels

Figuur 4.5 toont hoe tien beugels worden opgeslagen. Elke rij bevat de informatie voor één beugel. In de database wordt voor elke coördinaat een aparte kolom aangemaakt, zodat deze coördinaten later kunnen worden opgevraagd.

De langswapening aan de boven- en onderzijde van de doorsnede wordt opgesplitst in continue en extra langswapening. In elke wapeningsconfiguratie bevindt zich een continue langswapeningstaaf in de hoeken van een beugel, wat betekent dat er zowel boven als onder twee continue wapeningsstaven aanwezig zijn. Deze staven lopen over de volledige lengte en hebben aan elk uiteinde een opbuiging. De langswapening aan de onderzijde buigt naar boven, terwijl die aan de

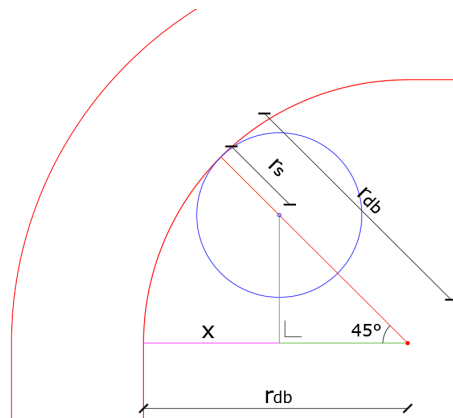
bovenzijde naar beneden buigt. Deze opbuigingen overlappen elkaar, waarbij de langswapening aan de onderzijde altijd aan de buitenzijde zit. Het schema van de continue langswapening wordt getoond in Figuur 4.6.



Figuur 4.6: Schematisering continue langswapening

Figuur 4.6 toont twee zijaanzichten en één vooraanzicht van een continue langswapeningsstaaf die aan de onderzijde van de balk wordt geplaatst. Deze wapeningsstaaf wordt beschreven door zes posities, waarbij elke positie een letter toegewezen krijgt en een coördinaat heeft voor elke richting. Naast de zes posities wordt ook informatie over de diameter van de staaf en de diameter van de bocht van de aslijn in de database opgeslagen. Voor de continue wapeningsstaven die aan de bovenzijde worden geplaatst, wordt hetzelfde schema gebruikt, maar dan gespiegeld.

Zoals eerder gezegd worden de continue wapeningsstaven in de hoeken van de beugels geplaatst. Doordat de hoeken van de beugels een buiging hebben, worden de wapeningsstaven met een kleine afwijking geplaatst. Deze afwijking wordt getoond in Figuur 4.7



Figuur 4.7: Afwijking langswapening door kromming beugel

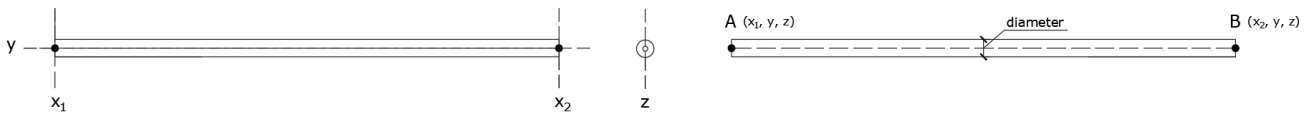
De afwijking wordt in Figuur 4.7 weergegeven door de afstand x . Deze afstand kan worden berekend op basis van de doorndiameter van de beugel en de diameter van de langswapeningsstaaf. De formule voor deze afstand wordt gegeven in formule (4.1).

$$x = r_{db} - ((r_{db} - r_s) \times \cos(45)) \quad (4.1)$$

waarin:

- r_{db} de doorndiameter van de beugel is;
- r_s de diameter van de wapeningsstaaf is.

De extra wapeningsstaven worden gebruikt wanneer de hoeveelheid continue langswapening niet voldoende is. Deze staven zijn rechte staven zonder opbuiging en worden voorgesteld volgens het schema in Figuur 4.8.



Figuur 4.8: Schematisering extra langswapening

Figuur 4.8 toont opnieuw twee zijaanzichten en een vooraanzicht. De extra langswapeningsstaaf wordt beschreven door twee posities. Zowel de y- als z-coördinaat voor deze staaf hebben slechts één waarde. Naast de twee posities wordt ook de diameter van de staaf in de database opgeslagen.

4.4 Staal

Voor de berekeningen van staal in het programma moet de gebruiker de lengte, het profiel en de vloeigrens van het staal opgeven. Met deze gegevens kan het programma de balk controleren op sterkte, stabiliteit en doorbuiging. Daarnaast kan het programma voor elk van deze criteria ook het profiel optimaliseren. Dit kan betekenen dat een groter profiel wordt gekozen als het huidige profiel niet voldoet, of een kleiner profiel als het huidige profiel overgedimensioneerd is. In deze sectie wordt eerst uitgelegd hoe de controles worden uitgevoerd, gevolgd door een uitleg over het proces van het optimaliseren van de doorsnede.

4.4.1 Controle doorsnede

De functies voor de controle van de doorsnede op sterkte, stabiliteit en doorbuiging hebben allemaal dezelfde structuur. De structuur van deze functies is trouwens ook gelijkaardig aan de structuur van de functies voor de controle van de doorsnede bij beton, zoals uitgelegd in Sectie 4.3.2. De functies die de stabiliteit van een stalen balk controleert worden verder besproken in dit gedeelte. De overkoepelende functie voor deze controle wordt getoond in Code 4.31.

```

1 def control_stability(self, section: dict, iteration: int):
2     stability_check = []
3     for i in range(len(self.stability_supports) - 1):
4         interval = (self.stability_supports[i], self.
5 stability_supports[i + 1])
6         max_moments = self.query_max_moment("ULS-P", interval[0],
7 interval[1], iteration)
8         stability_check.append(control_section_stability(self.fy,
9 section, max_moments, interval, iteration))
10    return all(stability_check)

```

Code 4.31: Overkoepelende functie voor controle stabiliteit

De functie in Code 4.31 voert de controle per interval uit. Een interval is de afstand tussen twee steunen die het kippen van de balk belemmeren. De functie begint met het aanmaken van een lege lijst om de controle-resultaten op te slaan. Vervolgens doorloopt de functie alle intervals. Voor elk

interval zoekt de functie *query_max_moments* op regel 5 het grootste optredende moment binnen het interval voor alle combinaties in UGT-B. Daarna voert de functie de daadwerkelijke controle uit met de functie *control_section_stability*. De uitkomst van deze controle wordt toegevoegd aan de eerder aangemaakte lijst. Ten slotte retourneert de functie *True* als de controle in elk interval voldoet, en *False* als dit niet het geval is.

De overkoepelende functies voor de controle van sterkte en doorbuiging zijn vergelijkbaar. Bij de controle van sterkte wordt echter de volledige balk in één keer gecontroleerd zonder gebruik van intervallen, en worden alle interne krachten opgevraagd in plaats van alleen het maximale moment. Bij de controle van doorbuiging wordt een interval bepaald door de afstand tussen twee steunpunten, in plaats van tussen steunpunten en kipbeveiligingen. Bovendien worden bij de controle van doorbuiging de doorbuigingen op elke positie uit de database opgevraagd, in plaats van alleen het maximale moment, en dit kan voor BGT-K, BGT-F en BGT-Q.

De functie *control_section_stability* die de controle van een interval uitvoert, wordt getoond in Code 4.32.

```

1  def control_section_stability(
2      fy: int,
3      section: dict,
4      max_moments: dict,
5      interval: tuple,
6      iteration: int
7  ) -> bool:
8      L = (interval[1] - interval[0]) * 1000
9      Mb_rd = calculate_lt_resistance(L, E, G, fy, section)
10
11     check_list_stability = []
12     database = []
13     for key, data in max_moments.items():
14         Med = abs(data[1])
15         stability_ratio, stability_check = control_bending(Med,
16         Mb_rd)
17         database.append((interval[0], interval[1], round(
18         stability_ratio, 2), key, iteration))
19         check_list_stability.append(stability_check)
20
21     check_stability = all(check_list_stability)
22     dtb_insert_steel_controls_stability(database)
23     return check_stability

```

Code 4.32: Functie die stabiliteit controleer

In de functie van Code 4.32 wordt eerst een functie opgeroepen die de kipweerstand berekent, zoals uitgelegd in Sectie 3.3.4. Vervolgens initialiseert de functie twee lege lijsten: één voor het opslaan van de controle-resultaten en de andere voor het opslaan van alle resultaten in de database. Daarna wordt voor elke combinatie in UGT-B het optredende moment vergeleken met de weerstand tegen kip (zie regel 15). Van deze controle wordt de uitkomst (*True* of *False*) en de verhouding opgeslagen in de tweede lijst voor de database, terwijl alleen de uitkomst wordt

opgeslagen in de eerste lijst. Indien alle combinaties voldoen, retourneert de functie *True*, anders *False*.

De controlefuncties voor sterkte en doorbuiging vertonen opnieuw gelijkenissen, maar hebben elk hun eigen specifieke controles. Bij sterkte wordt gecontroleerd op buiging, afschuiving en een combinatie van beide. Bij doorbuiging wordt gecontroleerd of de optredende doorbuiging kleiner is dan een opgegeven maximale waarde.

4.4.2 Optimalisatie profiel

Het staalprofiel kan geoptimaliseerd worden voor sterkte, stabiliteit en doorbuiging, of voor een combinatie van deze criteria. Elk van deze criteria heeft zijn eigen functies met een vergelijkbare structuur. Daarnaast heeft de gebruiker de mogelijkheid om te kiezen of alleen IPE- of alleen HEB-profielen moeten worden overwogen. De functie die de optimale sectie vindt voor de volledige balk op basis van stabiliteit wordt weergegeven in Code 4.33.

```
1 def optimize_section_stability(self, iteration: int, section: dict,
2   section_type: str = ""):
3     while True:
4         best_section = None
5         for i in range(len(self.stability_supports) - 1):
6             interval = (self.stability_supports[i], self.
7 stability_supports[i + 1])
8             max_moments = self.query_max_moment("ULS-P", interval
9 [0], interval[1], iteration)
10            # Search section for this interval
11            optimized_section = optimize_steel_section_stability(
12 self.fy, section, max_moments, interval, section_type, iteration)
13            # Update best section if needed
14            if best_section == None or optimized_section["Wply"] >
15 best_section["Wply"]:
16                best_section = optimized_section
17            # Check if the section still passes with new |self| weight
18            combinations = [combination for combination in self.analysis
19 .combinations if combination.criteria == "ULS-P"]
20            self.analyse_beam(best_section, combinations, True)
21            stability_check = self.control_stability(best_section,
22 iteration)
23            if stability_check:
24                break
25            else:
26                iteration = self.iteration
27                section = best_section
28        return best_section
```

Code 4.33: Functie dat profiel optimaliseert voor stabiliteit

De functie in Code 4.33 begint met een While-lus die doorgaat totdat een geschikte sectie is gevonden. Aanvankelijk wordt de variabele *best_section* ingesteld op *None*, omdat er nog geen

sectie is berekend. Vervolgens wordt elk interval doorlopen, vergelijkbaar met de functie in Code 4.31. Hiervoor worden de maximale momenten uit de database gehaald en wordt de functie *optimize_steel_section_stability* aangeroepen om de optimale sectie voor dat interval te bepalen.

Als er nog geen beste sectie is gevonden (*best_section == None*) of als het traagheidsmoment van de gevonden sectie groter is dan die van de eerder gevonden sectie, wordt de gevonden sectie toegewezen als beste sectie. De sectie met het grootste traagheidsmoment wordt verkozen, omdat deze ook voldoet aan alle voorgaande intervallen.

Nadat alle intervallen zijn doorlopen, wordt de balk opnieuw geanalyseerd met het eigengewicht van de nieuwe sectie. Vervolgens wordt de balk opnieuw gecontroleerd. Als de controle slaagt, wordt de lus doorbroken en wordt de gekozen sectie geretourneerd. Indien niet, wordt de lus herhaald met de interne krachten uit de laatste analyse.

De functie die de optimale sectie vindt voor één interval voor de stabiliteit wordt getoond in Code 4.34.

```
1 def optimize_steel_section_stability(  
2     fy: int,  
3     section: dict,  
4     max_moments: dict,  
5     interval: tuple,  
6     section_type: str,  
7     iteration: int,  
8 ) -> dict[str, float]:  
9     check_stability = control_section_stability(fy, section,  
10    max_moments, interval, True)  
11     if not check_stability:  
12         sections = dtb_query_section_by_Wply(section["Wply"],  
13    section_type, True)  
14         ordered_sections = sorted(sections, key=lambda x: x["Wply"])  
15         for section in ordered_sections:  
16             if control_section_stability(fy, section, max_moments,  
17    interval, iteration):  
18                 return section  
19             raise ValueError("No section found.")  
20     else:  
21         sections = dtb_query_section_by_Wply(section["Wply"] + 1,  
22    section_type, False)  
23         ordered_sections = sorted(sections, key=lambda x: x["Wply"],  
24    reverse=True)  
25         for section in ordered_sections:  
26             if not control_section_stability(fy, section,  
27    max_moments, interval):  
28                 return sections[sections.index(section) - 1]  
29             elif sections.index(section) == len(sections) - 1:  
30                 return section
```

Code 4.34: Functie dat optimaal profiel vindt voor één interval op basis van stabiliteit

In de functie van Code 4.34 begint het proces met een controle om te zien of het huidige profiel voldoet op basis van stabiliteit. Als het profiel niet voldoet, wordt er gezocht naar een groter profiel (regel 10). Hiervoor worden alle profielen uit de database gehaald die een groter traagheidsmoment hebben dan het huidige profiel. Deze profielen worden vervolgens gesorteerd van klein naar groot. Voor elk van deze profielen wordt gecontroleerd of het aan de vereisten voldoet. Als een profiel voldoet, wordt dat profiel geretourneerd en de lus beëindigd. Als geen enkel profiel aan de criteria voldoet, wordt er een foutmelding gegenereerd.

Als het huidige profiel wel voldoet aan de controle, wordt er gezocht naar een kleiner profiel (regel 17). Hiervoor worden alle profielen uit de database gehaald die een traagheidsmoment hebben dat kleiner is dan of gelijk is aan het traagheidsmoment van het huidige profiel. Deze profielen worden gesorteerd van groot naar klein. Vervolgens wordt een For-lus gestart om elk van deze profielen te controleren (regel 20). Hier wordt gecontroleerd of het profiel niet langer aan de stabiliteit voldoet. Als dat het geval is, wordt het vorige profiel, dat nog wel aan de controle voldeed, geretourneerd. Als alle profielen aan de controle voldoen, wordt het laatst gecontroleerde profiel geretourneerd, omdat dit het profiel is met het kleinste traagheidsmoment.

De functies voor optimalisatie van sterkte en doorbuiging delen dezelfde structuur, maar verschillen in de specifieke controles die worden uitgevoerd voor elk criterium. De gebruiker heeft de mogelijkheid om het profiel te optimaliseren voor sterkte, stabiliteit en doorbuiging. In dat geval wordt het profiel gekozen dat aan alle drie criteria voldoet. De functie die deze optimalisatie uitvoert, wordt weergegeven in Code 4.35.

```
1 def optimize_section(self, section: dict, section_type: str,
2   strength: bool, stability: bool, deflection: bool, criteria: list):
3     original_iteration = self.iteration
4     results = []
5     if strength:
6         optimized_section = self.optimize_section_strength(
7         original_iteration, section_type)
8         results.append(optimized_section)
9     if stability:
10        optimized_section = self.optimize_section_stability(
11        original_iteration, section, section_type)
12        results.append(optimized_section)
13    if deflection:
14        for crit in criteria:
15            optimized_section = self.optimize_section_deflections(
16            crit, original_iteration, section, section_type)
17            results.append(optimized_section)
18    best_section = max(results, key=lambda x: x["Wply"])
19    return best_section
```

Code 4.35: Overkoepelende functie voor optimalisatie

De functie in Code 4.35 begint met het vastleggen van het oorspronkelijke iteratienummer. Dit zorgt ervoor dat de optimalisatie wordt uitgevoerd op basis van de interne krachten en doorbuiging van de oorspronkelijke analyse, in plaats van de waarden van het reeds geoptimaliseerde profiel.

Vervolgens initialiseert de functie een lege lijst om de resultaten op te slaan. Voor elk gekozen criterium (sterkte, stabiliteit en doorbuiging) wordt een geoptimaliseerd profiel bepaald en aan deze lijst toegevoegd. Na het bepalen van de profielen wordt op regel 14 het profiel met het grootste traagheidsmoment gekozen, omdat dit profiel aan alle criteria voldoet. Uiteindelijk wordt dit optimale profiel geretourneerd.

4.5 Hout

Voor het berekenen van houten balken in het programma moet de gebruiker de balkafmetingen, de houtsterkteklasse en de vochtigheidsklasse invoeren. Met deze informatie controleert het programma de balk op sterkte, stabiliteit en doorbuiging. Bovendien kan het programma de afmetingen van de doorsnede voor elk van deze criteria optimaliseren. Dit houdt in dat het programma een grotere doorsnede kiest als de huidige afmetingen onvoldoende zijn, of een kleinere doorsnede als de huidige afmetingen te groot zijn. In deze sectie wordt eerst uitgelegd hoe het programma kruip bij houten balken in rekening brengt. Daarna worden de controleprocessen beschreven, gevolgd door een uitleg over de optimalisatiefuncties.

4.5.1 Belastingscombinaties voor kruip

In formules (3.98) tot en met (3.100) van Sectie 3.4.5 is uitgelegd dat bij het berekenen van de totale doorbuiging inclusief kruip, bepaalde factoren moeten worden opgeteld bij de onmiddellijke doorbuiging in BGT-K. Welke factoren moeten worden opgeteld, hangt af van het type belasting: blijvende, overheersende veranderlijke of niet-overheersende veranderlijke belasting. Hierdoor kan de doorbuiging in BGT-K niet eenvoudig met één enkele factor worden vermenigvuldigd om de totale doorbuiging te berekenen. Deze factoren moeten echter voor de analyse in rekening worden gebracht. Het programma doet dit door aparte belastingscombinaties te genereren waarbij de combinatiesets (zie Sectie 4.2.2) rekening houden met deze factoren. Code 4.36 toont de functie die deze combinatiesets genereert.

```
1 def create_combinationsets_SLS_C_timber(load_groups, Kdef: float):
2     combinationsets = []
3     # Combinations with variable load groups
4     for lg in load_groups:
5         if isinstance(lg, PermanentLoadGroup):
6             continue
7
8         leading_lg = lg
9         factors = []
10        for _lg in load_groups:
11            if isinstance(_lg, PermanentLoadGroup):
12                factors.append((_lg.unfavourable_SLS * (1 + Kdef),
13                               _lg.favourable_SLS * (1 + Kdef)))
14
15            elif isinstance(_lg, VariableLoadGroup):
16                if _lg == leading_lg:
17                    factors.append((1 + _lg.psi_2 * Kdef, 1 + _lg.
18                                   psi_2 * Kdef))
```

```

17         else:
18             factors.append((_lg.psi_0+_lg.psi_2 *Kdef), 0))
19         combinationsets.append(factors)
20     # Combinations with only permanent load groups
21     factors = []
22     for _lg in load_groups:
23         if isinstance(_lg, PermanentLoadGroup):
24             factors.append((_lg.unfavourable_SLS * (1 + Kdef), _lg.
25 favourable_SLS * (1 + Kdef)))
26
27         elif isinstance(_lg, VariableLoadGroup):
28             factors.append((0, 0))
29         combinationsets.append(factors)
30     return combinationsets

```

Code 4.36: Functie dat combinatiesets genereert voor kruip in rekening te brengen

De functie in Code 4.36 heeft dezelfde structuur als de functie in Code 4.10 in Sectie 4.2.2. Eerst genereert de functie combinatiesets voor de veranderlijke belastingsgevallen, waarbij elk veranderlijk belastingsgeval als overheersend wordt beschouwd. Bij deze sets worden de factoren, zoals beschreven in formules (3.98) tot en met (3.100), toegevoegd aan de bijbehorende belastingsgevallen. Nadat alle combinatiesets met veranderlijke belastingsgevallen zijn gemaakt, worden de sets met enkel blijvende belastingsgevallen gegenereerd. Hierbij worden alleen de factoren van de blijvende belastingsgevallen toegevoegd en de rest wordt gelijkgesteld aan nul.

Na het genereren van de combinatiesets worden de belastingscombinaties samengesteld, zoals uitgelegd in Sectie 4.2.2. Vervolgens wordt de analyse uitgevoerd met deze belastingscombinaties, waardoor de totale doorbuigingen in de database worden opgeslagen onder de unieke combinatienummers zodat deze gegevens later kunnen worden opgevraagd wanneer nodig.

4.5.2 Controle doorsnede

In het programma worden houten balken gecontroleerd op sterkte, stabiliteit en doorbuiging. De onmiddellijke doorbuiging kan worden gecontroleerd voor BGT-K, BGT-F en BGT-Q, maar de doorbuiging met kruip kan alleen worden gecontroleerd voor BGT-K. De functies die deze controles uitvoeren, hebben dezelfde structuur als de controlefuncties voor beton en staal. Code 4.37 toont de functie die de balk op doorbuiging controleert.

```

1 def control_deflections(self, iteration: int, criteria: str):
2     deflection_check = []
3     max_deflection = self.max_deflection[criteria]
4     for interval in self.analysis.intervals:
5         deflections = self.query_deflections(criteria, interval[0],
6 interval[1], iteration)
7         deflection_check.append(control_section_deflections(interval
8 , deflections, max_deflection, iteration))
9     return all(deflection_check)

```

Code 4.37: Functie die balk controleert op doorbuiging

De functie in Code 4.37 heeft een parameter, *criteria*, die het type combinatie aangeeft, zoals *SLS-C* (BGT-K), *SLS-F* (BGT-F), *SLS-Q* (BGT-Q) of *SLS-C FINAL* (BGT-K met kruip) waarvoor de controle wordt uitgevoerd. Deze aanpak maakt het mogelijk om met één functie alle combinatievarianten te behandelen.

De functie begint met het initiëren van een lege lijst om de resultaten van de doorbuigingscontroles voor elk interval op te slaan. Vervolgens bepaalt de functie de maximale doorbuiging voor het opgegeven type combinatie uit een vooraf aangemaakte bibliotheek, waarin de maximale doorbuigingen voor elk type zijn opgeslagen. De gebruiker kan voor elke combinatievariant een maximale doorbuiging instellen in deze bibliotheek.

Daarna doorloopt de functie elk interval tussen twee steunpunten. Elk interval wordt gecontroleerd op doorbuiging met de functie *control_section_deflections*, en de uitkomst wordt in de eerder aangemaakte lijst opgeslagen. Als alle resultaten in deze lijst positief zijn, retourneert de functie *True* en anders retourneert ze *False*.

Code 4.38 toont de functie die één interval controleert op doorbuiging.

```
1 def control_section_deflections(  
2     interval: int,  
3     deflections: dict,  
4     max_def: float,  
5     iteration: int,  
6 ) -> bool:  
7  
8     checklist_deflections = []  
9     database = []  
10  
11     for key, deflection in deflections.items():  
12         for i in range(len(deflection)):  
13             d = round(deflection[i][1], 2)  
14             L = (interval[1] - interval[0]) * 1000  
15             check, max_d = control_deflection(L, d, max_def)  
16             database.append((deflection[i][0], d, max_d, key,  
17 iteration))  
18             checklist_deflections.append(check)  
19  
20     check_deflection = all(checklist_deflections)  
21     dtb_insert_timber_deflections(database)  
22     return check_deflection
```

Code 4.38: Functie die één interval op doorbuiging controleert

De functie in Code 4.38 werkt op dezelfde manier als de controles bij beton en staal. Eerst worden twee lijsten geïnitieerd: één om de resultaten van de controle op te slaan en een tweede om de informatie voor de database te bewaren. Daarna worden twee for-lussen gestart. De eerste for-lus (regel 11) doorloopt alle combinaties voor het opgegeven type, en de tweede for-lus (regel 12) doorloopt alle posities binnen die combinatie. Voor elke positie wordt de doorbuiging berekend, de overspanning bepaald en de controle uitgevoerd. De resultaten worden vervolgens in de twee lijsten opgeslagen.

Na het doorlopen van alle combinaties en posities controleert de functie op regel 19 of elke positie geslaagd is. Daarna wordt alle informatie naar de database geschreven en wordt het eindresultaat van de controles geretourneerd.

4.5.3 Optimalisatie doorsnede

De optimalisatie voor de breedte en hoogte van de doorsnede kan voor sterkte, stabiliteit en doorbuiging voor verschillende combinatievarianten. Het programma heeft zoals staal een overkoepelende functie die andere functies oproept voor de optimalisatie uit te voeren voor één criterium. Nadat alle optimalisaties zijn uitgevoerd, retourneert deze functie de doorsnede die alle criteria voldeed (zie de functie in Code 4.35).

De individuele optimalisatiefuncties hebben allemaal dezelfde structuur maar verschillen in de specifieke uitgevoerde controle. Code 4.39 toont de optimalisatiefunctie voor doorbuiging.

```
1 def optimize_section_deflections(self, b: int, h: int, iteration:
  int, criteria: str):
2     combinations = [combination for combination in self.analysis.
  combinations if combination.criteria == criteria]
3     deflection_check = self.control_deflections(iteration, criteria)
4     optimized_b = b
5     optimized_h = h
6
7     if not deflection_check:
8         while True:
9             optimized_b += 10
10            optimized_h += 20
11            self.analyse_beam(optimized_b, optimized_h, combinations
  , True)
12            deflection_check = self.control_deflections(self.
  iteration, criteria)
13            if deflection_check:
14                break
15        else:
16            while True:
17                optimized_b -= 10
18                optimized_h -= 20
19                self.analyse_beam(optimized_b, optimized_h, combinations
  , True)
20                deflection_check = self.control_deflections(self.
  iteration, criteria)
21                if not deflection_check:
22                    optimized_b += 10
23                    optimized_h += 20
24                    break
25        return optimized_b, optimized_h
```

Code 4.39: Functie die doorsnede optimaliseert voor doorbuiging

De functie in Code 4.39 optimaliseert de doorsnede van een balk voor een opgegeven combinatie-variant. Eerst verzamelt de functie alle relevante combinaties. Vervolgens controleert de functie of de opgegeven doorsnede voldoet aan de eisen.

Als de doorsnede niet voldoet, wordt een while-lus gestart (regel 8 tot en met 14) om de eerstvolgende doorsnede te vinden die wel voldoet. Hierbij wordt de breedte telkens met tien millimeter en de hoogte met twintig millimeter vergroot. Hierbij wordt de balk steeds geanalyseerd en gecontroleerd totdat de controle geslaagd is.

Als de opgegeven doorsnede wel voldoet, zoekt de functie naar de kleinste doorsnede die ook nog voldoet (regel 16 tot en met 24). De breedte wordt hierbij steeds met tien millimeter en de hoogte met twintig millimeter verkleind. De balk wordt telkens opnieuw geanalyseerd en gecontroleerd totdat de controle niet meer voldoet. Zodra de doorsnede niet meer voldoet, wordt de breedte met tien millimeter en de hoogte met twintig millimeter vergroot om de laatst geldige doorsnede te vinden en te retourneren.

Hoofdstuk 5

Gebruikersinterface

In dit hoofdstuk worden het ontwerp en de functionaliteit van de gebruikersinterface besproken. Deze interface is gebouwd met de Python-bibliotheek `PyQt5` [47] en vormgegeven met Qt Designer [13]. Qt Designer stelt gebruikers in staat om interface-elementen eenvoudig te slepen en neer te zetten op een canvas. Zodra het ontwerp klaar is, wordt dit automatisch omgezet in Python-code met `PyQt5`-attributen.

De gebruikersinterface bestaat uit vijf tabbladen:

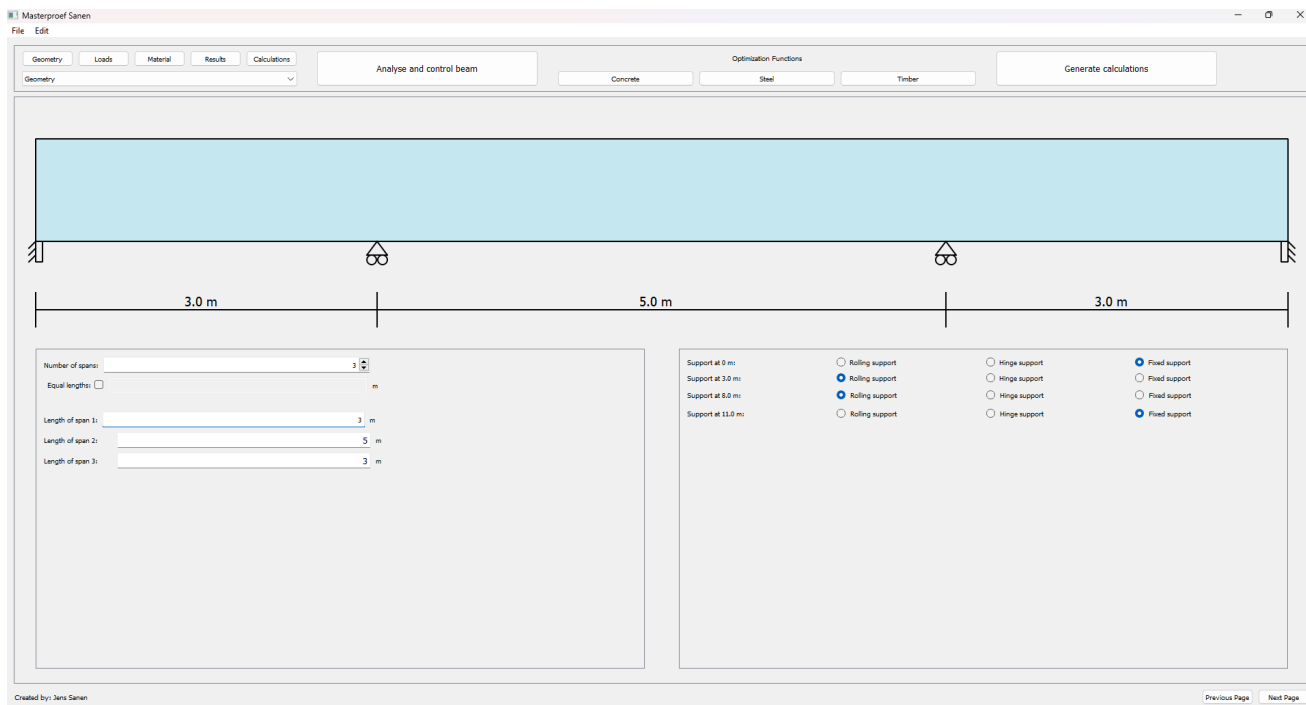
1. **Geometrie en steunpunten:** Hier kunnen gebruikers de geometrie en steunpunten van de balk specificeren.
2. **Belastingsgevallen en -combinaties:** Dit tabblad biedt de mogelijkheid om belastingsgevallen en belastingen te definiëren en toe te voegen. Tevens kunnen hier combinaties van belastingen worden gegenereerd en wordt een overzicht gegeven van de waarde van elk belastingsgeval binnen deze combinaties.
3. **Materiaal:** Dit tabblad is verdeeld in drie secties, elk bedoeld voor een specifiek materiaal. Gebruikers kunnen hier de geometrie van de doorsnede, de maximale waarden en andere materiaalgerelateerde parameters instellen.
4. **Resultaten:** Na het analyseren en controleren van de balk kunnen gebruikers hier de resultaten visualiseren.
5. **Berekeningen:** Dit tabblad biedt een overzicht van de berekeningen voor de belangrijkste controles.

Gebruikers kunnen door de tabbladen navigeren met de knoppen in de bovenste balk van het programma of met de twee knoppen onderaan het scherm (zie Figuur 5.1). In de dropdownmenu van de bovenste balk wordt steeds weergegeven welk tabblad actief is. Deze balk bevat ook een knop om de balk te analyseren en te controleren, drie knoppen voor de optimalisatie van de balk voor elk soort materiaal en een knop om het formulier met de berekeningen en rekenwaarden te genereren.

In de volgende secties worden de tabbladen in detail gepresenteerd en toegelicht.

5.1 Geometrie en steunpunten

Het scherm voor het specificeren van de geometrie van de balk en de steunpunten bevat een tekenveld, een numeriek invoerveld voor het aantal overspanningen, een tekstinvoerveld voor elke overspanning, en een groep van drie knoppen voor elk steunpunt. Dit scherm is weergegeven in Figuur 5.1.



Figuur 5.1: Scherm voor geometrie en steunpunten

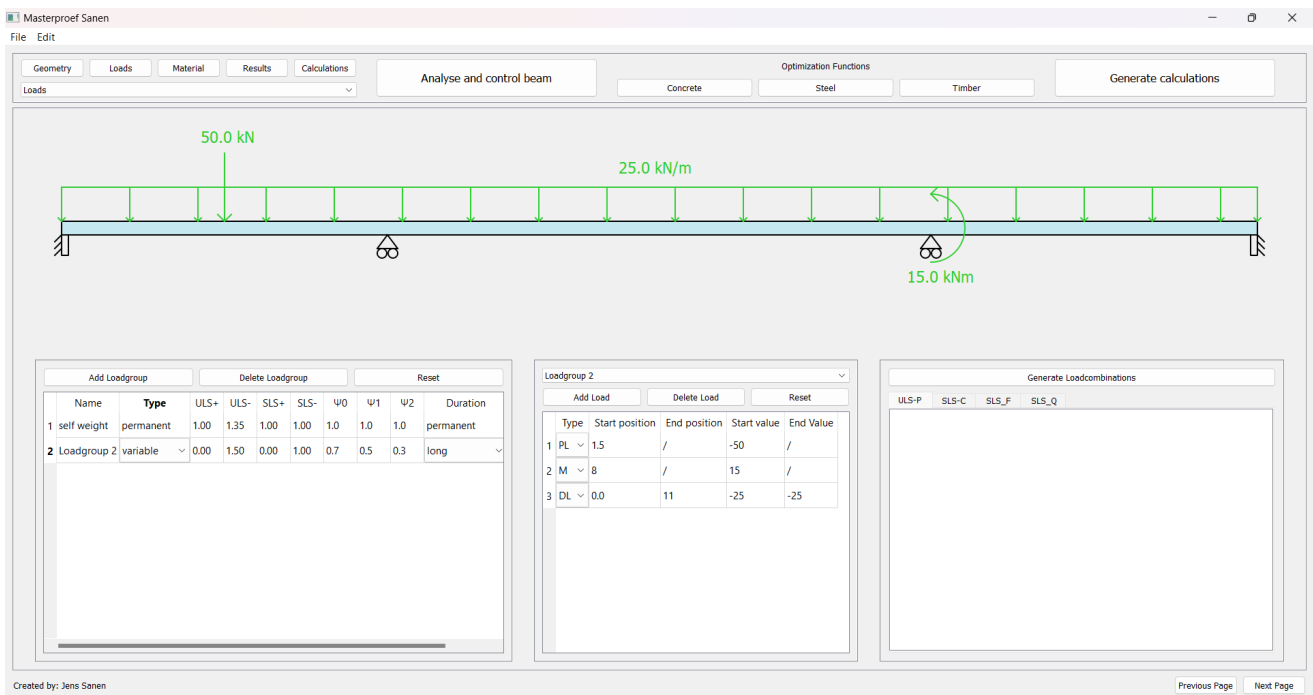
Zoals weergegeven in Figuur 5.1, is er een balk gemodelleerd met drie overspanningen. Bij het opstarten van het programma is er slechts één overspanning en drie knoppen voor het eerste steunpunt aanwezig. Zodra de gebruiker de lengte van deze overspanning invoert, verschijnt er een tweede set knoppen voor het steunpunt aan het einde van de balk. Als de gebruiker het aantal overspanningen verhoogt, genereert het programma extra invoervelden en knoppen.

Het tekenveld toont de afmetingen van de overspanningen en de posities van de steunpunten. Dit tekenveld wordt automatisch bijgewerkt bij wijzigingen van de lengte van een overspanning of het type steunpunt, waarbij de tekening altijd de juiste hoogte en lengte behoudt.

Daarnaast biedt het scherm de mogelijkheid om alle overspanningen dezelfde lengte te geven door het selectievakje naast *equal lengths* aan te vinken en de gewenste lengte in het naastliggende tekstinvoerveld in te voeren.

5.2 Belastingsgevallen, belastingen en combinaties

Het scherm voor het aanmaken van belastingsgevallen, belastingen en belastingscombinaties bestaat uit een tekenveld en drie kolommen. De eerste kolom is bedoeld voor het creëren van belastingsgevallen. De tweede kolom wordt gebruikt om belastingen aan een specifiek belastingsgeval toe te voegen op de balk, terwijl de derde kolom de verschillende waarden van een belastingsgeval voor elke combinatie toont. Dit scherm is weergegeven in Figuur 5.2.



Figuur 5.2: Scherm voor belastingsgevallen, belastingen en belastingscombinaties

In Figuur 5.2 worden belastingen aangebracht op dezelfde balk als weergegeven in Figuur 5.1. Eerst genereert het programma automatisch een belastingsgeval voor het eigengewicht, waarbij de gebruiker geen aanpassingen kan maken of belastingen kan toevoegen. Het eigengewicht wordt berekend op basis van het geselecteerde materiaal en de sectie op het volgende tabblad.

Daarnaast toont Figuur 5.2 een tweede belastingsgeval in de meest linkse tabel. Dit belastingsgeval kan blijvend of veranderlijk zijn, waarbij het programma automatisch de juiste partiële factoren toepast. Voor een veranderlijk belastingsgeval kan de gebruiker de drie combinatiefactoren aanpassen naar de gewenste waarden. Daarnaast biedt het programma de mogelijkheid om het veranderlijk belastingsgeval te specificeren voor nuttige lasten volgens de Nationale Bijlage van Eurocode 0 [28]. Hierbij worden de bijbehorende combinatiefactoren automatisch ingevuld en kunnen niet door de gebruiker worden gewijzigd.

Voor elk belastingsgeval, behalve het eigengewicht, kan de gebruiker een of meerdere belastingen op de balk plaatsen. De gebruiker selecteert het belastingsgeval via het keuzemenu in de middelste kolom en specificeert de belastingen in de tabel eronder. Hierbij kan gekozen worden tussen een puntlast, een moment of een verdeelde belasting. Het programma geeft automatisch aan welke velden ingevuld moeten worden voor elk type belasting. Deze belastingen worden vervolgens in het tekenveld weergegeven. Wanneer de gebruiker wisselt van belastingsgeval, worden de belastingen uit het tekenveld verwijderd om de belastingen van het nieuwe belastingsgeval te tonen.

Tot slot kan de gebruiker met een knop in de derde kolom alle belastingscombinaties genereren op basis van de aangemaakte belastingsgevallen voor UGT-B, BGT-K, BGT-F en BGT-Q. Deze combinaties worden vervolgens in de tabel van de derde kolom weergegeven, met de waarden van elk belastingsgeval binnen de combinaties.

5.3 Materialen

De voorgaande tabbladen zijn gezamenlijk bedoeld voor alle drie de materialen. In het derde tabblad moet de gebruiker echter een keuze maken voor welk materiaal de balk berekend moet worden. Dit kan door bovenaan in het tabblad op één van de drie knoppen te klikken. Wanneer de gebruiker van pagina wisselt, wordt de ingevoerde informatie automatisch opgeslagen, zodat deze behouden blijft bij terugkeer naar de betreffende pagina. In de volgende drie secties wordt de pagina voor elk materiaal besproken.

5.3.1 Beton

De pagina voor beton is opgedeeld in drie segmenten. Het eerste segment is bedoeld voor het specificeren van de doorsnede en materiaaleigenschappen. In het tweede segment kan de gebruiker onder-, bovenwapening en dwarskrachtwapening toevoegen aan de balk. Het derde segment biedt de mogelijkheid om maximale waarden in te voeren die worden gebruikt voor de controle van spanningen, doorbuiging en scheurwijdte. Figuur 5.3 toont het scherm voor beton.

Masterproof Sanen

File Edit

Geometry Loads Material Results Calculations

Analyse and control beam

Optimization Functions

Concrete Steel Timber

Generate calculations

Material: ☒ Concrete ☐ Steel ☐ Timber

Section and material properties:

Section height: 500 mm Concrete strength class: C25/30

Section width: 250 mm Steel strength class: B500S

Concrete cover: 25 mm Exposure class: XD1

Creep factor: 2.15

! Minimum strength class concrete for exposure class C30/37

Reinforcement:

Longitudinal bars

Type	Start position [m]	End position [m]	Amount [-]	Diameter [mm]	As [mm²]
1 continuous	0	11.0	2	20	628.0
2 extra	3	8	1	25	491.0

Maximum values:

Maximum deflection in SLS-Ci: 300 (1/...) Maximum stress concrete in SLS-Ci: 0.55 (1/... x f_{ck}) ! Maximum stress concrete for exposure class in SLS-Ci: 0.60 f_{ck} !

Maximum deflection in SLS-Fi: 300 (1/...) Maximum stress concrete in SLS-Fi: 0.45 (1/... x f_{ck}) ! Maximum stress concrete for exposure class in SLS-Fi: 0.50 f_{ck} !

Maximum deflection in SLS-Qi: 300 (1/...) Maximum stress concrete in SLS-Qi: 0.45 (1/... x f_{ck}) ! Maximum stress for concrete SLS-Qi: 0.45 f_{ck} !

Maximum deflection in SLS-Q with creep: 250 (1/...) Maximum stress reinforcement: 0.75 (1/... x f_{yk}) ! Maximum stress for reinforcement: 0.80 f_{yk} !

Maximum crack width in SLS-Q with creep: 0.4 mm

Maximum crack width for exposure class: 0.3 mm !

Created by: Jens Sanen

Previous Page Next Page

Figuur 5.3: Scherm voor configuratie beton

Zoals te zien is in Figuur 5.3, moet de gebruiker in het eerste segment de hoogte, breedte, betondekking, kruipcoëfficiënt, sterkteklasse voor beton, sterkteklasse voor staal en milieuklasse opgeven. Bij het invoeren van de breedte, hoogte en betondekking wordt automatisch een schets van de doorsnede getekend. Het programma controleert vervolgens of de geselecteerde sterkteklasse voor beton voldoet aan de minimale eisen voor de gekozen milieuklasse. Indien dit niet het geval is, verschijnt er een waarschuwing in het rood die de gebruiker hiervan op de hoogte stelt. Desondanks kan de gebruiker ervoor kiezen om toch met de geselecteerde sterkteklasse door te rekenen. Naast de velden voor betondekking en de kruipcoëfficiënt bevinden zich knoppen waarmee de gebruiker een scherm kan openen om deze waarden te berekenen volgens de formules van Eurocode 2. Het scherm voor de betondekking wordt getoond in Figuur 5.4 en dat voor de kruipcoëfficiënt in Figuur 5.5.

Figuur 5.4: Scherm voor betondekking te bepalen

Figuur 5.5: Scherm voor kruipcoëfficiënt te bepalen

Zoals te zien is in Figuur 5.4, bevat het scherm verschillende knoppen, tekstvelden en keuzemenu's voor het invoeren van de benodigde parameters. Linksonder in het scherm wordt de berekende waarde van de betondekking weergegeven. De gebruiker kan ervoor kiezen om deze berekende waarde in het programma vast te leggen of simpelweg het venster te sluiten.

In Figuur 5.5 wordt een doorsnede van de balk weergegeven, met selectievakjes aan elke zijde. De gebruiker kan hiermee aangeven welke zijden van de doorsnede in contact staan met de lucht. Andere parameters kunnen worden ingevoerd via numerieke velden en een keuzemenu. De berekende kruipcoëfficiënt wordt linksonder in het scherm getoond, en de gebruiker kan besluiten om deze toe te voegen aan het programma of het venster te sluiten zonder de waarde vast te leggen.

In het tweede segment kan de gebruiker wapening plaatsen met behulp van tabellen, zoals weergegeven in Figuur 5.3. Voor de onder- en bovenwapening wordt standaard een staaf over de volledige lengte van de balk geplaatst in de uiterste hoeken, waarbij alleen de diameter van de staven kan worden aangepast. Extra wapening kan worden toegevoegd door op de knop *Add row* boven de tabel te klikken. Hierbij moet de gebruiker de start- en eindpositie, het aantal staven, en de diameter specificeren.

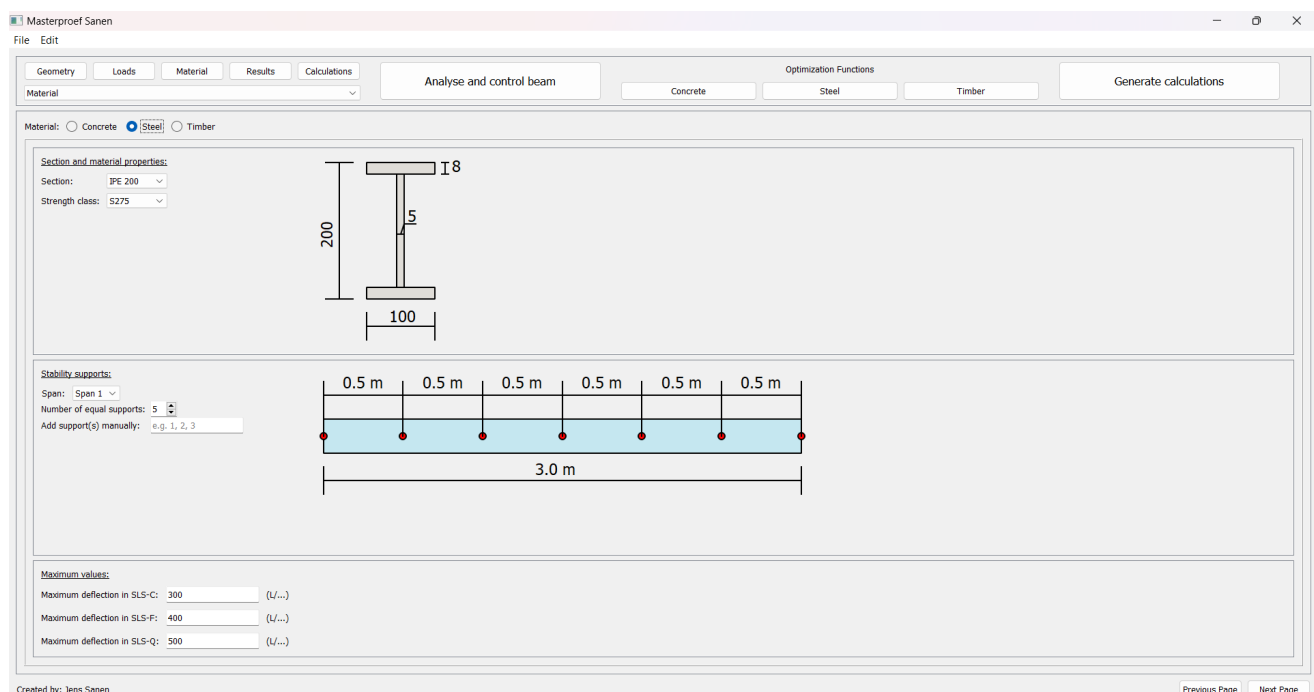
Voor de dwarskrachtwapening wordt in eerste instantie geen wapening geplaatst. De gebruiker kan deze toevoegen door het keuzevak voor dwarskrachtwapening te selecteren en vervolgens een rij in de tabel toe te voegen. Hierbij dienen de start- en eindpositie, de diameter, en de tussenafstand van de wapening te worden ingevoerd.

Rechts van deze tabellen toont het programma een schets van de wapening in de langsrichting, en helemaal rechts wordt een doorsnede van de balk weergegeven. Met de schuifbalk onder de middelste tekening kan de gebruiker door de balk scrollen om andere delen zichtbaar te maken. Het programma identificeert automatisch de posities in de balk waar de wapening verandert, en alleen deze posities zijn met de schuifbalk te selecteren. Op de geselecteerde posities wordt de balk doorgesneden, en de uiterst rechtse tekening toont alleen de wapening die op die doorsnede aanwezig is, met uitzondering van het begin en einde van de balk, waar alle wapening in de doorsnede wordt weergegeven. Zodra de wapening, breedte, hoogte, lengte of betondekking worden gewijzigd, worden de tekeningen en de schuifbalk automatisch bijgewerkt.

In het derde segment wordt de gebruiker gevraagd de factor op te geven waarmee de overspanningslengte moet worden gedeeld om de maximale doorbuiging te berekenen. Dit moet afzonderlijk worden gedaan voor BGT-K, BGT-F, en BGT-Q, en opnieuw voor BGT-Q voor de doorbuiging met kruip. Daarnaast kan in dit segment de maximaal toelaatbare spanning in het beton en de wapening voor BGT-K, BGT-F, en BGT-Q worden ingesteld. Als deze waarde afwijkt van de voorgeschreven waarde volgens de milieuklasse, geeft het programma een waarschuwing. Ook moet de gebruiker de maximale toegestane scheurwijdte specificeren. Als deze waarde niet voldoet aan de eisen van de milieuklasse, zal het programma een waarschuwing geven (zie Figuur 5.3).

5.3.2 Staal

De pagina voor staal is, net als die voor beton, opgedeeld in drie segmenten. Het eerste en derde segment hebben dezelfde functionaliteit als bij beton, maar in het tweede segment kan de gebruiker in plaats van wapening extra steunpunten tegen kip toevoegen. Figuur 5.6 toont het scherm voor staal.



Figuur 5.6: Scherm voor configuratie staal

In het eerste segment selecteert de gebruiker het profiel en de sterkteklassen. Hierbij kan gekozen worden uit IPE-profielen en HEB-profielen. Het programma genereert vervolgens een schets van een I-profiel, waarbij de hoogte, breedte, flensdikte en lijfdikte worden aangegeven.

In het tweede segment kan de gebruiker extra steunpunten tegen kip toevoegen. Dit kan voor elk spant afzonderlijk worden ingesteld. De gebruiker kan kiezen om het aantal steunpunten te specificeren, waarbij het programma automatisch gelijkmatige tussenafstanden berekent, of om zelf de exacte posities van de steunpunten op te geven.

In het derde segment kan de gebruiker, net als bij beton, de factoren invoeren voor het bepalen van de maximale doorbuiging voor BGT-K, BGT-F en BGT-Q.

5.3.3 Hout

De pagina voor hout is opnieuw opgedeeld in drie segmenten, vergelijkbaar met de pagina voor staal, maar met enkele verschillen in de velden van het eerste en derde segment. Figuur 5.7 toont het scherm voor hout.

The screenshot shows the 'Masterproef Sanen' software interface. At the top, there are tabs for 'Geometry', 'Loads', 'Material', 'Results', and 'Calculations'. Below these, there are buttons for 'Analyse and control beam', 'Concrete', 'Steel', and 'Timber' (which is selected). A 'Generate calculations' button is on the right. The 'Material' section has radio buttons for 'Concrete', 'Steel', and 'Timber' (selected). The 'Section and material properties' section includes input fields for 'Section height: 300 mm', 'Section width: 200 mm', 'Strength class: C24', and 'Service class: 2'. A diagram shows a rectangular cross-section with dimensions 300 and 200. The 'Stability supports' section has a 'Span: Span 1' dropdown, 'Number of equal supports: 5', and a text field 'Add support(s) manually: e.g. 1, 2, 3'. A diagram shows a beam with 6 supports (7 points) and 5 equal spans of 0.5 m each, totaling 3.0 m. The 'Maximum values' section has a table with four rows: 'Maximum deflection in SLS-C: 300 (U...)', 'Maximum deflection in SLS-F: 400 (U...)', 'Maximum deflection in SLS-Q: 500 (U...)', and 'Maximum deflection in SLS-Q with creep: 500 (U...)'. At the bottom, it says 'Created by: Jens Sanen' and has 'Previous Page' and 'Next Page' buttons.

Figuur 5.7: Scherm voor configuratie hout

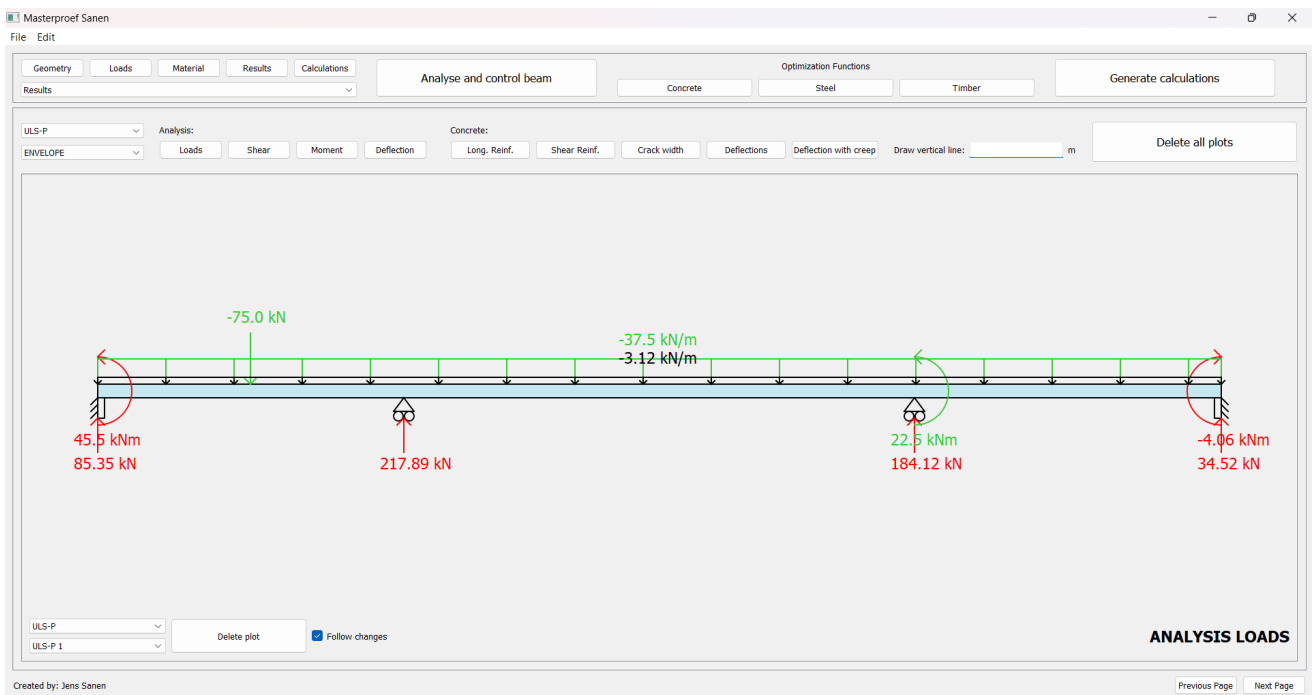
In het eerste segment moet de gebruiker de hoogte en breedte van de doorsnede opgeven. Het programma tekent een schets van de doorsnede en toont deze maten. Bovendien dient de gebruiker de sterkteklasse en klimaatklasse te selecteren.

Het tweede segment is identiek aan dat voor staal. Als de gebruiker steunpunten tegen kip instelt bij staal, worden deze instellingen automatisch overgenomen voor hout, en vice versa. Hierdoor hoeft de gebruiker deze instellingen slechts eenmaal te specificeren als de balk voor beide materialen gecontroleerd moet worden.

In het derde segment kan, naast de maximale factoren voor BGT-K, BGT-F en BGT-Q, ook de maximale factor voor doorbuiging met kruip in BGT-Q worden ingesteld.

5.4 Resultaten

De balk wordt geanalyseerd en gecontroleerd door op de centrale knop bovenaan de pagina te klikken. Dit opent een nieuw venster waarin de gebruiker de grootte van de mesh moet specificeren. Standaard is deze ingesteld op acht per overspanning, met een maximum van twintig om de rekentijd te beperken. Nadat de berekeningen zijn uitgevoerd, kan de gebruiker de gewenste resultaten opvragen. Voor de analyse kan voor elke combinatie de rekenwaarden van de belastingen, de dwarskrachten, het moment, en de vervorming worden geplott. Figuur 5.8 toont een voorbeeld van de opgevraagde belastingswaarden.



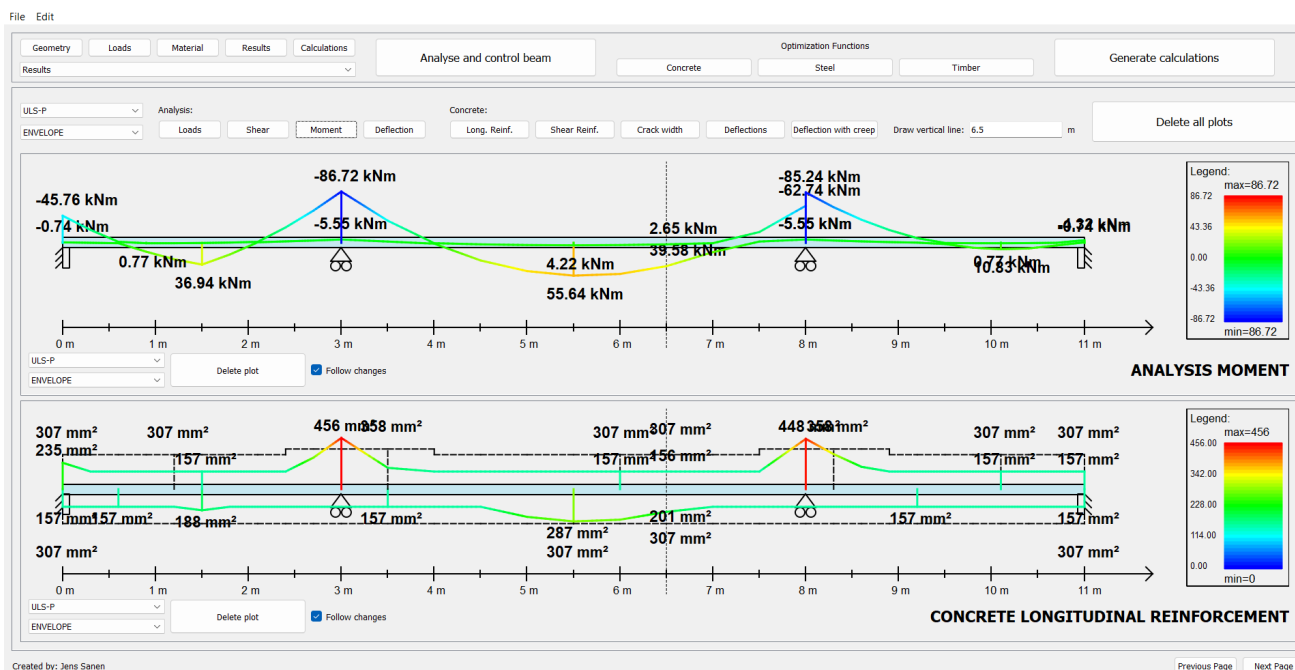
Figuur 5.8: Voorbeeld van resultaten belastingswaarden

In Figuur 5.8 worden de belastingen en krachten in verschillende kleuren weergegeven. De zwarte belasting staat altijd voor het eigengewicht van de balk, terwijl de reactiekrachten in rood worden weergegeven. Elk ander belastingsgeval krijgt een unieke kleur. In dit voorbeeld is er slechts één extra belastingsgeval naast het eigengewicht, dat in groen wordt getekend. Linksonder in het scherm geven de keuzevakken aan voor welke combinatie de huidige tekening geldt, en de gebruiker kan met deze keuzevakken ook tussen combinaties wisselen.

Met de knop naast de keuzevakken kan de gebruiker de resultaten verwijderen. Het aanvinkvakje daarnaast bepaalt of deze resultaten automatisch worden bijgewerkt wanneer de combinatie in de keuzevakken bovenaan de pagina wordt aangepast. Deze keuzevakken bovenaan bepalen voor welke combinatie nieuwe plots worden aangemaakt. Daarnaast kunnen deze keuzevakken bovenaan gebruikt worden om, indien mogelijk, alle al getekende resultaten bij te werken naar de nieuw gekozen combinatie.

Het is ook mogelijk om meerdere resultaten tegelijkertijd te plotten, waardoor de belastingswaarden voor twee verschillende combinaties onder elkaar kunnen worden bekeken. Dit biedt studenten een beter inzicht in hoe belastingscombinaties de waarden van de belastingen beïnvloeden.

Voor elk type materiaal kunnen verschillende resultaten worden opgevraagd. Voor beton kunnen de controles op langs- en dwarskrachtwapening, scheurwijdte, en vervormingen met en zonder kruip worden geraadpleegd. Voor staal en hout kunnen de controles op afschuiving, buiging, sterkte (de hoogste waarde bij afschuiving en buiging op elke positie), stabiliteit, en vervorming worden opgevraagd. Figuur 5.9 toont de resultaten voor de momenten en de controle op langswapening van een betonnen balk.



Figuur 5.9: Resultaten momenten en controle langwapening

In Figuur 5.9 wordt bovenaan de momenten uit de analyse en onderaan de controle van de langwapening weergegeven. Bij de controle van de langs- of hoofdwapening wordt nagegaan of de benodigde wapening kleiner is dan de geplaatste wapening. De resultaten tonen de benodigde wapening met een gekleurde lijn en de geplaatste wapening met stippellijnen. De lijnen bovenaan representeren de wapening aan de bovenkant van de balk, terwijl de lijnen onderaan de wapening aan de onderkant weergeven. Zoals te zien is in de figuur, voldoet de wapening overall, behalve bij de steunpunten, waar er aan de bovenkant van de balk te weinig wapening is. Door deze controle tegelijk met de momentenlijn te plotten, wordt duidelijk waarom op bepaalde posities meer wapening nodig is dan op andere.

Op Figuur 5.9 wordt steeds een enveloppe van de combinaties voor UGT-B weergegeven. Dit betekent dat voor elke positie in de balk de grootste waarde van alle combinaties wordt getoond. Dit verklaart ook waarom er twee momentenlijnen zichtbaar zijn: de ene toont alle maximale waarden, terwijl de andere de minimale waarden toont.

Ten slotte is het mogelijk om een verticale lijn in de resultaten te tekenen, zoals in Figuur 5.9 op de positie van 6,5 meter is gedaan. Dit wordt gedaan door de positie rechtsbovenaan in het tekstinvoerveld in te voeren. Deze lijn helpt bij het gemakkelijker analyseren van de resultaten tussen verschillende getekende waarden en geeft steeds de specifieke waarde op die positie weer.

5.5 Optimalisatie

Nadat een balk is gecontroleerd, kan de gebruiker ervoor kiezen om deze te optimaliseren. Hiervoor moet de gebruiker op een van de drie knoppen bovenaan het scherm klikken, afhankelijk van het type materiaal. Deze knoppen openen nieuwe vensters waarin de gebruiker de optimalisatiecriteria kan instellen. Dit zijn de criteria waarop de balk wordt geoptimaliseerd. In de volgende secties worden de optimalisatieschermen voor elk materiaaltype verder besproken.

5.5.1 Beton

Het optimalisatiescherm voor beton wordt weergegeven in Figuur 5.10.

Reinforcement ☒ Reinforcement and section

Optimization criteria:

ULS-P: ☒ Strength

SLS-C: ☒ Stresses

SLS-F: ☒ Stresses

SLS-Q: ☒ Stresses ☒ Crack Width ☒ Deflection with creep

Longitudinal below

	Type	Start position [m]	End position [m]	Amount [-]	Diameter [mm]	As [mm²]
1	continuous	0.0	11.0	2	14	307

Optimized dimensions: h = 250.0 mm, b = 125.0 mm

Figuur 5.10: Optimalisatiescherm voor beton

Aanvankelijk wordt de tabel in Figuur 5.10 niet weergegeven. Deze tabel verschijnt pas nadat het programma de optimalisatie heeft uitgevoerd. Voor beton kan de gebruiker ervoor kiezen om alleen de wapening te optimaliseren, of zowel de wapening als de doorsnede. Als er echter bij enkel wapeningsoptimalisatie onvoldoende wapening in de balk kan worden geplaatst, geeft het programma een foutmelding en raadt het aan om ook de doorsnede te optimaliseren. De optimalisatiecriteria worden eenvoudig ingesteld door het bijbehorende aankruisvakje te selecteren. Vervolgens kan de balk met de knop *Optimize* worden geoptimaliseerd.

Na de optimalisatie toont het programma de gevonden wapening en, indien van toepassing, de aangepaste afmetingen. Als de gebruiker op de knop *Apply* drukt, worden de wapening en afmetingen in het tabblad voor beton bijgewerkt naar de geoptimaliseerde waarden, zodat de gebruiker nog eventuele wijzigingen kan aanbrengen. De resultaten worden onmiddellijk aangepast aan de nieuwe situatie. Bij de knop *Close* verandert er niets.

5.5.2 Staal

Het optimalisatiescherm van staal wordt weergegeven in Figuur 5.11.

☒ Only IPE ☐ Only HEB ☐ IPE and HEB

Optimization criteria:

ULS-P: ☒ Strength ☒ Stability

SLS-C: ☒ Deflection

SLS-F: ☒ Deflection

SLS-Q: ☒ Deflection

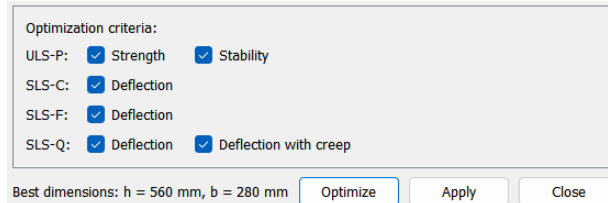
Best section: IPE 240

Figuur 5.11: Optimalisatiescherm voor staal

Bij de optimalisatie voor staal kan de gebruiker ervoor kiezen om alleen IPE-profielen, alleen HEB-profielen, of beide in overweging te nemen in het optimalisatieproces. De optimalisatiecriteria worden opnieuw ingesteld door het bijbehorende aankruisvakje te selecteren. Na het uitvoeren van de optimalisatie toont het programma linksonder het gevonden profiel. De gebruiker kan vervolgens beslissen om het profiel al dan niet toe te passen.

5.5.3 Hout

Het optimalisatiescherm voor hout wordt weergegeven in Figuur 5.12.



Figuur 5.12: Optimalisatiescherm voor hout

Bij de optimalisatie van houten balken hoeft de gebruiker alleen de gewenste optimalisatiecriteria te selecteren. Nadat de optimalisatie is uitgevoerd, kan de gebruiker ervoor kiezen om de voorgestelde doorsnede toe te passen of niet.

5.6 Berekeningen

Het tabblad *Calculations* biedt toegang tot de berekeningen, inclusief formules en rekenwaarden voor het meest kritieke punt voor elke controle. Eerst worden de gegevens samengevat, gevolgd door de controles. Elk materiaal heeft een eigen formulier. Figuur 5.13 toont de controle van de langswapening in een betonnen balk, en Figuur 5.14 toont de stabiliteitscontrole bij hout.

Masterproef Sanen

File Edit

Geometry Loads Material Results Calculations

Calculations

Analyse and control beam Concrete Steel Timber Generate calculations

Area of concrete in tension $A_{ct} = b \times \frac{h}{2} = 250 \text{ mm} \times \frac{500 \text{ mm}}{2} = 62500 \text{ mm}^2$

Factor for height $k = 1.21 - 0.7 \times \frac{h}{1000} = 1.21 - 0.7 \times \frac{500}{1000} = 0.86$

Minimum reinforcement for crack control $A_{s,min,crackcontrol} = \frac{k_s \times k \times f_{ctm} \times A_{ct}}{f_{yk}} = \frac{0.4 \times 0.86 \times 2.6 \text{ N/mm}^2 \times 62500 \text{ mm}^2}{500 \text{ N/mm}^2} = 112 \text{ mm}^2$

Minimum reinforcement $A_{s,min} = \max(A_{s,min,abs}; A_{s,min,brittle}; A_{s,min,crackcontrol}) = \max(149 \text{ mm}^2; 154 \text{ mm}^2; 112 \text{ mm}^2) = 154 \text{ mm}^2$

Control of Longitudinal Reinforcement:

Design value of moment at 2.5 m $M_{Ed} = 106.93 \text{ kNm}$

Effective depth at 2.5 m $d = 457 \text{ mm}$

Provided reinforcement at 2.5 m $A_{s,prov} = 628 \text{ mm}^2$

Design value of concrete compressive strength $f_{cd} = \alpha_{cc} \times \frac{f_{ck}}{\gamma_c} = 0.85 \times \frac{25 \text{ N/mm}^2}{1.5} = 14.17 \text{ N/mm}^2$

Design value of reduced bending moment $\mu_d = \frac{M_{Ed}}{b \times d^2 \times f_{cd}} = \frac{106.93 \times 10^6 \text{ Nmm}}{250 \text{ mm} \times (457 \text{ mm})^2 \times 14.17 \text{ N/mm}^2} = 0.1446$

Ratio lever arm to effective depth $\zeta = 0.9167$ (Interpolation in Table 4.2.3.7a)

Design value of stress in reinforcement $\sigma_s = 435 \text{ N/mm}^2$ (Interpolation in Table 4.2.3.7a)

Calculated reinforcement $A_{s,calculated} = \frac{M_{Ed}}{\zeta \times d \times \sigma_s} = \frac{106.93 \times 10^6 \text{ Nmm}}{0.9167 \times 457 \text{ mm} \times 435 \text{ N/mm}^2} = 586 \text{ mm}^2$

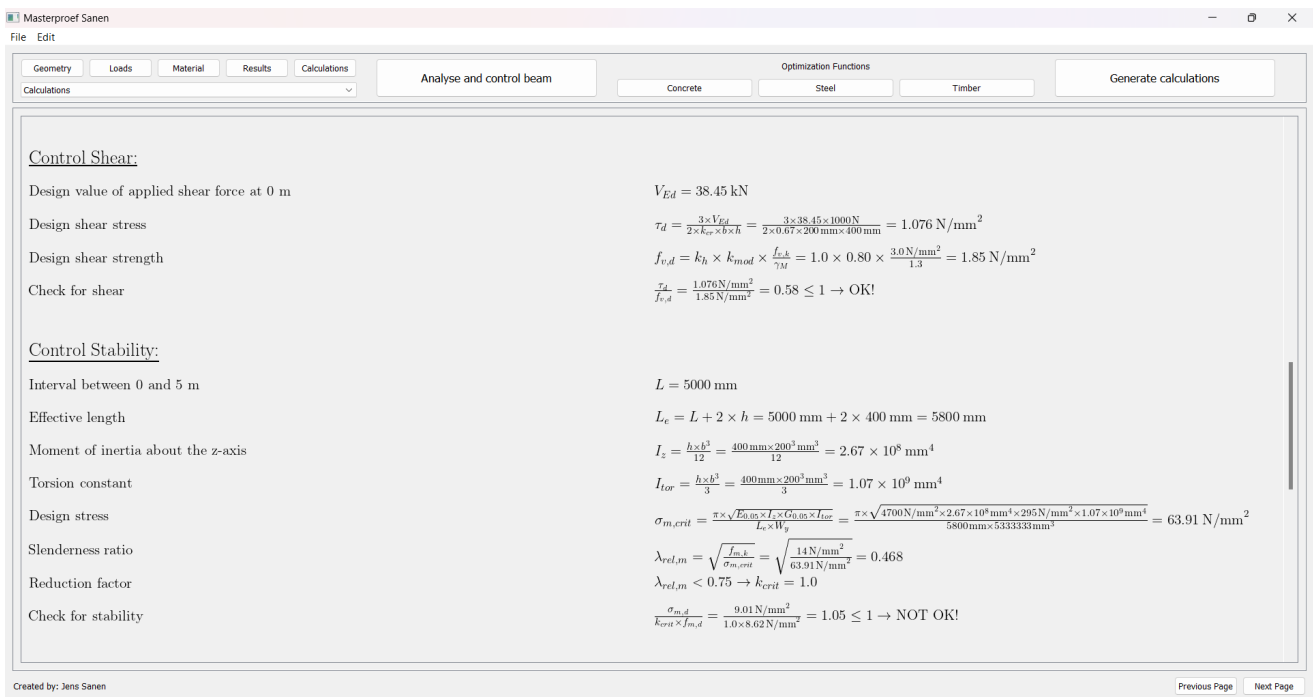
Required reinforcement $A_{s,req} = \max(A_{s,min}; A_{s,calculated}) = \max(154 \text{ mm}^2; 586 \text{ mm}^2) = 586 \text{ mm}^2$

Check if provided reinforcement is sufficient $A_{s,prov} \geq A_{s,req} = 628 \text{ mm}^2 \geq 586 \text{ mm}^2 \Rightarrow \text{OK}$

Created by: Jens Sanen

Previous Page Next Page

Figuur 5.13: Voorbeeld berekeningen beton



Figuur 5.14: Voorbeeld berekeningen hout

Zoals weergegeven in Figuren 5.13 en 5.14, wordt aan de linkerkant aangegeven wat er wordt berekend. Daarop volgen de grootte, de bijbehorende formule in grootheden, de formule met rekenwaarden, en tenslotte de uitkomst. Deze opzet maakt het mogelijk voor gebruikers om de waarden in de resultaten te begrijpen en na te gaan, mocht dit niet direct duidelijk zijn.

Het genereren van deze formulieren vereist enige tijd. Dit komt doordat PyQt5 geen directe ondersteuning biedt voor het weergeven van wiskundige formules. Om dit te verhelpen, wordt een LaTeX [48] bibliotheek gebruikt om de formules te renderen en op te slaan als afbeeldingen. Deze afbeeldingen worden vervolgens op de juiste positie in de interface ingeladen. Omdat dit proces enige tijd vergt en niet altijd noodzakelijk is, is er knop in de bovenste balk voorzien om het formulier te genereren. Op deze manier wordt de tijd die nodig is voor het analyseren en controleren van een balk zo minimaal gehouden.

In bijlagen G, H en I zijn respectievelijk de volledige formulieren voor beton, staal en hout opgenomen.

Hoofdstuk 6

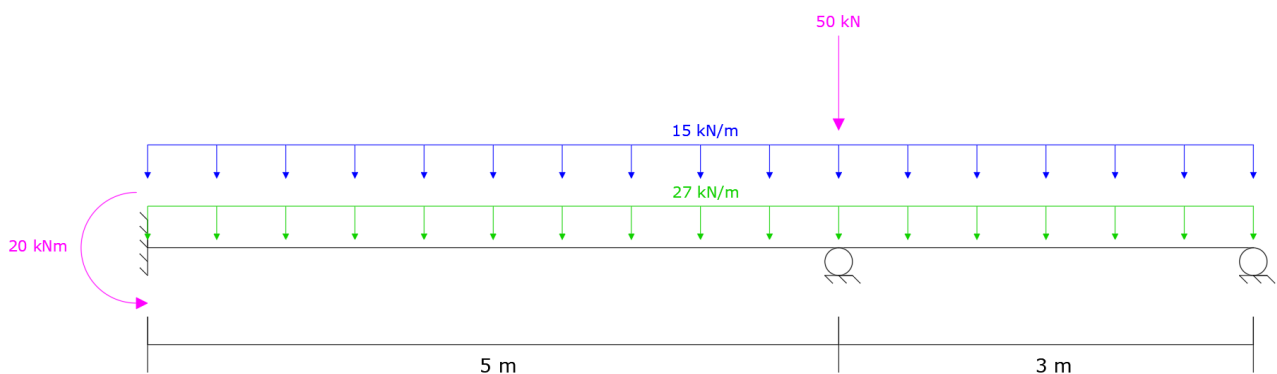
Uitgewerkt voorbeeld

In dit hoofdstuk wordt een eenzelfde balk geanalyseerd en gecontroleerd voor drie verschillende materialen. De resultaten hiervan worden vergeleken met die van het bestaande programma Buildsoft Diamonds [5].

De randvoorwaarden voor de balk zijn als volgt: het betreft een balk met twee overspanningen, met een totale lengte van acht meter. De eerste overspanning is vijf meter lang en de tweede drie meter. Het eerste steunpunt is een inklemming, terwijl de andere twee steunpunten rolopleggingen zijn. Er worden vier belastingsgevallen onderscheiden:

1. Eigengewicht van de balk
2. Blijvend belastingsgeval
3. Nuttige last voor een commercieel gebouw ($\psi_0 = 0,7$; $\psi_1 = 0,7$; $\psi_2 = 0,6$)
4. Overige veranderlijk belastingsgeval ($\psi_0 = 0,7$; $\psi_1 = 0,5$; $\psi_2 = 0,3$)

Het programma berekent automatisch de belasting voor het eerste belastingsgeval. In het blijvende belastingsgeval is er een gelijkmatig verdeelde belasting van 27 kN/m over de volledige lengte van de balk, terwijl de nuttige last een verdeelde belasting van 15 kN/m veroorzaakt. Ten slotte is er in het overige belastingsgeval een moment van 20 kNm aanwezig bij het eerste steunpunt en een puntlast van 50 kN op het middelste steunpunt. Figuur 6.1 geeft een weergave van de balk.



Figuur 6.1: Schets configuratie uitgewerkt voorbeeld

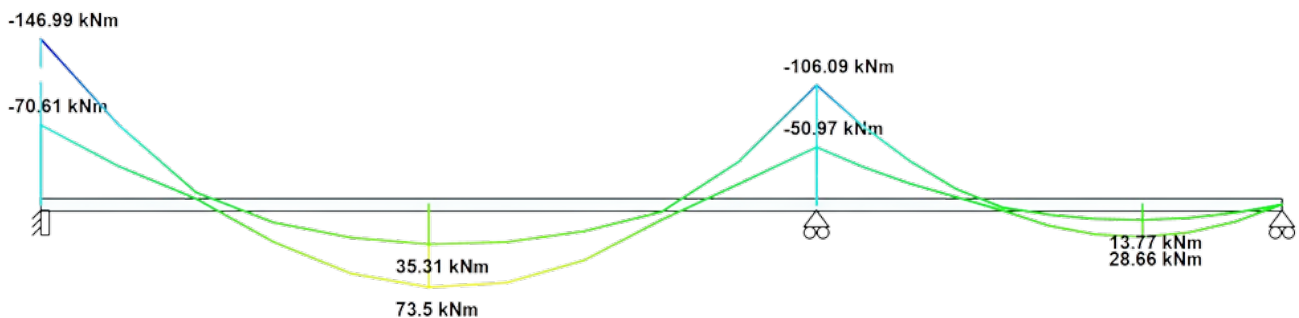
In de volgende sectie worden de resultaten van de analyse vergeleken met die van Diamonds. De daaropvolgende drie secties behandelen respectievelijk de resultaten voor beton, staal en hout.

6.1 Analyse

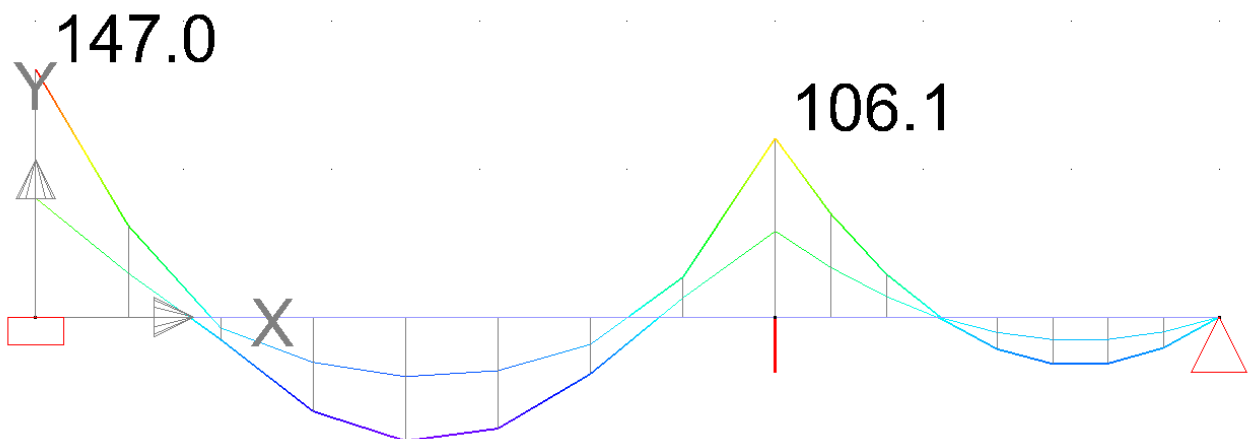
Om de resultaten van de analyse te bespreken, is een betonnen balk met een breedte van 300 mm en een hoogte van 500 mm geanalyseerd. De balk heeft een eigen gewicht van 3,75 kN/m. In beide programma's zijn de belastingscombinaties voor UGT-B, BGT-K, BGT-F, en BGT-Q automatisch gegenereerd. Beide programma's hebben voor deze criteria een gelijk aantal combinaties bepaald: twintig voor UGT-B, vijf voor BGT-K en BGT-F, en vier voor BGT-Q. De analyse in beide programma's is uitgevoerd met een net van acht per overspanning.

6.1.1 Moment

Het bespreken van de momentenlijn of andere resultaten voor elke combinatie van de criteria is niet mogelijk. Daarom wordt steeds één combinatie gekozen om te bespreken, waarbij de nadruk nu ligt op de enveloppe-momentenlijn voor UGT-B. Figuur 6.2 toont de momentenlijn van het in deze masterproef ontwikkelde programma, terwijl Figuur 6.3 de momentenlijn in Diamonds weergeeft.



Figuur 6.2: Momentendiagram van ontwikkeld programma

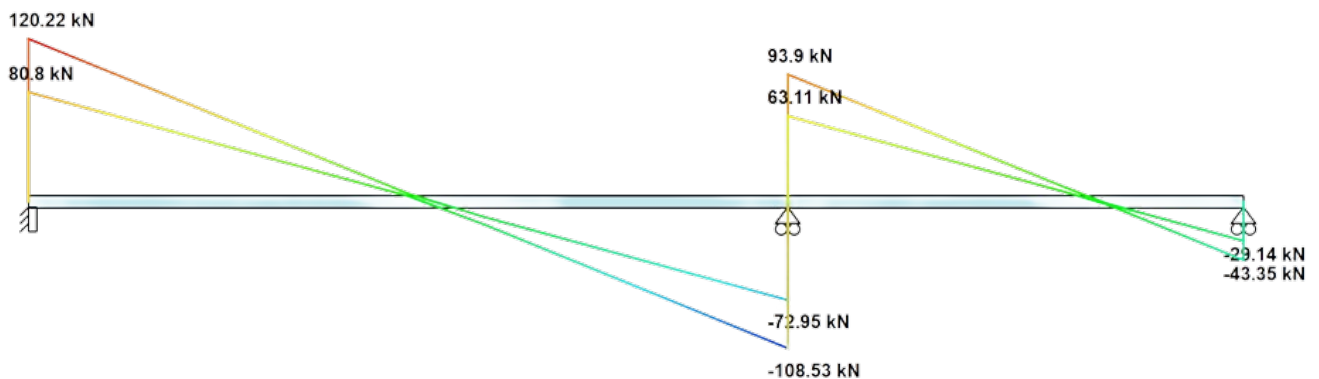


Figuur 6.3: Momentendiagram van Diamonds

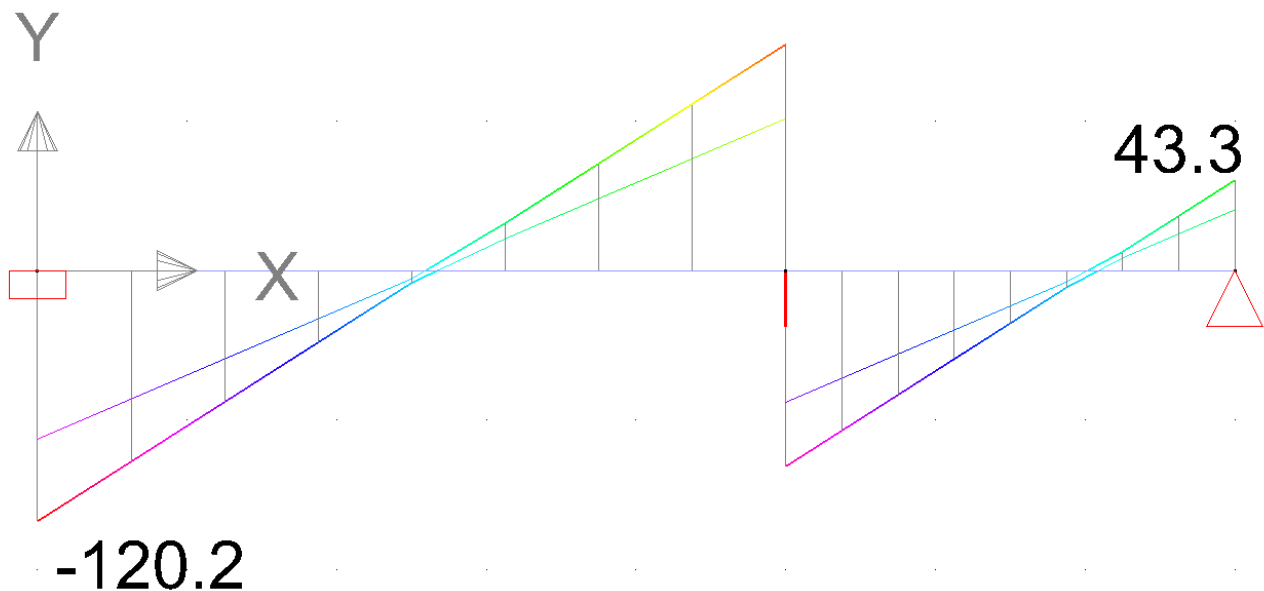
Zoals te zien is in Figuur 6.2 en Figuur 6.3, hebben beide momentenlijnen een vergelijkbaar verloop. De waarden van de momenten zijn ook gelijk, alleen het teken verschilt.

6.1.2 Dwarskracht

Voor de dwarskracht worden de enveloppe-diagrammen voor BGT-K met elkaar vergeleken. Figuur 6.4 toont het dwarskrachtdiagram van het ontwikkelde programma, en Figuur 6.5 toont het dwarskrachtdiagram van Diamonds.



Figuur 6.4: Dwarskrachtendiagram van ontwikkeld programma

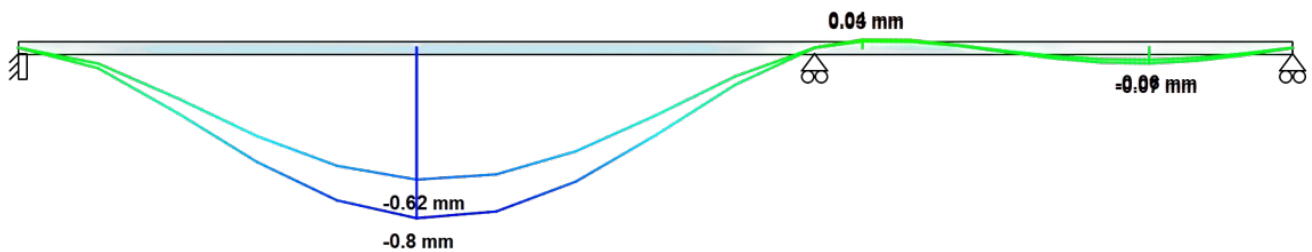


Figuur 6.5: Dwarskrachtendiagram van Diamonds

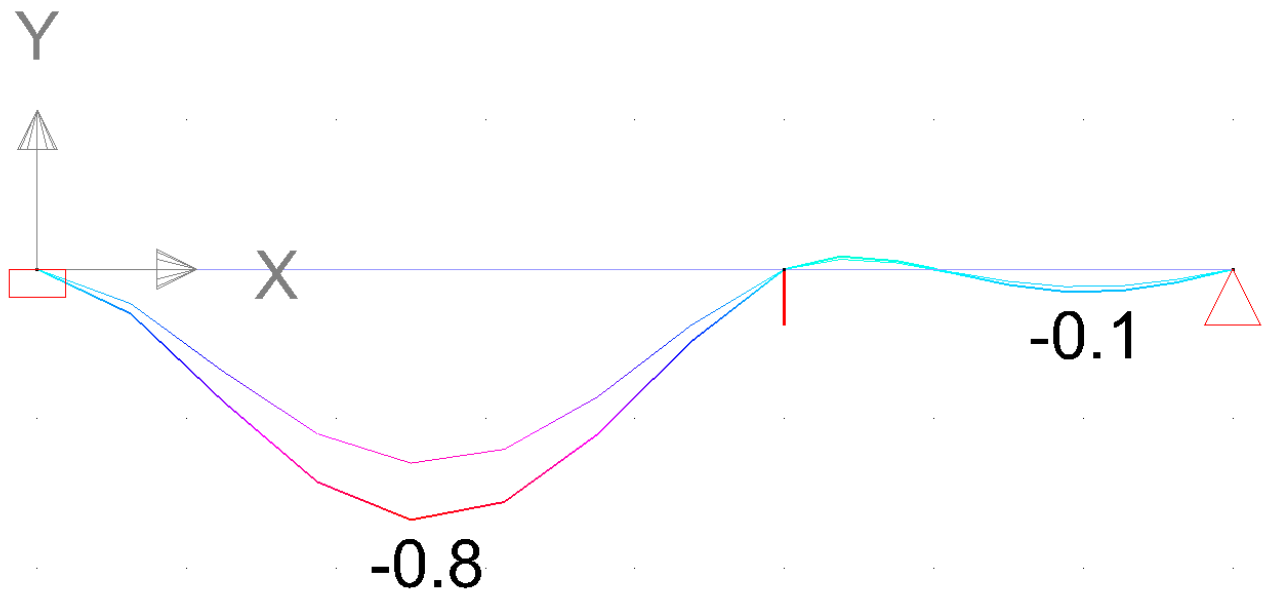
Door Figuur 6.4 en Figuur 6.5 te vergelijken, lijken de diagrammen op het eerste gezicht verschillend. Echter, dit is niet het geval, de waarden zijn alleen gespiegeld rond de x-as. De waarden komen weer overeen, maar het teken verschilt. Dit is echter geen probleem omdat het teken van de dwarskracht geen invloed heeft op de berekeningen.

6.1.3 Vervorming

Ten slotte worden voor de analyse ook de berekende vervormingen met elkaar vergeleken. Dit wordt gedaan voor de enveloppe in BGT-Q. Figuur 6.6 en Figuur 6.7 tonen respectievelijk de vervormingen van het ontwikkelde programma en van Diamonds.



Figuur 6.6: Vervormingen van ontwikkeld programma



Figuur 6.7: Vervormingen van Diamonds

De vervormingen in Figuur 6.6 vertonen hetzelfde verloop en dezelfde waarden als de vervormingen in Figuur 6.7.

6.1.4 Conclusie

Alle waarden van de verschillende grootheden zijn identiek voor het ontwikkelde programma en Diamonds. Dit bevestigt dat de belastingscombinaties correct worden aangemaakt en dat de eindige-elementenmethode juist wordt uitgevoerd.

6.2 Beton

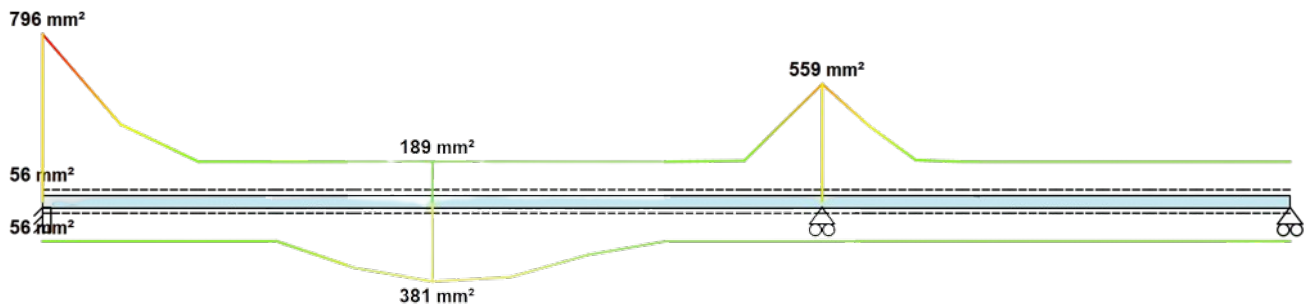
In deze sectie wordt een betonnen balk controleerd op basis van de benodigde langs- en dwarskrachtwapening, toelaatbare spanningen, scheurwijdte, en vervorming met kruip. Bij de berekeningen voor de benodigde wapening wordt aanvankelijk geen wapening in de balk geplaatst, omdat dit geen invloed heeft op de uitkomsten en bovendien niet mogelijk is binnen Diamonds. Vervolgens wordt, op basis van de verkregen resultaten in beide softwareprogramma's, dezelfde hoeveelheid wapening aangebracht voordat de controles op spanning, scheurwijdte en vervorming met kruip worden uitgevoerd.

Voor deze controles wordt een balk gebruikt met een breedte van 300 mm en een hoogte van 500 mm. In het ontwikkelde programma is de betondekking vastgesteld op 30 mm, terwijl in

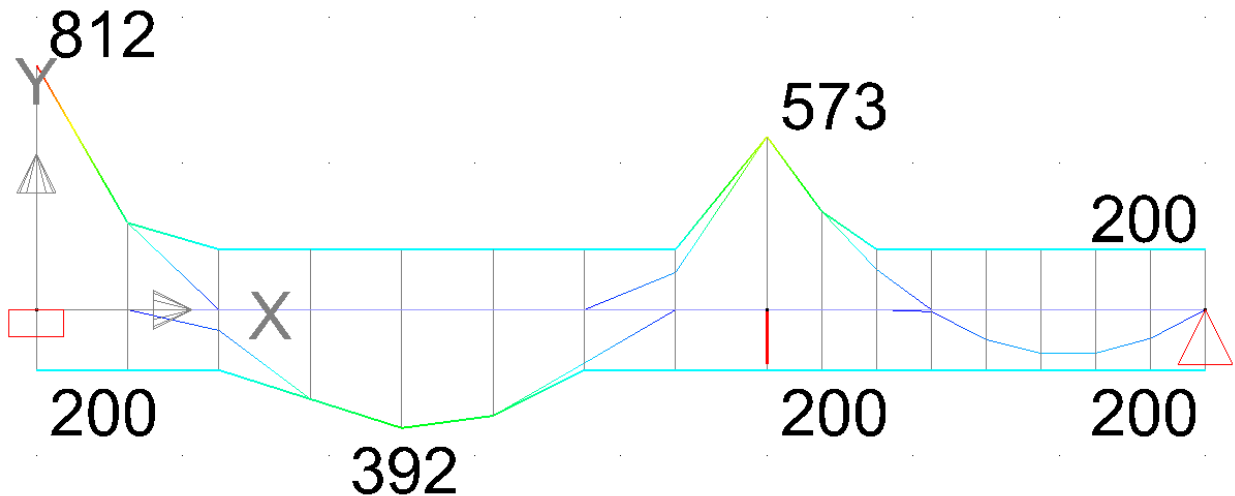
Diamonds deze wordt ingesteld op 33 mm. Dit verschil ontstaat doordat in Diamonds de betondekking wordt berekend als de afstand tot het zwaartepunt van de primaire wapening, terwijl in het ontwikkelde programma de afstand tot de eerste wapening wordt gehanteerd. In het ontwikkelde programma worden minimaal twee staven met een diameter van zes mm gebruikt, waarbij dus de helft van deze diameter wordt opgeteld bij de betondekking om dezelfde betondekking te verkrijgen. Hierdoor is de betondekking in Diamonds aangepast naar 33 mm. Voor de berekeningen wordt uitgegaan van betonkwaliteit C25/30 en staalkwaliteit BE500S.

6.2.1 Langswapening

De benodigde boven- en onderwapening volgens het ontwikkelde programma wordt weergegeven in Figuur 6.8, en voor Diamonds in Figuur 6.9.



Figuur 6.8: Benodigde langswapening volgens ontwikkeld programma



Figuur 6.9: Benodigde langswapening volgens Diamonds

Uit Figuren 6.8 en 6.9 blijkt dat de resultaten een vergelijkbaar patroon vertonen. Het verschil in waarden tussen de pieken is minder dan 3% en kan als verwaarloosbaar worden beschouwd. Desondanks worden de berekeningen volgens de site van Diamonds [49] in formules (6.1) tot en met (6.4) verder uitgewerkt om mogelijke oorzaken van dit verschil te onderzoeken.

$$d = h - c = 500 \text{ mm} - 33 \text{ mm} = 467 \text{ mm} \quad (6.1)$$

$$\mu_d = \frac{M_{Ed}}{b \times d^2 \times \frac{\alpha \times f_{ck}}{\gamma_c}} = \frac{147 \times 10^6 \text{ Nmm}}{300 \text{ mm} \times (467 \text{ mm})^2 \times \frac{0,85 \times 25 \text{ N/mm}^2}{1,5}} = 0.1586 \quad (6.2)$$

$$\omega = 0,1762 + \frac{0,1642 - 0,1762}{0,15 - 0,16} \times (0,1586 - 0,16) = 0,1745 \quad (6.3)$$

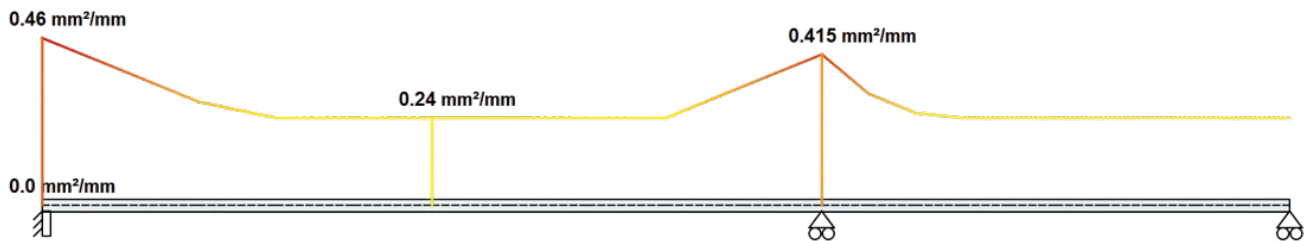
$$A_s = \omega \times b \times d \times \frac{\frac{\alpha \times f_{ck}}{\gamma_c}}{\frac{f_{yk}}{\gamma_s}} = 0,1745 \times 300 \text{ mm} \times 467 \text{ mm} \times \frac{\frac{0,85 \times 25 \text{ N/mm}^2}{1,5}}{\frac{500 \text{ N/mm}^2}{1,15}} = 797 \text{ mm}^2 \quad (6.4)$$

De uitkomst van deze berekening komt dicht bij het resultaat van het ontwikkelde programma dan bij dat van Diamonds. De precieze oorzaak van dit verschil blijft echter onduidelijk.

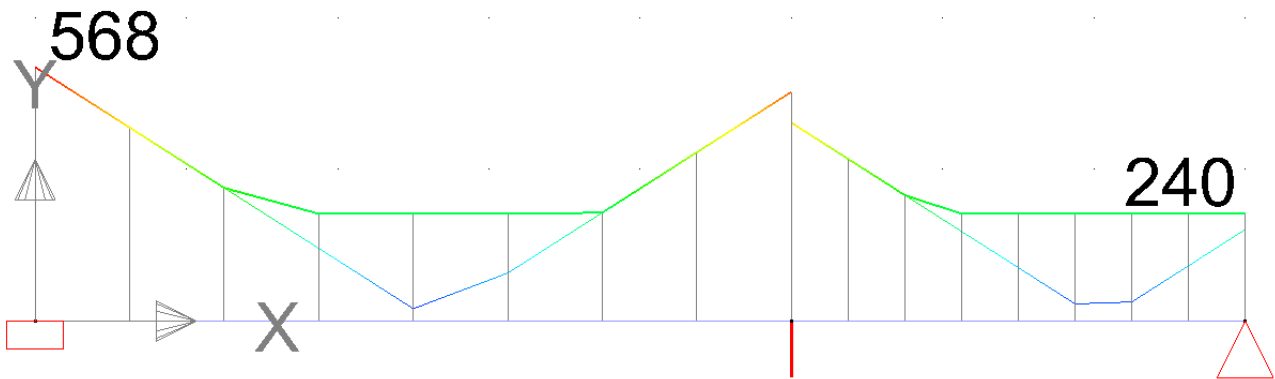
Het verschil in de minimale wapening tussen het ontwikkelde programma en Diamonds bedraagt slechts 6%. Dit verschil wordt verklaard doordat Diamonds de minimale langwapening berekent op basis van de hoogte van de balk, in plaats van de nuttige hoogte, zoals in het ontwikkelde programma wordt gedaan [49].

6.2.2 Dwarskrachtwapening

De resultaten voor de benodigde dwarskrachtwapening wordt getoond in Figuur 6.4 voor het ontwikkelde programma en in Figuur 6.5 voor Diamonds.



Figuur 6.10: Benodigde dwarskrachtwapening volgens ontwikkeld programma



Figuur 6.11: Benodigde dwarskrachtwapening volgens Diamonds

Het eerste dat opvalt bij het vergelijken van Figures 6.10 en 6.11 is de orde van grootte van de getallen. In het ontwikkelde programma worden deze waarden kleiner dan één weergegeven, terwijl in Diamonds de waarden in de honderden liggen. Dit verschil ontstaat doordat in het ontwikkelde programma de benodigde dwarskrachtwapening wordt uitgedrukt in mm^2/mm , terwijl Diamonds dit doet in mm^2/m .

Een tweede opvallend punt is het verschil in de piekwaarden, dat 23,5% bedraagt. Dit verschil is te verklaren door de schoorhoek waarmee wordt gerekend. volgens de website van Diamonds [50]

gebruikt Diamonds een schoorhoek van 31° , terwijl het ontwikkelde programma een hoek van $26,6^\circ$ hanteert. Ter illustratie van dit verschil worden beide berekeningen getoond. Formule (6.6) geeft de berekening voor het ontwikkelde programma, terwijl Formule (6.7) de berekening voor Diamonds laat zien.

$$\frac{A_{sw}}{s} = \frac{V_{Ed}}{0,9 \times d \times f_{ywd} \times \cot \theta} \quad (6.5)$$

$$\left(\frac{A_{sw}}{s} \right)_{\text{programma}} = \frac{168,2 \times 1000 \text{ N}}{0,9 \times 467 \text{ mm} \times \frac{500 \text{ N/mm}^2}{1,15} \times \cot (26,6^\circ)} = 0,461 \text{ mm}^2/\text{mm} \quad (6.6)$$

$$\left(\frac{A_{sw}}{s} \right)_{\text{Diamonds}} = \frac{168,2 \times 1000 \text{ N}}{0,9 \times 467 \text{ mm} \times \frac{500 \text{ N/mm}^2}{1,15} \times \cot (31^\circ)} = 0,553 \text{ mm}^2/\text{mm} \quad (6.7)$$

De minimum hoeveelheid minimum dwarskrachtwapening wordt op dezelfde manier berekend en bedraagt ook identiek dezelfde waarde.

6.2.3 Spanningen

In Diamonds is het niet mogelijk om de spanningen in het beton en de wapeningsstaven te berekenen. Het ontwikkelde programma berekent en controleert deze spanningen echter wel, maar biedt geen mogelijkheid om deze te plotten. Wel is het mogelijk om de berekening van deze controle op te vragen.

Voordat deze berekening kan worden weergegeven, moet er wapening in de balk worden aangebracht. In dit voorbeeld zijn er aan de onderzijde twee staven met een diameter van zestien millimeter ($A_s = 402, \text{mm}^2$), en aan de bovenzijde twee staven met een diameter van twintig millimeter en één staaf met een diameter van zestien millimeter ($A_s = 829, \text{mm}^2$). Daarnaast is de waarde van de kruipcoëfficiënt nodig om de spanningen te berekenen. Deze wordt verondersteld gelijk te zijn aan twee, maar kan niet in Diamonds worden ingesteld.

De spanning wordt berekend ter hoogte van de inklemming, omdat daar het moment het grootst is en voor BGT-Q omdat deze waarden nog nodig zijn voor de berekeningen van de scheurwijdte en vervorming met kruip. Voor deze berekeningen moet de betondrukhoogte en het traagheidsmoment van de niet- en wel-gescheurde doorsnede bekend zijn. Deze worden automatisch berekend in het programma en zijn weergegeven in Figuur 6.12. Zodra deze parameters zijn vastgesteld, kunnen de spanningen bij de inklemming worden berekend. Deze spanningen worden ook in het programma berekend en weergegeven in Figuur 6.13.

$$x_{\text{uncracked,creep}} = \frac{\frac{b \times h^2}{2} + \alpha_{\text{creep}} \times (A_{s1} \times d + A_{s2} \times d_2)}{A_{s1} \times \alpha_{\text{creep}} + A_{s2} \times \alpha_{\text{creep}} + b \times h}$$

$$x_{\text{uncracked,creep}} = \frac{\frac{300 \text{ mm} \times (500 \text{ mm})^2}{2} + 19.35 \times (829 \text{ mm}^2 \times 452 \text{ mm} + 402 \text{ mm}^2 \times 46 \text{ mm})}{829 \text{ mm}^2 \times 19.35 + 402 \text{ mm}^2 \times 19.35 + 300 \text{ mm} \times 500 \text{ mm}} = 259 \text{ mm}$$

$$I_{\text{uncracked,creep}} = \frac{b \times x_{\text{uncracked,creep}}^3}{3} + \frac{b \times (h - x_{\text{uncracked,creep}})^3}{3} + \alpha_{\text{creep}} \times A_{s1} \times (d - x_{\text{uncracked,creep}})^2 + \alpha_{\text{creep}} \times A_{s2} \times (x_{\text{uncracked,creep}} - d_2)^2$$

$$I_{\text{uncracked,creep}} = \frac{300 \text{ mm} \times (259 \text{ mm})^3}{3} + \frac{300 \text{ mm} \times (500 \text{ mm} - 259 \text{ mm})^3}{3} + 19.35 \times 829 \text{ mm}^2 \times (452 \text{ mm} - 259 \text{ mm})^2 + 19.35 \times 402 \text{ mm}^2 \times (259 \text{ mm} - 46 \text{ mm})^2$$

$$I_{\text{uncracked,creep}} = 4.14 \times 10^9 \text{ mm}^4$$

$$M_{cr} = \frac{I_{\text{uncracked,creep}} \times f_{ctm}}{h - x_{\text{uncracked,creep}}} = \frac{4.14 \times 10^9 \text{ mm}^4 \times 2.6 \text{ N/mm}^2}{500 \text{ mm} - 259 \text{ mm}} = 44 \text{ kNm}$$

$$x_{\text{cracked,creep}} = \frac{-\alpha_{\text{creep}} \times (A_{s1} + A_{s2})}{b} + \sqrt{\left(\frac{\alpha_{\text{creep}} \times (A_{s1} + A_{s2})}{b} \right)^2 + \frac{2 \times \alpha_{\text{creep}} \times (A_{s1} \times d + A_{s2} \times d_2)}{b}}$$

$$x_{\text{cracked,creep}} = \frac{-19.35 \times (829 \text{ mm}^2 + 402 \text{ mm}^2)}{300 \text{ mm}} + \sqrt{\left(\frac{19.35 \times (829 \text{ mm}^2 + 402 \text{ mm}^2)}{300 \text{ mm}} \right)^2 + \frac{2 \times 19.35 \times (829 \text{ mm}^2 \times 452 \text{ mm} + 402 \text{ mm}^2 \times 46 \text{ mm})}{300 \text{ mm}}} = 159 \text{ mm}$$

$$I_{\text{cracked,creep}} = \frac{b \times x_{\text{cracked,creep}}^3}{3} + \alpha_{\text{creep}} \times A_{s1} \times (d - x_{\text{cracked,creep}})^2 + \alpha_{\text{creep}} \times A_{s2} \times (x_{\text{cracked,creep}} - d_2)^2$$

$$I_{\text{cracked,creep}} = \frac{300 \text{ mm} \times (162 \text{ mm})^3}{3} + 19.35 \times 829 \text{ mm}^2 \times (452 \text{ mm} - 162 \text{ mm})^2 + 19.35 \times 402 \text{ mm}^2 \times (162 \text{ mm} - 46 \text{ mm})^2 = 1.88 \times 10^9 \text{ mm}^4$$

Figuur 6.12: Berekeningen van sectie grootheden met kruip

$$M_{Ed} = 91.3 \text{ kNm}$$

$$M_{Ed} > M_{cr} = 91.3 \text{ kNm} > 44 \text{ kNm} \Rightarrow \text{CRACKED}$$

$$\sigma_c = \frac{M_{Ed} \times (x_{\text{cracked,creep}})}{I_{\text{cracked,creep}}} = \frac{91.3 \times 10^6 \text{ Nmm} \times 162 \text{ mm}}{1.88 \times 10^9 \text{ mm}^4} = 7.7 \text{ N/mm}^2$$

$$\sigma_{s1} = \frac{d - x_{\text{cracked,creep}}}{x_{\text{cracked,creep}}} \times \sigma_c \times \alpha_{\text{creep}} = \frac{452 \text{ mm} - 162 \text{ mm}}{162 \text{ mm}} \times 7.7 \text{ N/mm}^2 \times 19.35 = 275 \text{ N/mm}^2$$

$$\sigma_{s2} = \frac{x_{\text{cracked,creep}} - d_2}{x_{\text{cracked,creep}}} \times \sigma_c \times \alpha_{\text{creep}} = \frac{162 \text{ mm} - 46 \text{ mm}}{162 \text{ mm}} \times 7.7 \text{ N/mm}^2 \times 19.35 = 106 \text{ N/mm}^2$$

$$\sigma_c \leq \sigma_{c,max,SLS-Q} = 7.7 \text{ N/mm}^2 \leq 11.25 \text{ N/mm}^2 \Rightarrow \text{OK}$$

$$\sigma_{s1} \leq \sigma_{s,max} = 275 \text{ N/mm}^2 \leq 400 \text{ N/mm}^2 \Rightarrow \text{OK}$$

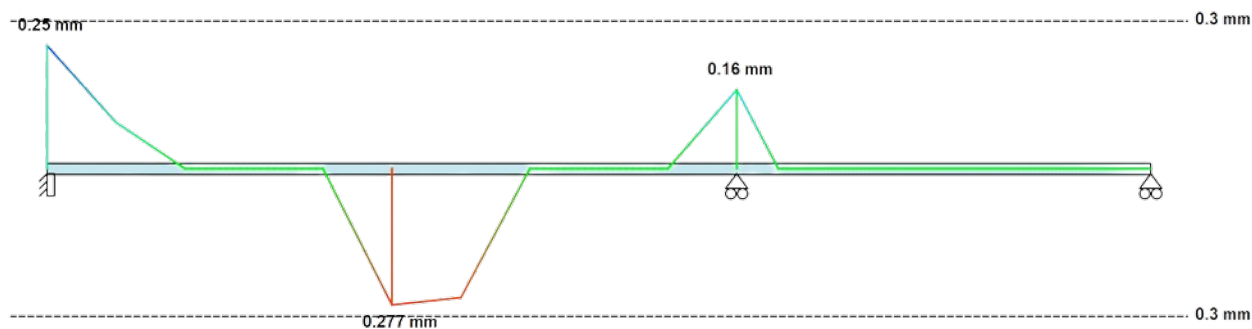
$$\sigma_{s2} \leq \sigma_{s,max} = 106 \text{ N/mm}^2 \leq 400 \text{ N/mm}^2 \Rightarrow \text{OK}$$

Figuur 6.13: Berekeningen van spanningen in BGT-Q

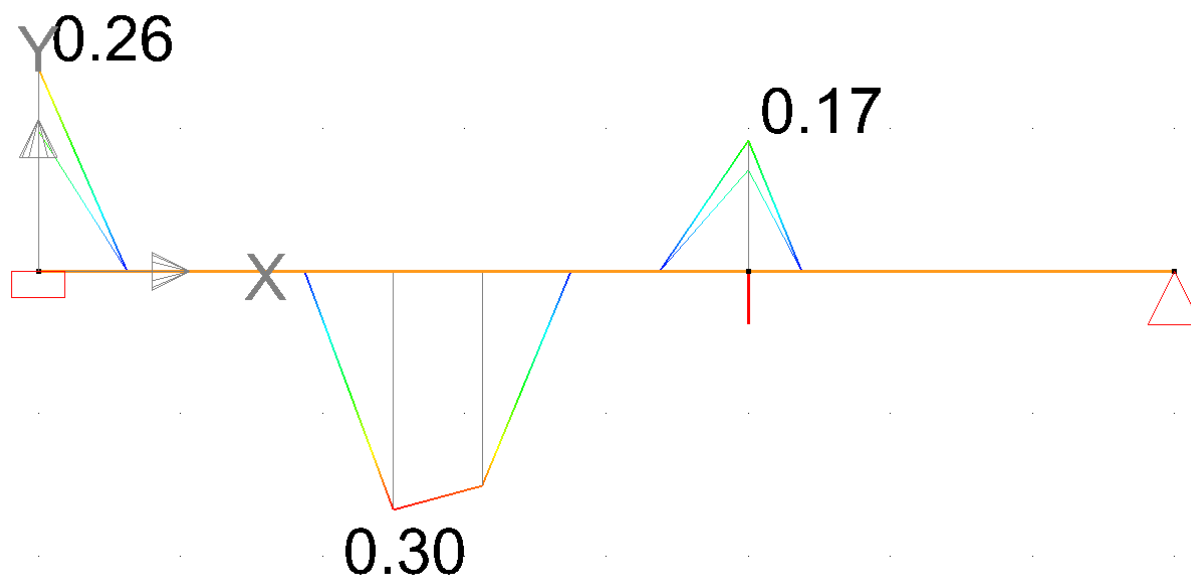
Uit Figuur 6.13 blijkt dat de maximale spanning in BGT-Q 7,7 N/mm² bedraagt. De spanning in de primaire wapening is 275 N/mm², terwijl de spanning in de secundaire wapening 106 N/mm² is. Al deze spanningen voldoen aan de maximaal toelaatbare spanningen voor BGT-Q.

6.2.4 Scheurwijdte

De scheurwijdte kan wel in beide programma's worden berekend. Figuur 6.14 toont de resultaten voor het ontwikkelde programma en Figuur 6.15 voor Diamonds.



Figuur 6.14: Resultaten scheurwijdte in ontwikkeld programma

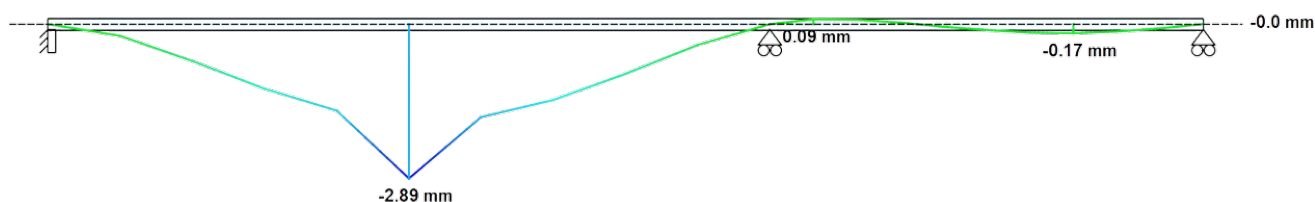


Figuur 6.15: Resultaten scheurwijdte in Diamonds

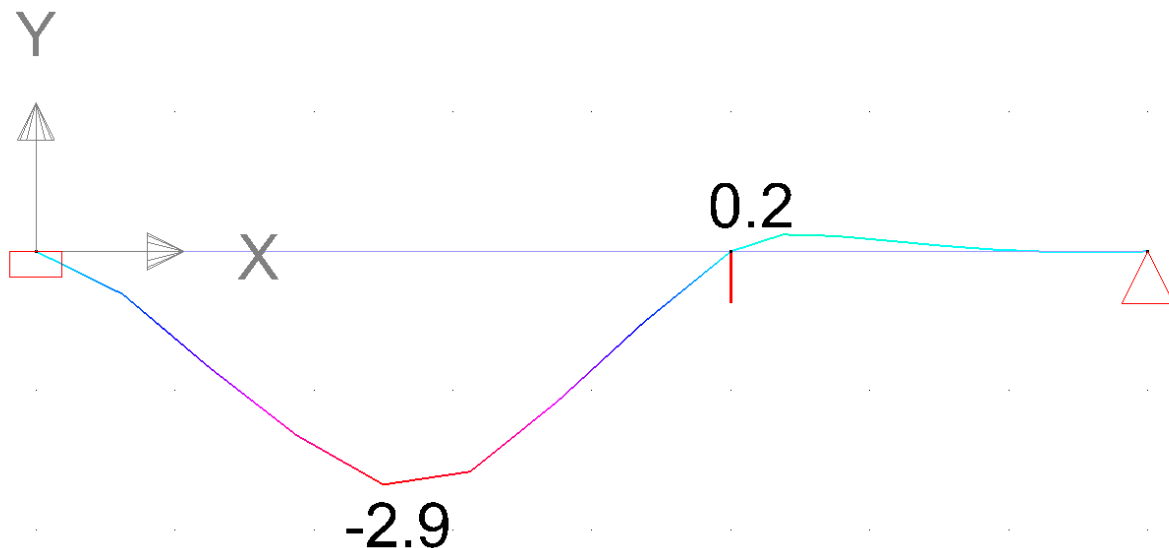
De resultaten voor de scheurwijdtes in Figuren 6.14 en 6.15 tonen een vergelijkbaar verloop. De waarden verschillen echter maximaal 8,5% van elkaar. Een mogelijke verklaring kan zijn dat er een andere kruipcoëfficiënt wordt gehanteerd, waardoor de elasticiteitsmodulus bij kruip in Diamonds verschilt van die in het ontwikkelde programma. Dit verschil beïnvloedt de effectieve hoogte en het traagheidsmoment van zowel de gescheurde als ongescheurde doorsnede. Wat vervolgens een invloed heeft op de spanningen en de scheurwijdte.

6.2.5 Vervorming met kruip

De vervormingen met kruip volgens het ontwikkelde programma worden getoond in Figuur 6.16 en voor Diamonds in Figuur 6.17



Figuur 6.16: Resultaten vervormingen met kruip in ontwikkeld programma

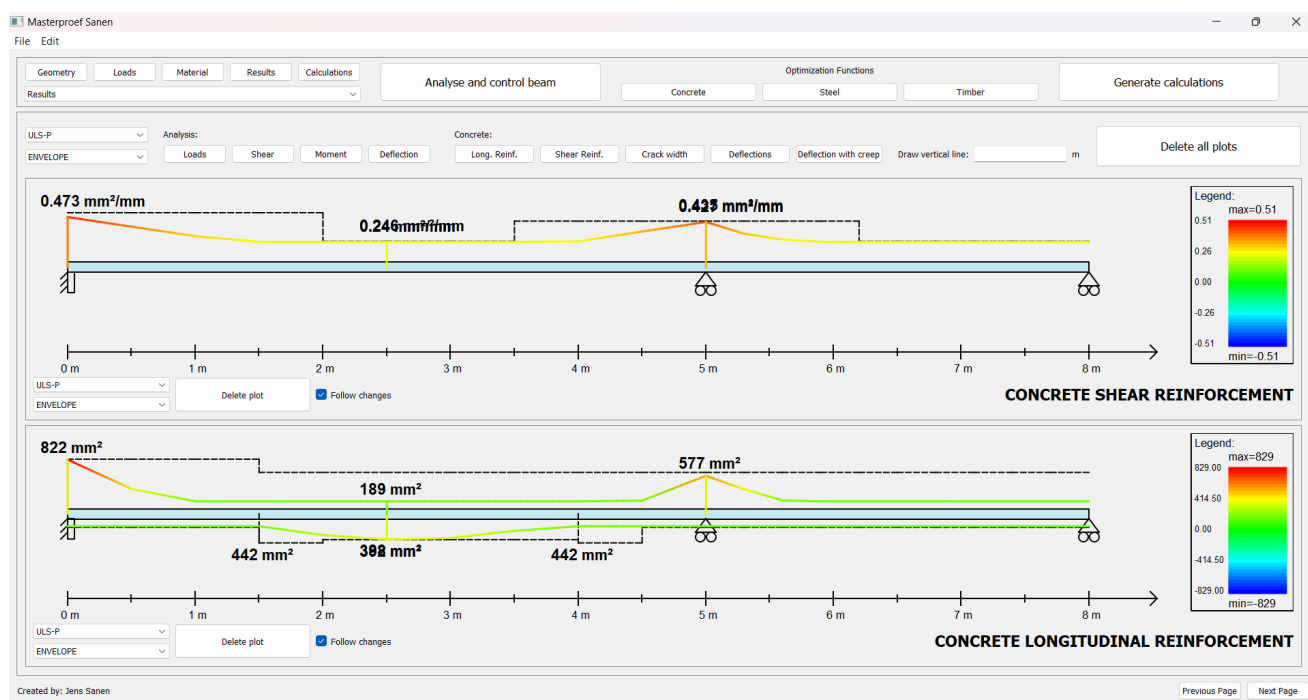


Figuur 6.17: Resultaten vervormingen met kruip in Diamonds

De vervorming in Figuur 6.16 en Figuur 6.17 vertonen vrijwel hetzelfde patroon. De piek van de vervorming in het midden van de eerste overspanning verschilt minder dan 0,5%, wat verwaarloosbaar is. Echter, de piekwaarde bij het middelste steunpunt in Diamonds is 120% groter dan die in het ontwikkelde programma. De reden voor dit verschil is echter niet duidelijk.

6.2.6 Optimalisatie

Het ontwikkelde programma biedt de mogelijkheid om de wapening te optimaliseren. Wanneer deze functie wordt toegepast op alle mogelijke criteria, resulteert dit in de wapeningsconfiguratie zoals weergegeven in Figuur 6.18. In deze figuur is bovenaan de controle op dwarskrachtwapening te zien, en onderaan de controle op langswapening.



Figuur 6.18: Resultaat controles wapening na optimalisatie

Bovenaan in Figuur 6.18 is duidelijk te zien dat er extra dwarskrachtwapening wordt geplaatst nabij de steunpunten. Dit is logisch, aangezien daar de meeste dwarskracht optreedt en er dus extra wapening nodig is. Onderaan in de figuur wordt de langswapening weergegeven. Hier is te zien dat de piekbelasting bij de inklemming wordt opgevangen door een extra wapeningsstaaf toe te voegen. Bovendien zijn er ook onderaan extra wapeningstaven aangebracht in het midden van de eerste overspanning.

6.2.7 Conclusie

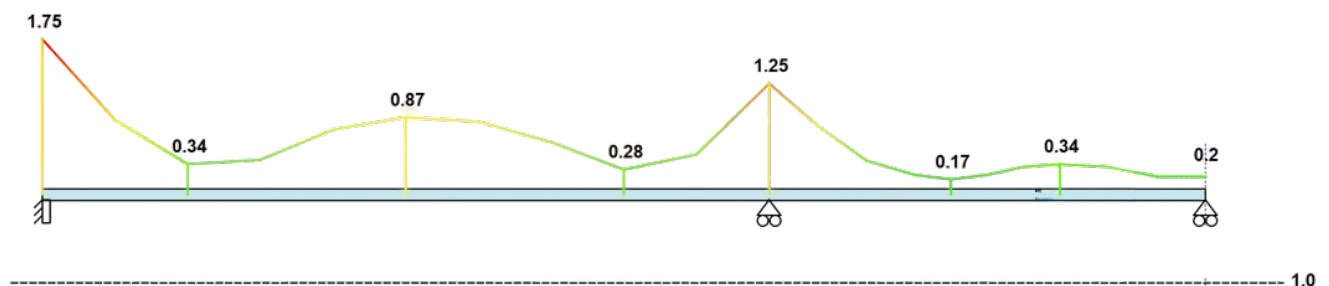
Alle resultaten voor beton in het ontwikkelde programma komen overeen met die van Diamonds. Hieruit kan worden geconcludeerd dat het ontwikkelde programma de controles correct uitvoert.

6.3 Staal

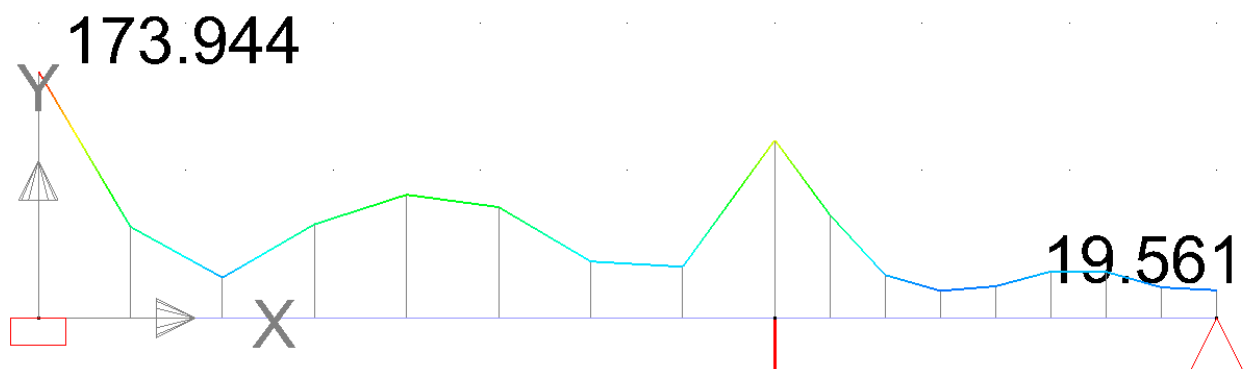
In deze sectie wordt een stalen balk gecontroleerd op sterkte, stabiliteit en vervorming. Hierbij wordt gebruikgemaakt van een IPE-200-profiel met staalkwaliteit S355.

6.3.1 Sterkte

De controle van de sterkte van de stalen balk in het ontwikkelde programma wordt getoond in Figuur 6.19 en voor Diamonds in Figuur 6.20.



Figuur 6.19: Resultaten sterkte in ontwikkeld programma



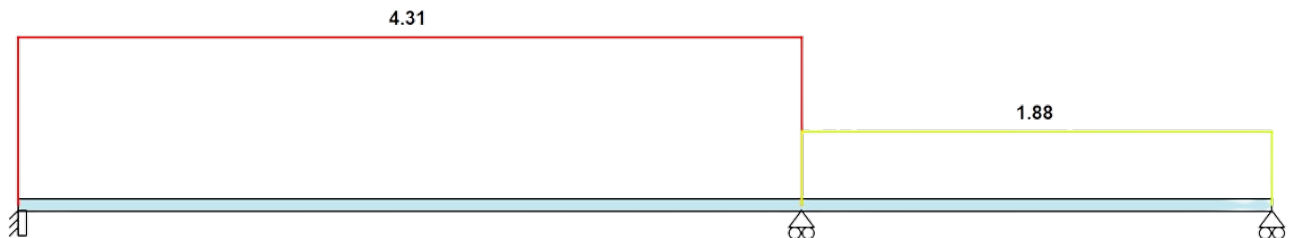
Figuur 6.20: Resultaten sterkte in Diamonds

De resultaten in Figuren 6.19 en 6.20 vertonen hetzelfde verloop. Hoewel de waarden in grootte verschillen, komt dit doordat het ontwikkelde programma op een schaal van 1 werkt, terwijl Diamonds op een schaal van 100 werkt. De afwijking tussen de resultaten bedraagt minder

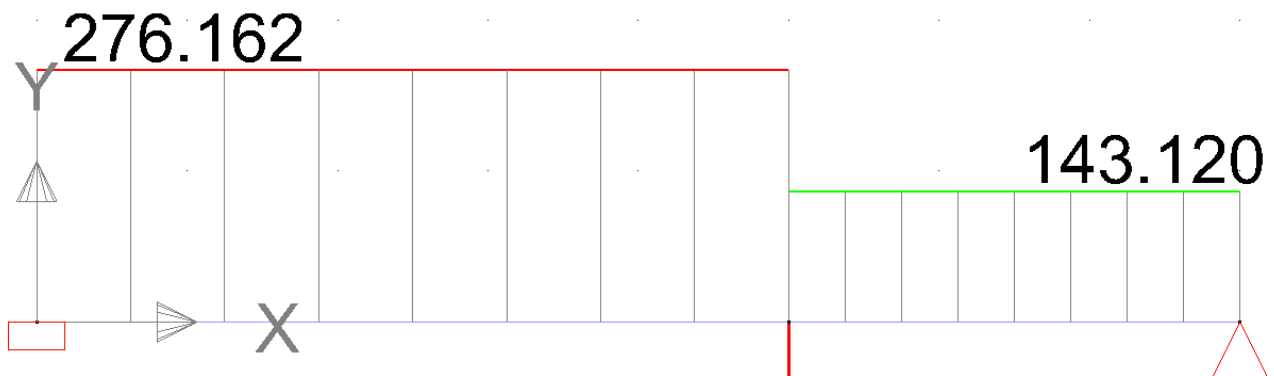
dan 2,5%, wat verwaarloosbaar is. Dit verschil wordt veroorzaakt door een klein verschil in het weerstandsbiedend moment tussen het ontwikkelde programma en Diamonds, evenals door afrondingen van de getallen in het ontwikkelde programma.

6.3.2 Stabiliteit

De resultaten van de stabiliteitscontrole met het ontwikkelde programma zijn te zien in Figuur 6.21, terwijl Figuur 6.22 de resultaten voor Diamonds weergeeft.



Figuur 6.21: Resultaten stabiliteit in ontwikkeld programma



Figuur 6.22: Resultaten stabiliteit in ontwikkeld Diamonds

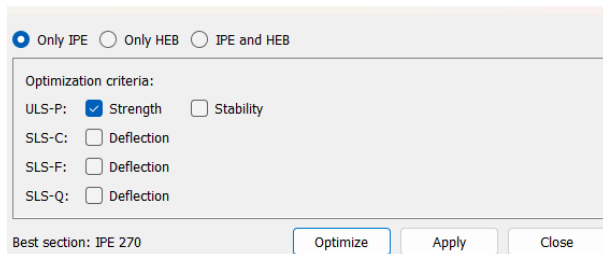
Figuur 6.21 laat zien dat de stabiliteitsresultaten van het ontwikkelde programma voor de balk respectievelijk 4,31 en 1,38 zijn voor de eerste en tweede overspanning, wat betekent dat de balk faalt. In vergelijking daarmee toont Figuur 6.22 resultaten van 2,76 en 1,43 voor Diamonds. Het verschil tussen de twee resultaten is te wijten aan het gebruik van vaste waarden voor de C_1 - en C_2 -factoren in het ontwikkelde programma, die alleen geschikt zijn voor opgelegde balken met een uniforme belasting. Deze waarden zijn niet toepasbaar op de huidige configuratie van de balk. Diamonds berekent de juiste factoren en voert daarom de stabiliteitscontrole correct uit. Het ontwikkelde programma kon deze factoren niet berekenen vanwege een gebrek aan formules (zie Sectie 3.3.4).

6.3.3 Vervorming

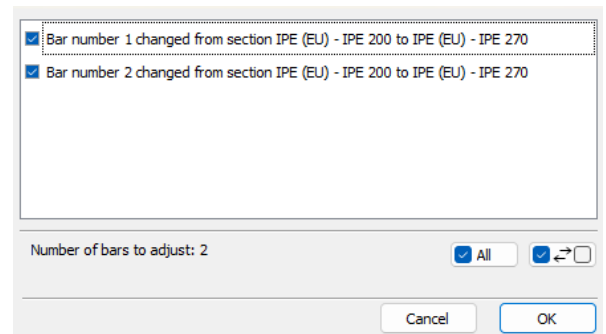
De vervormingen van staal worden gecontroleerd aan de hand van de resultaten uit de analyse. Omdat deze vervormingen in beide programma's hetzelfde zijn (zoals besproken in Sectie 6.1.3), worden ze hier niet verder besproken.

6.3.4 Optimalisatie

Beide programma's beschikken over een functie om de balk te optimaliseren. Deze functie is in beide programma's toegepast om de balken te optimaliseren op enkel sterkte omdat de stabiliteitscontrole niet overeenkomt tussen beide programma's. Het resultaat van het ontwikkelde programma is te zien in Figuur 6.23, en het resultaat in Diamonds wordt getoond in Figuur 6.24.



Figuur 6.23: Resultaat optimalisatie in ontwikkeld programma



Figuur 6.24: Resultaat optimalisatie in Diamonds

Zoals te zien is in Figuren 6.23 en 6.24 is te zien dat beide programma's het IPE-270 profiel hebben bepaald als het optimale profiel voor dit voorbeeld.

6.3.5 Conclusie

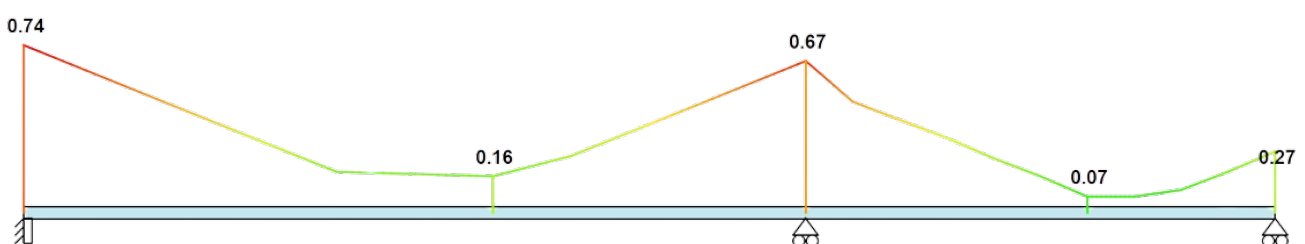
Het ontwikkelde programma blijkt geschikt voor de controles op sterkte en vervormingen voor staal. Om ook voor stabiliteitscontroles geschikt te zijn, moet de C1-factor echter nog correct worden berekend in het programma.

6.4 Hout

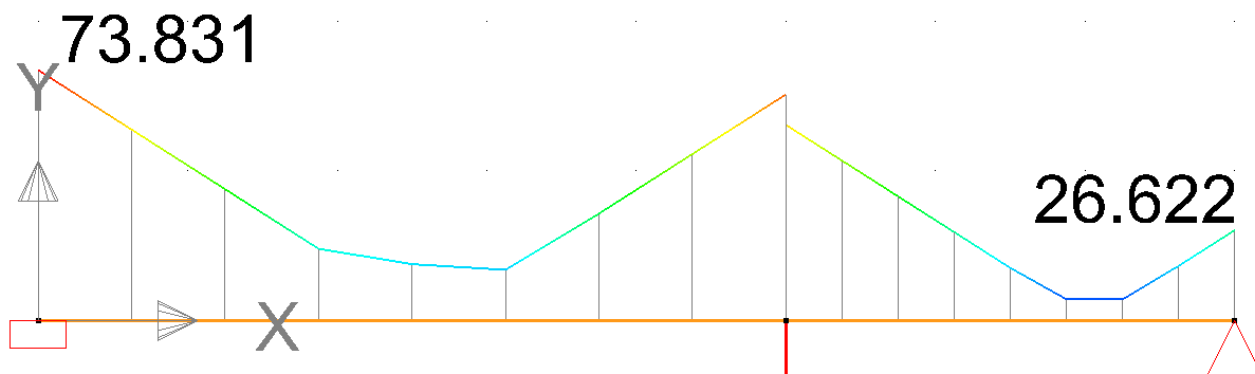
Bij houten balken worden sterkte, stabiliteit en vervormingen, zowel zonder als met kruip, gecontroleerd. In dit voorbeeld is een balk met een hoogte van 700 mm en een breedte van 350 mm gebruikt. De veranderlijke belastingsgevallen zijn beide langdurig, en de klimaatklasse is 1. Het gekozen materiaal is GL26h.

6.4.1 Sterkte

De resultaten van de sterktecontrole worden weergegeven in Figuur 6.25 voor het ontwikkelde programma en in Figuur 6.26 voor Diamonds.



Figuur 6.25: Resultaten sterkecontrole in ontwikkeld programma

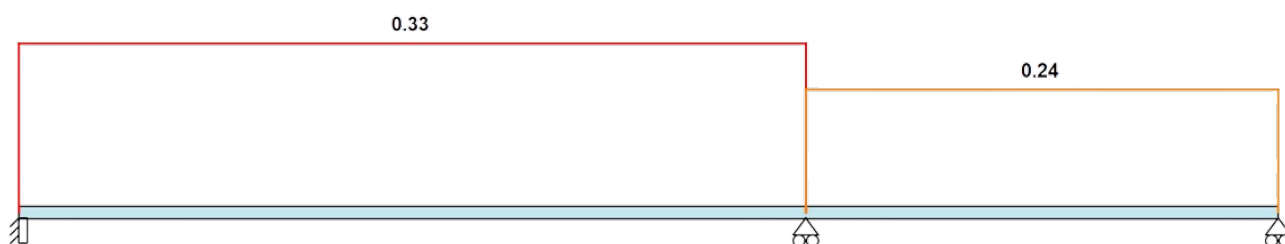


Figuur 6.26: Resultaten sterktecontrole in Diamonds

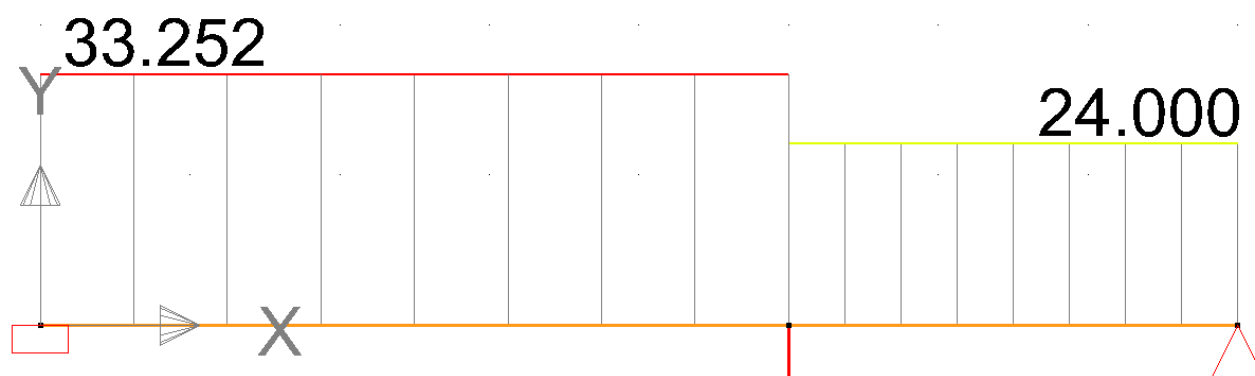
De resultaten in Figuur 6.25 komen overeen met die in Figuur 6.26. Het enige verschil ligt in de afrondingsmethode en de schaal waarop de resultaten worden weergegeven. In het ontwikkelde programma zijn de resultaten opnieuw geschaald op één, terwijl bij Diamonds de schaal op 100 staat.

6.4.2 Stabiliteit

De resultaten van de stabiliteitscontrole in het ontwikkelde programma worden getoond in Figuur 6.27, terwijl de resultaten voor de stabiliteitscontrole in Diamonds zijn weergegeven in Figuur 6.28.



Figuur 6.27: Resultaten sterktecontrole in ontwikkeld programma

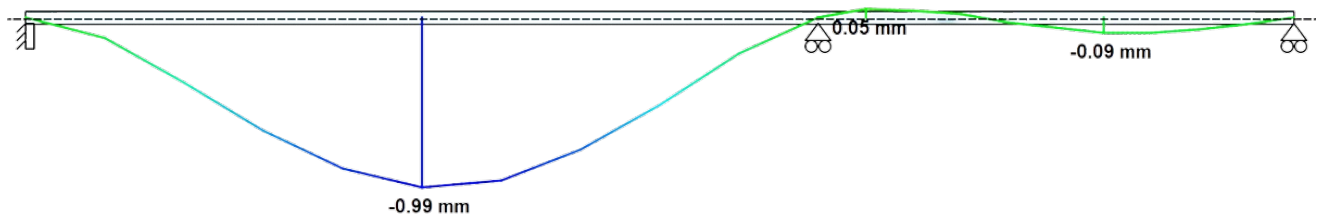


Figuur 6.28: Resultaten stabiliteitcontrole in Diamonds

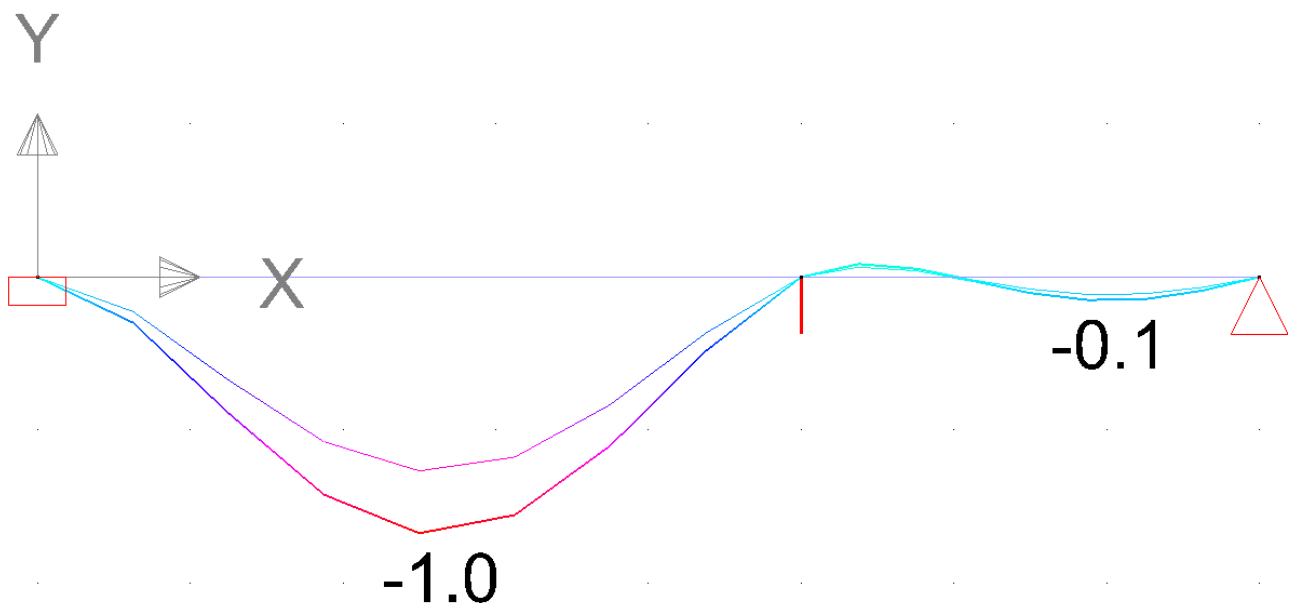
De resultaten in Figuren 6.27 en 6.28 zijn identiek. Het enige verschil is opnieuw de manier van afronden en de schaal.

6.4.3 Vervormingen

De vervormingen zonder kruip worden, net als bij staal, uit de analyse verkregen en worden daarom niet verder besproken. De berekening van de vervormingen met kruip gebeurt in BGT-Q en is weergegeven in Figuur 6.29 voor het ontwikkelde programma en in Figuur 6.30 voor Diamonds.



Figuur 6.29: Resultaten vervorming met kruip in ontwikkeld programma



Figuur 6.30: Resultaten vervorming met kruip in Diamonds

De resultaten in Figuur 6.29 en Figuur 6.30 tonen een vergelijkbare trend. Het verschil van 1% is te verwaarlozen.

6.4.4 Optimalisatie

Beide programma's bieden de mogelijkheid om de doorsnede te optimaliseren op basis van sterkte en stabiliteit. In Diamonds is het echter alleen mogelijk om of enkel de hoogte of enkel de breedte te optimaliseren, terwijl in het ontwikkelde programma enkel beide tegelijkertijd geoptimaliseerd worden. Hierdoor is een vergelijking van deze functies niet mogelijk. Wanneer de functie in het ontwikkelde programma wordt gebruikt, resulteert dit in een hoogte van 620 mm en een breedte van 310 mm. De resultaten van de sterkte- en stabiliteitscontrole voor deze geoptimaliseerde doorsnede zijn weergegeven in Figuur 6.23.



Figuur 6.31: Resultaten geoptimaliseerde doorsnede

Figuur 6.31 toont de stabiliteitscontrole bovenaan en de sterktecontrole onderaan. De waarden van deze controles zijn hoger dan die van de originele doorsnede, maar blijven onder de grens van 1. Dit bevestigt dat de controles geslaagd zijn en de optimalisatie correct is uitgevoerd.

6.4.5 Conclusie

De resultaten van elke controle zijn identiek aan die van de controles in Diamonds, wat bevestigt dat het ontwikkelde programma houten balken correct controleert.

Hoofdstuk 7

Conclusie

In deze masterproef is een softwareprogramma ontwikkeld voor de analyse en controle van hyperstatische balken in beton, staal en hout. Het programma maakt gebruik van de relevante Eurocodes voor elk materiaal, evenals Eurocode 0 voor de bepaling van belastingcombinaties. Het programma voert controles uit en biedt optimalisatiemogelijkheden op basis van deze controles. Alle berekeningen en controles zijn geverifieerd door vergelijking met een bestaand softwareprogramma, waarbij alle resultaten overeenkomen met het bestaande programma, behalve de stabiliteitscontrole van stalen balken. De reden hiervoor is het ontbreken van de juiste formules voor de bepaling van de C_1 -factor en C_2 -factor voor het kritisch buigmoment.

De gebruikersinterface is ontworpen met een logische structuur, waardoor gebruikers eenvoudig en snel een balk kunnen analyseren en controleren. De interface biedt toegang tot de resultaten en de bijbehorende formules met rekenwaarden, en is gebruiksvriendelijk met voldoende visuele ondersteuning. Het programma is gericht op het dienen als een effectief hulpmiddel voor het verkrijgen van een beter inzicht van de leerstof over balken.

Hoewel de berekeningen en gebruikersinterface operationeel zijn, moeten er nog functies worden toegevoegd voor het opslaan, openen en exporteren van bestanden om het gebruiksgemak van het programma te verhogen. Het programma is nog niet getest door andere gebruikers, dus het is belangrijk om toekomstige feedback van gebruikers te verzamelen en eventuele problemen op te lossen.

Verder zijn er tal van uitbreidingsmogelijkheden voor het programma. Zo kan de mogelijkheid worden toegevoegd om balken met normaalkrachten te analyseren en te controleren. Ook kunnen de soorten doorsneden voor elk materiaal worden uitgebreid, inclusief de bijbehorende controles. Toekomstige versies kunnen functies voor accidentele belastingen en tweede-orde effecten integreren. De gebruikersinterface kan verder worden geoptimaliseerd door het toevoegen van sneltoetsen en het verbeteren van de esthetische waarde.

Literatuurlijst

- [1] NBN, *NBN EN 1992-1-1: Eurocode 2: Ontwerp en berekeningen van betonconstructies - Deel 1-1: Algemene regels en regels voor gebouwen (+AC:2010)*. Bureau voor Normalisatie, 2005.
- [2] NBN, *NBN EN 1993-1-1: Eurocode 3: Ontwerp en berekeningen voor staalconstructies - Deel 1-1: Algemene regels en regels voor gebouwen (+AC:2009)*. Bureau voor Normalisatie, 2005.
- [3] NBN, *NBN EN 1995-1-1: Eurocode 5: Ontwerp en berekeningen voor houtconstructies - Deel 1-1: Algemeen - Gemeenschappelijke regels en regels voor gebouwen (+AC:2006)*. Bureau voor Normalisatie, 2005.
- [4] “Scia engineer.” [Online]. Available: <https://www.scia.net/nl/scia-engineer> [Geopend: 1 oktober 2023].
- [5] “Diamonds-2d-starter.” [Online]. Available: <https://www.buildsoft.eu/en/webshop/diamonds-2d-starter> [Geopend: 1 oktober 2023].
- [6] “Axisvm.” [Online]. Available: <https://axisvm.eu/axisvm-eng/> [Geopend: 2 februari 2024].
- [7] P. Gielkens, J. Jans, and J. Sanen, *Towards new visualisation tools for a deeper understanding of the design of reinforced concrete beams*. [eindwerk], Diepenbeek: Gezamenlijke opleiding Industriële ingenieurswetenschappen UHasselt en KU Leuven, 2023.
- [8] NBN, *NBN EN 1990: Eurocode - Grondslagen van het constructief ontwerp*. Bureau voor Normalisatie, 2002.
- [9] “Python.” [Online]. Available: <https://www.python.org/> [Geopend: 1 oktober 2023].
- [10] “Wat is python?.” Uitleg over de programmeertaal,” data science partners, 19 augustus 2019. [Online]. Available: <https://datasciencepartners.nl/wat-is-python/>. [Geopend 1 oktober 2023].
- [11] “Db browser for sqlite.” [Online]. Available: <https://sqlitebrowser.org> [Geopend: 4 juli 2024].
- [12] “Werking van fem sterkteberekening.” [Online]. Available: <https://sterkteberekening.nl/werking-sterkteberekening/> [Geopend: 23 februari 2024].
- [13] “Qt creator.” [Online]. Available: <https://www.qt.io/product/development-tools> [Geopend: 1 oktober 2023].

- [14] “Concrete (plus).” [Online]. Available: <https://www.buildsoft.eu/nl/product/concrete-plus> [Geopend: 2 februari 2024”].
- [15] “Tekla tedds.” [Online]. Available: <https://www.tekla.com/products/tekla-tedds> [Geopend: 2 februari 2024”].
- [16] “Frilo continuous beam.” [Online]. Available: <https://www.frilo.eu/en/product/continuous-beam/> [Geopend: 2 februari 2024”].
- [17] “Strusoft structural analysis software.” [Online]. Available: <https://strusoft.com/software/3d-structural-analysis-software-fem-design/> [Geopend: 2 februari 2024”].
- [18] “Technosoft liggers.” [Online]. Available: <https://www.technosoft.nl/liggers> [Geopend: 2 februari 2024”].
- [19] R. Bloemen and J. Jehaes, *Ontwikkeling en implementatie van een iOS-app voor de berekening en optimalisatie van hyperstatistische balken*. [eindwerk], Diepenbeek: Gezamenlijke opleiding Industriële ingenieurswetenschappen UHasselt en KU Leuven, 2017.
- [20] “About buildsoft.” [Online]. Available: <https://www.buildsoft.eu/en/about-buildsoft> [Geopend: 23 februari 2024”].
- [21] “Buildsoft webshop.” [Online]. Available: <https://www.buildsoft.eu/nl/webshop/diamonds-2d-starter> [Geopend: 23 februari 2024”].
- [22] “Over scia.” [Online]. Available: <https://www.scia.net/nl/over-ons/over-scia> [Geopend: 24 februari 2024”].
- [23] “Scia software shop.” [Online]. Available: <https://www.scia.net/nl/webshop> [Geopend: 24 februari 2024”].
- [24] “Tekla tedds engineering library.” [Online]. Available: <https://www.tekla.com/products/tekla-tedds/calculation-list> [Geopend: 25 februari 2024”].
- [25] “About us.” [Online]. Available: <https://axisvm.eu/about-us/> [Geopend: 25 februari 2024”].
- [26] L. Pyl, *Metaalbouw I*. [cursus], Sint-Katelijne-Waver: Campus DE NAYER, 2011.
- [27] UHasselt. (2023). Belastingen en materiaal: les 01. [PowerPoint]. Niet-gepubliceerd.
- [28] NBN, *NBN EN 1990 ANB: Eurocode - Grondslagen van het constructief ontwerp - Nationale bijlage*. Bureau voor Normalisatie, 2013.
- [29] UHasselt, (2012). Les 3: Eindige-elementenmethode (deel 2). [PowerPoint]. Niet-gepubliceerd.
- [30] “Beton.” [Online]. Available: <https://nl.wikipedia.org/wiki/Beton> [Geopend: 28 mei 2024”].
- [31] UHasselt. (2023). Balken UGT: les 02. [PowerPoint]. Niet-gepubliceerd.
- [32] NBN, *NBN EN 1992-1-1 ANB: Eurocode 2: Ontwerp en berekeningen van betonconstructies - Deel 1-1: Algemene regels en regels voor gebouwen (+AC:2010)*. Bureau voor Normalisatie, 2010.

- [33] NBN, *NBN B 15-001:2024: Beton - Specificatie, eigenschappen, vervaardiging en conformiteit - Nationale aanvulling bij NBN EN 206:2013+A2:2021*. Bureau voor Normalisatie, 2014.
- [34] UHasselt. (2023). Balken GGT: les 03. [PowerPoint]. Niet-gepubliceerd.
- [35] UHasselt. (2023). Constructieve schikkingen: les 04. [PowerPoint]. Niet-gepubliceerd.
- [36] UHasselt. (2023). Dwarskracht: les 5-a. [PowerPoint]. Niet-gepubliceerd.
- [37] NBN, *NBN EN 1993-1-1 ANB: Eurocode 3: Ontwerp en berekeningen voor staalconstructies - Deel 1-1: Algemene regels en regels voor gebouwen - Nationale bijlage*. Bureau voor Normalisatie, 2010.
- [38] “Product catalogues - sections.” [Online]. Available: https://sections.arcelormittal.com/products_and_solutions/Product_catalogues/EN [Geopend: 3 juli 2024].
- [39] “Kippen (techniek).” [Online]. Available: [https://nl.wikipedia.org/wiki/Kippen_\(techniek\)](https://nl.wikipedia.org/wiki/Kippen_(techniek)) [Geopend: 3 juli 2024].
- [40] NBN, *Structural timber - Strength classes*. Bureau voor Normalisatie, 2016.
- [41] NBN, *Houtconstructies - Gelijmd gelamineerd hout en gelijmd massief hout - Eisen*. Bureau voor Normalisatie, 2013.
- [42] NBN, *NBN EN 1995-1-1 ANB: Eurocode 5: Ontwerp en berekeningen voor houtconstructies - Deel 1-1: Algemeen - Gemeenschappelijke regels en regels voor gebouwen - Nationale bijlage*. Bureau voor Normalisatie, 2012.
- [43] UHasselt. (2023). HC-HS3: Ontwerp van houtelementen (zonder instabiliteit). [PowerPoint]. Niet-gepubliceerd.
- [44] NBN, *prEN 1995-1-1:2023 Eurocode 5: Ontwerp en berekeningen voor houtconstructies - Deel 1-1: Algemeen - Gemeenschappelijke regels en regels voor gebouwen*. Bureau voor Normalisatie, 2023.
- [45] “Sqlite.” [Online]. <https://nl.wikipedia.org/wiki/SQLite> [Geopend: 4 juli 2024].
- [46] “Sql.” [Online]. <https://nl.wikipedia.org/wiki/SQL> [Geopend: 4 juli 2024].
- [47] “Pyqt5.” [Online]. <https://pypi.org/project/PyQt5> [Geopend: 10 augustus 2024].
- [48] “Latex – a document preparation system.” [Online]. <https://www.latex-project.org/> [Geopend: 14 augustus 2024].
- [49] “Dc ec 02: longitudinal reinforcement in beam under pure bending (uls design).” [Online]. <https://support.buildsoft.eu/knowledge-base/dc-ec-02-longitudinal-reinforcement-in-beam-under-pure-bending/> [Geopend: 14 augustus 2024].
- [50] “Dc ec 01: shear reinforcement in beam.” [Online]. <https://support.buildsoft.eu/knowledge-base/dc-ec-01-shear-reinforcement-in-beam/> [Geopend: 14 augustus 2024].

Bijlagen

Bijlage A - Rapport SCIA Engineer	195
Bijlage B - Rapport Tekla Tedds	199
Bijlage C - Eurocode 2 regels die niet van toepassing zijn	211
Bijlage D - Tabel B4.2.3.7-a	213
Bijlage E - Eurocode 3 regels die niet van toepassing zijn	215
Bijlage F - Eurocode 5 regels die niet van toepassing zijn	217
Bijlage G - Formulier berekeningen beton	219
Bijlage H - Formulier berekeningen staal	223
Bijlage I - Formulier berekeningen hout	225

Bijlage A - Rapport SCIA Engineer

EC-EN 1993 UGT: staalcontrole

Lineaire berekening

Combinatie: UGT-Set B (automatisch)

Assenstelsel: Hoofd

Extreme 1D: Globaal

Selectie: Alle

EN 1993-1-1 Normcontrole

Nationale bijlage: Belgische NBN-EN NA

Element S1	3.000 / 6.000 m	IPE240	Gewalst	S 235	UGT-Set B (automatisch)	4.11 -
------------	-----------------	--------	---------	-------	-------------------------	--------

Combinatiesleutel

UGT-Set B (automatisch) / 1.35*BG1 + 1.35*BG2

Partiële veiligheidsfactoren

Weerstand van doorsneden	γ_{M0}	1.00
Weerstand tegen instabiliteit	γ_{M1}	1.00
Weerstand van nettodoorsneden	γ_{M2}	1.25

Materiaal

Vloeisterkte	f_y	235.0	MPa
Treksterkte	f_u	360.0	MPa

.....DOORSNEDECONTROLE:.....

De kritische controle is op positie 3.000 m

Interne krachten		Berekende	Eenheid
Normaalkracht	N_{Ed}	0.00	kN
Dwarskracht	$V_{y,Ed}$	0.00	kN
Dwarskracht	$V_{z,Ed}$	0.00	kN
Torsie	T_{Ed}	0.00	kNm
Buigend moment	$M_{y,Ed}$	153.70	kNm
Buigend moment	$M_{z,Ed}$	0.00	kNm

Classificatie voor doorsnede-ontwerp

Classificatie volgens EN 1993-1-1 Artikel 5.5.2

Classificatie van interne en uitkragende onderdelen volgens EN 1993-1-1 tabel 5.2 blad 1 en 2

Id	Type	c [mm]	t [mm]	σ_1 [kN/m ²]	σ_2 [kN/m ²]	Ψ [-]	k_σ [-]	α [-]	c/t [-]	Limiet klasse 1 [-]	Limiet klasse 2 [-]	Limiet klasse 3 [-]	Klasse
1	SO	42	10	-454485.405	-454485.405								
3	SO	42	10	-454485.405	-454485.405								
4	I	190	6	-375907.998	375907.998	-1.0		0.5	30.7	72.0	83.0	124.0	1
5	SO	42	10	454485.405	454485.405	1.0	0.4	1.0	4.3	9.0	10.0	14.0	1
7	SO	42	10	454485.405	454485.405	1.0	0.4	1.0	4.3	9.0	10.0	14.0	1

Opmerking: De classificatielimieten zijn ingesteld volgens Semi-Comp+.

De doorsnede is geclassificeerd als klasse 1

Controle buigend moment voor M_y

Volgens EN 1993-1-1 artikel 6.2.5 en formule (6.12),(6.13)

Plastische doorsnedemodulus	$W_{pl,y}$	3.6700e-04	m ³
Plastisch buigend moment	$M_{pl,y,Rd}$	86.25	kNm
Eenhedscontrole		1.78	-

Dwarskrachtcontrole voor V_z

Volgens EN 1993-1-1 artikel 6.2.6 en formule (6.17)

Correctiefactor voor dwarskracht	η	1.20	
Afschuifoppervlak	A_v	1.9128e-03	m ²
Plastische dwarskrachtweerstand voor V_z	$V_{pl,z,Rd}$	259.52	kN
Eenhedscontrole		0.00	-

De staaf voldoet NIET aan de doorsnedecontrole!

.....STABILITEITSCONTROLE:.....

Classificatie voor staafknikontwerp

Beslissende positie voor stabiliteitsclassificatie: 3.000 m

Beslissende utilisatiefactor η : 1.78

Classificatie volgens EN 1993-1-1 Artikel 5.5.2

Classificatie van interne en uitkragende onderdelen volgens EN 1993-1-1 tabel 5.2 blad 1 en 2

Id	Type	c [mm]	t [mm]	σ_1 [kN/m ²]	σ_2 [kN/m ²]	Ψ [-]	k_σ [-]	α [-]	c/t [-]	Limiet klasse 1 [-]	Limiet klasse 2 [-]	Limiet klasse 3 [-]	Klasse
1	SO	42	10	-454485.405	-454485.405								
3	SO	42	10	-454485.405	-454485.405								
4	I	190	6	-375907.998	375907.998	-1.0		0.5	30.7	72.0	83.0	124.0	1
5	SO	42	10	454485.405	454485.405	1.0	0.4	1.0	4.3	9.0	10.0	14.0	1
7	SO	42	10	454485.405	454485.405	1.0	0.4	1.0	4.3	9.0	10.0	14.0	1

Opmerking: De classificatielimieten zijn ingesteld volgens Semi-Comp+.

De doorsnede is geclassificeerd als klasse 1

Opmerking: De beslissende positie voor de stabiliteitsclassificatie is gebaseerd op de utilisatiefactor η volgens Semi-Comp+.

Kipcontrole

Volgens EN 1993-1-1 artikel 6.3.2.1 & 6.3.2.3 en formule (6.54)

Kip parameters			
Methode voor Kipcurve		Alternatief geval	
Plastische doorsnedemodulus	$W_{pl,y}$	3.6700e-04	m ³
Elastisch kritisch moment	M_{cr}	51.10	kNm
Relatieve slankheid	$\lambda_{rel,LT}$	1.30	
Limietslankheid	$\lambda_{rel,LT,0}$	0.20	
Kipcurve		b	
Imperfectie	α_{LT}	0.34	
Kipfactor	β	1.00	
Reductie factor	χ_{LT}	0.43	
Correctiefactor	k_c	0.94	
Correctiefactor	f	0.99	
Gewijzigde reductiefactor	$\chi_{LT,mod}$	0.43	
Rekenwaarde knikweerstand	$M_{b,Rd}$	37.40	kNm
Eenhedscontrole		4.11	-

M_{cr} parameters			
LTB lengte	l_{LT}	6.000	m
Invloed van lastpositie		geen invloed	
Correctiefactor	k	1.00	
Correctiefactor	k_w	1.00	
Kip moment factor	C_1	1.13	
Kip moment factor	C_2	0.45	
Kip moment factor	C_3	0.53	
Afschuifmiddelpuntafstand	d_z	0	mm
Afstand tot lastoepassing	z_g	0	mm
Mono-symmetrische constante	β_y	0	mm
Mono-symmetrische constante	z_1	0	mm

Opmerking: C parameters zijn bepaald volgens de ECCS 119 2006 / Galea 2002.

Opmerking: De correctiefactor k_c wordt bepaald op basis van C_1 .

Plooicontrole

Volgens EN 1993-1-5 artikel 5 & 7.1 en formule (5.10) & (7.1)

Plooi parameters			
Knik veldlengte	a	6.000	m
Lijf		niet-verstijfd	
Lijfhoogte	h_w	220	mm
Lijfdikte	t	6	mm
Materiaal coëfficiënt	ϵ	1.00	
Correctiefactor voor dwarskracht	η	1.20	

Plooi verificatie		
Lijf slankheid	h_w/t	35.55
Lijfslankheid limiet		60.00

Opmerking: De slankheid van het lijf is zo dat de Plooi effecten kunnen worden genegeerd volgens EN 1993-1-5 artikel 5.1(2).

De staaf voldoet NIET aan de stabiliteitscontrole!

Studentenversie

Bijlage B - Rapport Tekla Tedds

Project				Job no.	
Calcs for				Start page no./Revision 1	
Calcs by J	Calcs date 29/05/2024	Checked by	Checked date	Approved by	Approved date

RC MEMBER ANALYSIS & DESIGN (EN1992-1-1:2004)

In accordance with EN1992-1-1:2004 incorporating Corrigenda January 2008 and the recommended values

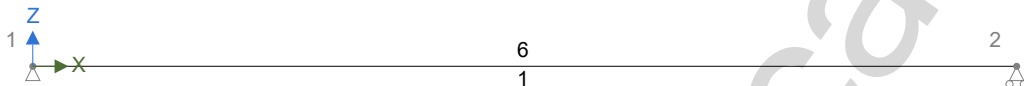
Tedds calculation version 3.3.10

ANALYSIS

Tedds calculation version 1.0.37

Geometry

Geometry (m) - Concrete (C25 2500 Quartzite) - R 300x500



Span	Length (m)	Section	Start Support	End Support
1	6	R 300x500	Pinned	Roller Pin X
R 300x500: Area 1500 cm ² , Inertia Major 312500 cm ⁴ , Inertia Minor 112500 cm ⁴ , Shear area parallel to Minor 1250 cm ² , Shear area parallel to Major = 1250 cm ²				
Concrete (C25 2500 Quartzite): Density 2500 kg/m ³ , Youngs 31.4758062 kN/mm ² , Shear 13.1149193 kN/mm ² , Thermal 0.00001 °C ⁻¹				

Loading

Self weight included

Permanent - Loading (kN/m)



Load combination factors

Load combination	Self Weight	Permanent	Imposed
1.35G + 1.5Q + 1.5RQ (Strength)	1.35	1.35	1.50
1.0G + 1.0Q + 1.0RQ (Service)	1.00	1.00	1.00
1.0G + 1.0ψ ₂ Q (Quasi)	1.00	1.00	0.30

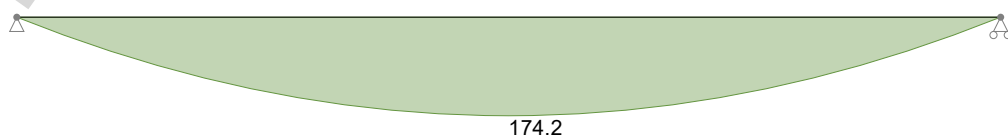
Member Loads

Member	Load case	Load Type	Orientation	Description
Beam	Permanent	UDL	GlobalZ	25 kN/m

Results

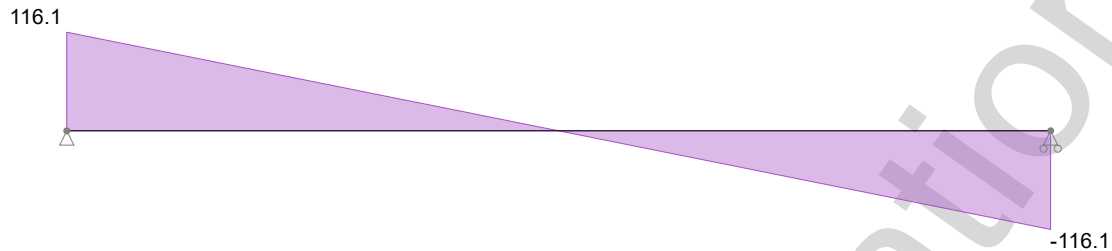
Forces

Strength combinations - Moment envelope (kNm)



Project				Job no.	
Calcs for				Start page no./Revision	
				2	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

Strength combinations - Shear envelope (kN)



Concrete details - Table 3.1. Strength and deformation characteristics for concrete

Concrete strength class	C25/30
Aggregate type	Quartzite
Aggregate adjustment factor - cl.3.1.3(2)	AAF = 1.0
Characteristic compressive cylinder strength	$f_{ck} = 25 \text{ N/mm}^2$
Mean value of compressive cylinder strength	$f_{cm} = f_{ck} + 8 \text{ N/mm}^2 = 33 \text{ N/mm}^2$
Mean value of axial tensile strength	$f_{ctm} = 0.3 \text{ N/mm}^2 \times (f_{ck} / 1 \text{ N/mm}^2)^{2/3} = 2.6 \text{ N/mm}^2$
Secant modulus of elasticity of concrete	$E_{cm} = 22 \text{ kN/mm}^2 \times (f_{cm} / 10 \text{ N/mm}^2)^{0.3} \times \text{AAF} = 31476 \text{ N/mm}^2$
Ultimate strain - Table 3.1	$\epsilon_{cu2} = 0.0035$
Shortening strain - Table 3.1	$\epsilon_{cu3} = 0.0035$
Effective compression zone height factor	$\lambda = 0.80$
Effective strength factor	$\eta = 1.00$
Coefficient k_1	$k_1 = 0.44$
Coefficient k_2	$k_2 = 1.25 \times (0.6 + 0.0014 / \epsilon_{cu2}) = 1.25$
Coefficient k_3	$k_3 = 0.54$
Coefficient k_4	$k_4 = 1.25 \times (0.6 + 0.0014 / \epsilon_{cu2}) = 1.25$
Partial factor for concrete - Table 2.1N	$\gamma_C = 1.50$
Compressive strength coefficient - cl.3.1.6(1)	$\alpha_{cc} = 1.00$
Design compressive concrete strength - exp.3.15	$f_{cd} = \alpha_{cc} \times f_{ck} / \gamma_C = 16.7 \text{ N/mm}^2$
Compressive strength coefficient - cl.3.1.6(1)	$\alpha_{ccw} = 1.00$
Design compressive concrete strength - exp.3.15	$f_{cwd} = \alpha_{ccw} \times f_{ck} / \gamma_C = 16.7 \text{ N/mm}^2$
Maximum aggregate size	$h_{agg} = 20 \text{ mm}$
Density of reinforced concrete	$\rho = 2500 \text{ kg/m}^3$
Monolithic simple support moment factor	$\beta_1 = 0.15$

Reinforcement details

Characteristic yield strength of reinforcement	$f_{yk} = 500 \text{ N/mm}^2$
Partial factor for reinforcing steel - Table 2.1N	$\gamma_S = 1.15$
Design yield strength of reinforcement	$f_{yd} = f_{yk} / \gamma_S = 435 \text{ N/mm}^2$

Nominal cover to reinforcement

Nominal cover to top reinforcement	$c_{nom_t} = 35 \text{ mm}$
Nominal cover to bottom reinforcement	$c_{nom_b} = 35 \text{ mm}$
Nominal cover to side reinforcement	$c_{nom_s} = 35 \text{ mm}$

Fire resistance

Standard fire resistance period	$R = 60 \text{ min}$
Number of sides exposed to fire	3

Project				Job no.	
Calcs for				Start page no./Revision	
				3	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

Minimum width of beam - EN1992-1-2 Table 5.5 $b_{min} = 120 \text{ mm}$

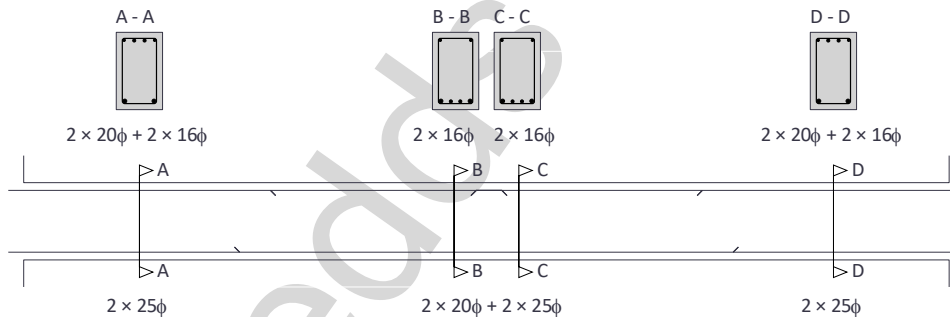
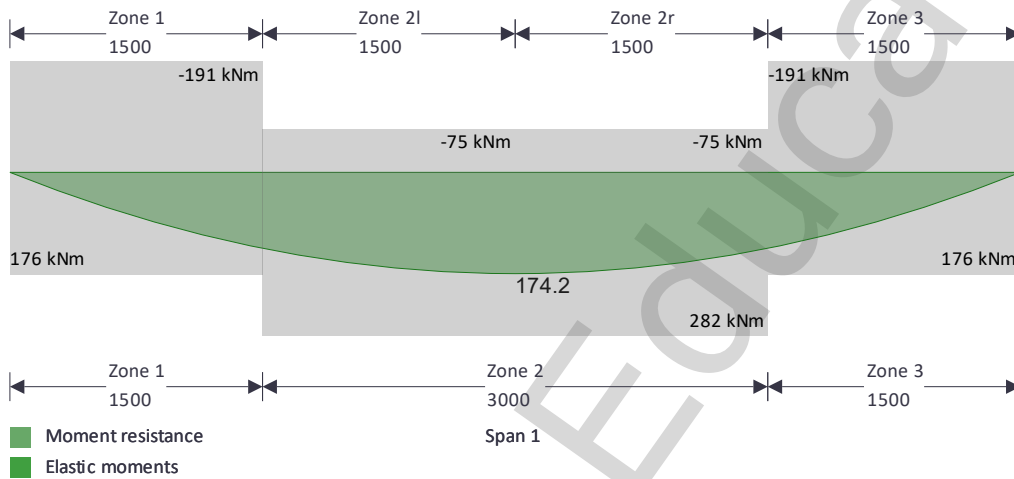
Beam - Span 1

Rectangular section details

Section width $b = 300 \text{ mm}$
Section depth $h = 500 \text{ mm}$

PASS - Minimum dimensions for fire resistance met

Moment design



Zone 1 (0 mm - 1500 mm) Positive moment - section 6.1

Design bending moment	$M = \text{abs}(M_{m1_s1_z1_max_red}) = 130.7 \text{ kNm}$
Effective depth of tension reinforcement	$d = 445 \text{ mm}$
Redistribution ratio	$\delta = \min(M_{pos_red_z1} / M_{pos_z1}, 1) = 1.000$
	$K = M / (b \times d^2 \times f_{ck}) = 0.088$
	$K' = (2 \times \eta \times \alpha_{cc} / \gamma_c) \times (1 - \lambda \times (\delta - k_1) / (2 \times k_2)) \times (\lambda \times (\delta - k_1) / (2 \times k_2)) = 0.196$
	$K' > K$ - No compression reinforcement is required
Lever arm	$z = \min(0.5 \times d \times [1 + (1 - 2 \times K / (\eta \times \alpha_{cc} / \gamma_c))^{0.5}], 0.95 \times d) = 413 \text{ mm}$
Depth of neutral axis	$x = 2 \times (d - z) / \lambda = 79 \text{ mm}$
Area of tension reinforcement required	$A_{s,req} = M / (f_{yd} \times z) = 728 \text{ mm}^2$
Tension reinforcement provided	$2 \times 25\phi$
Area of tension reinforcement provided	$A_{s,prov} = 982 \text{ mm}^2$
Minimum area of reinforcement - exp.9.1N	$A_{s,min} = \max(0.26 \times f_{ctm} / f_{yk}, 0.0013) \times b \times d = 178 \text{ mm}^2$
Maximum area of reinforcement - cl.9.2.1.1(3)	$A_{s,max} = 0.04 \times b \times h = 6000 \text{ mm}^2$

Project				Job no.	
Calcs for				Start page no./Revision	
				4	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

PASS - Area of reinforcement provided is greater than area of reinforcement required

Crack control - Section 7.3

Maximum crack width	$w_k = 0.30 \text{ mm}$
Design value modulus of elasticity reinf – 3.2.7(4)	$E_s = 200000 \text{ N/mm}^2$
Mean value of concrete tensile strength	$f_{ct,eff} = f_{ctm} = 2.6 \text{ N/mm}^2$
Stress distribution coefficient	$k_c = 0.4$
Non-uniform self-equilibrating stress coefficient	$k = \min(\max(1 + (300 \text{ mm} - \min(h, b)) \times 0.35 / 500 \text{ mm}, 0.65), 1) = 1.00$
Actual tension bar spacing	$s_{bar} = (b - (2 \times (c_{nom,s} + \phi_{m1,s1,z1,v}) + \phi_{m1,s1,z1,b,L1} \times N_{m1,s1,z1,b,L1})) / (N_{m1,s1,z1,b,L1} - 1) + \phi_{m1,s1,z1,b,L1} = 189 \text{ mm}$
Maximum stress permitted - Table 7.3N	$\sigma_s = 249 \text{ N/mm}^2$
Steel to concrete modulus of elast. ratio	$\alpha_{cr} = E_s / E_{cm} = 6.35$
Distance of the Elastic NA from bottom of beam	$y = (b \times h^2 / 2 + A_{s,prov} \times (\alpha_{cr} - 1) \times (h - d)) / (b \times h + A_{s,prov} \times (\alpha_{cr} - 1)) = 243 \text{ mm}$
Area of concrete in the tensile zone	$A_{ct} = b \times y = 73025 \text{ mm}^2$
Minimum area of reinforcement required - exp.7.1	$A_{s,min} = k_c \times k \times f_{ct,eff} \times A_{ct} / \sigma_s = 301 \text{ mm}^2$

PASS - Area of tension reinforcement provided exceeds minimum required for crack control

Quasi-permanent moment	$M_{QP} = \max(\beta_1 \times \text{abs}(M_{m1,s1,z2,neg_quasi}), \text{abs}(M_{m1,s1,z1,pos_quasi})) = 96.8 \text{ kNm}$
Permanent load ratio	$R_{PL} = M_{QP} / M = 0.74$
Service stress in reinforcement	$\sigma_{sr} = f_{yd} \times A_{s,req} / A_{s,prov} \times R_{PL} = 239 \text{ N/mm}^2$
Maximum bar spacing - Tables 7.3N	$s_{bar,max} = 201.5 \text{ mm}$

PASS - Maximum bar spacing exceeds actual bar spacing for crack control

Zone 1 (0 mm - 1500 mm) Negative moment - section 6.1

Design bending moment	$M = \max(\beta_1 \times \text{abs}(M_{m1,s1,max_red}), \text{abs}(M_{m1,s1,z1,min_red})) = 26.1 \text{ kNm}$
Effective depth of tension reinforcement	$d = 448 \text{ mm}$
Redistribution ratio	$\delta = 1 = 1.000$
	$K = M / (b \times d^2 \times f_{ck}) = 0.017$
	$K' = (2 \times \eta \times \alpha_{cc} / \gamma_c) \times (1 - \lambda \times (\delta - k_1) / (2 \times k_2)) \times (\lambda \times (\delta - k_1) / (2 \times k_2)) = 0.196$

K' > K - No compression reinforcement is required

Lever arm	$z = \min(0.5 \times d \times [1 + (1 - 2 \times K / (\eta \times \alpha_{cc} / \gamma_c))^{0.5}], 0.95 \times d) = 425 \text{ mm}$
Depth of neutral axis	$x = 2 \times (d - z) / \lambda = 56 \text{ mm}$
Area of tension reinforcement required	$A_{s,req} = M / (f_{yd} \times z) = 141 \text{ mm}^2$
Tension reinforcement provided	$2 \times 20\phi + 2 \times 16\phi$
Area of tension reinforcement provided	$A_{s,prov} = 1030 \text{ mm}^2$
Minimum area of reinforcement - exp.9.1N	$A_{s,min} = \max(0.26 \times f_{ctm} / f_{yk}, 0.0013) \times b \times d = 179 \text{ mm}^2$
Maximum area of reinforcement - cl.9.2.1.1(3)	$A_{s,max} = 0.04 \times b \times h = 6000 \text{ mm}^2$

PASS - Area of reinforcement provided is greater than area of reinforcement required

Crack control - Section 7.3

Maximum crack width	$w_k = 0.30 \text{ mm}$
Design value modulus of elasticity reinf – 3.2.7(4)	$E_s = 200000 \text{ N/mm}^2$
Mean value of concrete tensile strength	$f_{ct,eff} = f_{ctm} = 2.6 \text{ N/mm}^2$
Stress distribution coefficient	$k_c = 0.4$

Project				Job no.	
Calcs for				Start page no./Revision	
				5	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

Non-uniform self-equilibrating stress coefficient

$$k = \min(\max(1 + (300 \text{ mm} - \min(h, b)) \times 0.35 / 500 \text{ mm}, 0.65), 1) = \mathbf{1.00}$$

Actual tension bar spacing

$$s_{\text{bar}} = (b - (2 \times (c_{\text{nom}_s} + \phi_{m1_s1_z1_v}) + \phi_{m1_s1_z1_t_L1} \times N_{m1_s1_z1_t_L1} + \phi_{m1_s1_z2_t_L1} \times N_{m1_s1_z2_t_L1})) / ((N_{m1_s1_z1_t_L1} + N_{m1_s1_z2_t_L1}) - 1) + \phi_{m1_s1_z1_t_L1} = \mathbf{67.3 \text{ mm}}$$

Maximum stress permitted - Table 7.3N

$$\sigma_s = \mathbf{346 \text{ N/mm}^2}$$

Steel to concrete modulus of elast. ratio

$$\alpha_{cr} = E_s / E_{cm} = \mathbf{6.35}$$

Distance of the Elastic NA from bottom of beam

$$y = (b \times h^2 / 2 + A_{s,\text{prov}} \times (\alpha_{cr} - 1) \times (h - d)) / (b \times h + A_{s,\text{prov}} \times (\alpha_{cr} - 1)) = \mathbf{243 \text{ mm}}$$

Area of concrete in the tensile zone

$$A_{ct} = b \times y = \mathbf{72895 \text{ mm}^2}$$

Minimum area of reinforcement required - exp.7.1

$$A_{s,\text{c,min}} = k_c \times k \times f_{ct,\text{eff}} \times A_{ct} / \sigma_s = \mathbf{216 \text{ mm}^2}$$

PASS - Area of tension reinforcement provided exceeds minimum required for crack control

Quasi-permanent moment

$$M_{QP} = \max(\beta_1 \times \text{abs}(M_{m1_s1_z2_pos_quasi}), \text{abs}(M_{m1_s1_z1_neg_quasi})) = \mathbf{19.4 \text{ kNm}}$$

Permanent load ratio

$$R_{PL} = M_{QP} / M = \mathbf{0.74}$$

Service stress in reinforcement

$$\sigma_{sr} = f_{yd} \times A_{s,\text{req}} / A_{s,\text{prov}} \times R_{PL} = \mathbf{44 \text{ N/mm}^2}$$

Maximum bar spacing - Tables 7.3N

$$s_{\text{bar,max}} = \mathbf{300 \text{ mm}}$$

PASS - Maximum bar spacing exceeds actual bar spacing for crack control

Minimum bar spacing (Section 8.2)

Top bar spacing

$$s_{\text{top}} = (b - (2 \times (c_{\text{nom}_s} + \phi_{m1_s1_z1_v}) + \phi_{m1_s1_z1_t_L1} \times N_{m1_s1_z1_t_L1} + \phi_{m1_s1_z2_t_L1} \times N_{m1_s1_z2_t_L1})) / ((N_{m1_s1_z1_t_L1} + N_{m1_s1_z2_t_L1}) - 1) = \mathbf{47.3 \text{ mm}}$$

Minimum allowable top bar spacing

$$s_{\text{top,min}} = \max(\phi_{m1_s1_z1_t_L1} \times k_{s1}, h_{agg} + k_{s2}, 20 \text{ mm}) = \mathbf{25.0 \text{ mm}}$$

PASS - Actual bar spacing exceeds minimum allowable

Bottom bar spacing

$$s_{\text{bot}} = (b - (2 \times (c_{\text{nom}_s} + \phi_{m1_s1_z1_v}) + \phi_{m1_s1_z1_b_L1} \times N_{m1_s1_z1_b_L1})) / (N_{m1_s1_z1_b_L1} - 1) = \mathbf{164.0 \text{ mm}}$$

Minimum allowable bottom bar spacing

$$s_{\text{bot,min}} = \max(\phi_{m1_s1_z1_b_L1} \times k_{s1}, h_{agg} + k_{s2}, 20 \text{ mm}) = \mathbf{25.0 \text{ mm}}$$

PASS - Actual bar spacing exceeds minimum allowable

Zone 2 (1500 mm - 4500 mm) Positive moment - section 6.1

Design bending moment

$$M = \text{abs}(M_{m1_s1_z2_max_red}) = \mathbf{174.2 \text{ kNm}}$$

Effective depth of tension reinforcement

$$d = \mathbf{445 \text{ mm}}$$

Redistribution ratio

$$\delta = \min(M_{\text{pos_red_z2}} / M_{\text{pos_z2}}, 1) = \mathbf{1.000}$$

$$K = M / (b \times d^2 \times f_{ck}) = \mathbf{0.117}$$

$$K' = (2 \times \eta \times \alpha_{cc} / \gamma_c) \times (1 - \lambda \times (\delta - k_1) / (2 \times k_2)) \times (\lambda \times (\delta - k_1) / (2 \times k_2)) = \mathbf{0.196}$$

K' > K - No compression reinforcement is required

Lever arm

$$z = \min(0.5 \times d \times [1 + (1 - 2 \times K / (\eta \times \alpha_{cc} / \gamma_c))^{0.5}], 0.95 \times d) = \mathbf{402 \text{ mm}}$$

Depth of neutral axis

$$x = 2 \times (d - z) / \lambda = \mathbf{108 \text{ mm}}$$

Area of tension reinforcement required

$$A_{s,\text{req}} = M / (f_{yd} \times z) = \mathbf{996 \text{ mm}^2}$$

Tension reinforcement provided

$$2 \times 20\phi + 2 \times 25\phi$$

Area of tension reinforcement provided

$$A_{s,\text{prov}} = \mathbf{1610 \text{ mm}^2}$$

Minimum area of reinforcement - exp.9.1N

$$A_{s,\text{min}} = \max(0.26 \times f_{ctm} / f_{yk}, 0.0013) \times b \times d = \mathbf{178 \text{ mm}^2}$$

Maximum area of reinforcement - cl.9.2.1.1(3)

$$A_{s,\text{max}} = 0.04 \times b \times h = \mathbf{6000 \text{ mm}^2}$$

PASS - Area of reinforcement provided is greater than area of reinforcement required

Project				Job no.	
Calcs for				Start page no./Revision	
				6	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

Crack control - Section 7.3

Maximum crack width

$$w_k = 0.30 \text{ mm}$$

Design value modulus of elasticity reinf – 3.2.7(4)

$$E_s = 200000 \text{ N/mm}^2$$

Mean value of concrete tensile strength

$$f_{ct,eff} = f_{ctm} = 2.6 \text{ N/mm}^2$$

Stress distribution coefficient

$$k_c = 0.4$$

Non-uniform self-equilibrating stress coefficient

$$k = \min(\max(1 + (300 \text{ mm} - \min(h, b)) \times 0.35 / 500 \text{ mm}, 0.65), 1) = 1.00$$

Actual tension bar spacing

$$s_{bar} = (b - (2 \times (c_{nom,s} + \phi_{m1,s1,z2,v}) + \phi_{m1,s1,z2,b,L1} \times N_{m1,s1,z2,b,L1} + \phi_{m1,s1,z1,b,L1} \times N_{m1,s1,z1,b,L1})) / ((N_{m1,s1,z2,b,L1} + N_{m1,s1,z1,b,L1}) - 1) + \phi_{m1,s1,z1,b,L1} = 66.3 \text{ mm}$$

Maximum stress permitted - Table 7.3N

$$\sigma_s = 347 \text{ N/mm}^2$$

Steel to concrete modulus of elast. ratio

$$\alpha_{cr} = E_s / E_{cm} = 6.35$$

Distance of the Elastic NA from bottom of beam

$$y = (b \times h^2 / 2 + A_{s,prov} \times (\alpha_{cr} - 1) \times (h - d)) / (b \times h + A_{s,prov} \times (\alpha_{cr} - 1)) = 239 \text{ mm}$$

Area of concrete in the tensile zone

$$A_{ct} = b \times y = 71813 \text{ mm}^2$$

Minimum area of reinforcement required - exp.7.1

$$A_{s,min} = k_c \times k \times f_{ct,eff} \times A_{ct} / \sigma_s = 212 \text{ mm}^2$$

PASS - Area of tension reinforcement provided exceeds minimum required for crack control

Quasi-permanent moment

$$M_{QP} = \text{abs}(M_{m1,s1,z2,pos,quasi}) = 129.0 \text{ kNm}$$

Permanent load ratio

$$R_{PL} = M_{QP} / M = 0.74$$

Service stress in reinforcement

$$\sigma_{sr} = f_{yd} \times A_{s,req} / A_{s,prov} \times R_{PL} = 199 \text{ N/mm}^2$$

Maximum bar spacing - Tables 7.3N

$$s_{bar,max} = 250.9 \text{ mm}$$

PASS - Maximum bar spacing exceeds actual bar spacing for crack control

Deflection control - Section 7.4

Reference reinforcement ratio

$$\rho_{m0} = (f_{ck} / 1 \text{ N/mm}^2)^{0.5} / 1000 = 0.00500$$

Required tension reinforcement ratio

$$\rho_m = A_{s,req} / (b \times d) = 0.00746$$

Required compression reinforcement ratio

$$\rho'_m = A_{s2,req} / (b \times d) = 0.00000$$

Structural system factor - Table 7.4N

$$K_b = 1.0$$

Basic allowable span to depth ratio

$$\text{span_to_depth}_{basic} = K_b \times [11 + 1.5 \times (f_{ck} / 1 \text{ N/mm}^2)^{0.5} \times \rho_{m0} / (\rho_m - \rho'_m) + (f_{ck} / 1 \text{ N/mm}^2)^{0.5} \times (\rho'_m / \rho_{m0})^{0.5} / 12] = 16.030$$

Reinforcement factor - exp.7.17

$$K_s = \min(A_{s,prov} / A_{s,req} \times 500 \text{ N/mm}^2 / f_{yk}, 1.5) = 1.500$$

Flange width factor

$$F1 = 1 = 1.000$$

Long span supporting brittle partition factor

$$F2 = 1 = 1.000$$

Allowable span to depth ratio

$$\text{span_to_depth}_{allow} = \min(\text{span_to_depth}_{basic} \times K_s \times F1 \times F2, 40 \times K_b) = 24.045$$

Actual span to depth ratio

$$\text{span_to_depth}_{actual} = L_{m1,s1} / d = 13.469$$

PASS - Actual span to depth ratio is within the allowable limit

Minimum bar spacing (Section 8.2)

Top bar spacing

$$s_{top} = (b - (2 \times (c_{nom,s} + \phi_{m1,s1,z2,v}) + \phi_{m1,s1,z2,t,L1} \times N_{m1,s1,z2,t,L1})) / (N_{m1,s1,z2,t,L1} - 1) = 182.0 \text{ mm}$$

Minimum allowable top bar spacing

$$s_{top,min} = \max(\phi_{m1,s1,z2,t,L1} \times k_{s1}, h_{agg} + k_{s2}, 20 \text{ mm}) = 25.0 \text{ mm}$$

PASS - Actual bar spacing exceeds minimum allowable

Bottom bar spacing

$$s_{bot} = (b - (2 \times (c_{nom,s} + \phi_{m1,s1,z2,v}) + \phi_{m1,s1,z2,b,L1} \times N_{m1,s1,z2,b,L1} + \phi_{m1,s1,z1,b,L1} \times N_{m1,s1,z1,b,L1})) / ((N_{m1,s1,z2,b,L1} + N_{m1,s1,z1,b,L1}) - 1) = 41.3 \text{ mm}$$

Minimum allowable bottom bar spacing

$$s_{bot,min} = \max(\phi_{m1,s1,z1,b,L1} \times k_{s1}, h_{agg} + k_{s2}, 20 \text{ mm}) = 25.0 \text{ mm}$$

Project				Job no.	
Calcs for				Start page no./Revision	
				7	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

PASS - Actual bar spacing exceeds minimum allowable

Minimum bar spacing (Section 8.2)

Top bar spacing

$$S_{top} = (b - (2 \times (c_{nom_s} + \phi_{m1_s1_z2_v}) + \phi_{m1_s1_z2r_t_L1} \times N_{m1_s1_z2r_t_L1})) / (N_{m1_s1_z2r_t_L1} - 1) = 182.0 \text{ mm}$$

Minimum allowable top bar spacing

$$S_{top,min} = \max(\phi_{m1_s1_z2r_t_L1} \times k_{s1}, h_{agg} + k_{s2}, 20\text{mm}) = 25.0 \text{ mm}$$

PASS - Actual bar spacing exceeds minimum allowable

Zone 3 (4500 mm - 6000 mm) Positive moment - section 6.1

Design bending moment

$$M = \text{abs}(M_{m1_s1_z3_max_red}) = 130.7 \text{ kNm}$$

Effective depth of tension reinforcement

$$d = 445 \text{ mm}$$

Redistribution ratio

$$\delta = \min(M_{pos_red_z3} / M_{pos_z3}, 1) = 1.000$$

$$K = M / (b \times d^2 \times f_{ck}) = 0.088$$

$$K' = (2 \times \eta \times \alpha_{cc} / \gamma_c) \times (1 - \lambda \times (\delta - k_1) / (2 \times k_2)) \times (\lambda \times (\delta - k_1) / (2 \times k_2)) = 0.196$$

K' > K - No compression reinforcement is required

Lever arm

$$z = \min(0.5 \times d \times [1 + (1 - 2 \times K / (\eta \times \alpha_{cc} / \gamma_c))^{0.5}], 0.95 \times d) = 413 \text{ mm}$$

Depth of neutral axis

$$x = 2 \times (d - z) / \lambda = 79 \text{ mm}$$

Area of tension reinforcement required

$$A_{s,req} = M / (f_{yd} \times z) = 728 \text{ mm}^2$$

Tension reinforcement provided

$$2 \times 25\phi$$

Area of tension reinforcement provided

$$A_{s,prov} = 982 \text{ mm}^2$$

Minimum area of reinforcement - exp.9.1N

$$A_{s,min} = \max(0.26 \times f_{ctm} / f_{yk}, 0.0013) \times b \times d = 178 \text{ mm}^2$$

Maximum area of reinforcement - cl.9.2.1.1(3)

$$A_{s,max} = 0.04 \times b \times h = 6000 \text{ mm}^2$$

PASS - Area of reinforcement provided is greater than area of reinforcement required

Crack control - Section 7.3

Maximum crack width

$$w_k = 0.30 \text{ mm}$$

Design value modulus of elasticity reinf - 3.2.7(4)

$$E_s = 200000 \text{ N/mm}^2$$

Mean value of concrete tensile strength

$$f_{ct,eff} = f_{ctm} = 2.6 \text{ N/mm}^2$$

Stress distribution coefficient

$$K_c = 0.4$$

Non-uniform self-equilibrating stress coefficient

$$k = \min(\max(1 + (300 \text{ mm} - \min(h, b)) \times 0.35 / 500 \text{ mm}, 0.65), 1) = 1.00$$

Actual tension bar spacing

$$S_{bar} = (b - (2 \times (c_{nom_s} + \phi_{m1_s1_z3_v}) + \phi_{m1_s1_z3_b_L1} \times N_{m1_s1_z3_b_L1})) / (N_{m1_s1_z3_b_L1} - 1) + \phi_{m1_s1_z3_b_L1} = 189 \text{ mm}$$

Maximum stress permitted - Table 7.3N

$$\sigma_s = 249 \text{ N/mm}^2$$

Steel to concrete modulus of elast. ratio

$$\alpha_{cr} = E_s / E_{cm} = 6.35$$

Distance of the Elastic NA from bottom of beam

$$y = (b \times h^2 / 2 + A_{s,prov} \times (\alpha_{cr} - 1) \times (h - d)) / (b \times h + A_{s,prov} \times (\alpha_{cr} - 1)) = 243 \text{ mm}$$

Area of concrete in the tensile zone

$$A_{ct} = b \times y = 73025 \text{ mm}^2$$

Minimum area of reinforcement required - exp.7.1

$$A_{s,min} = K_c \times k \times f_{ct,eff} \times A_{ct} / \sigma_s = 301 \text{ mm}^2$$

PASS - Area of tension reinforcement provided exceeds minimum required for crack control

Quasi-permanent moment

$$M_{QP} = \max(\beta_1 \times \text{abs}(M_{m1_s1_z2_neg_quasi}), \text{abs}(M_{m1_s1_z3_pos_quasi})) = 96.8 \text{ kNm}$$

Permanent load ratio

$$R_{PL} = M_{QP} / M = 0.74$$

Service stress in reinforcement

$$\sigma_{sr} = f_{yd} \times A_{s,req} / A_{s,prov} \times R_{PL} = 239 \text{ N/mm}^2$$

Maximum bar spacing - Tables 7.3N

$$S_{bar,max} = 201.5 \text{ mm}$$

PASS - Maximum bar spacing exceeds actual bar spacing for crack control

Project				Job no.	
Calcs for				Start page no./Revision	
				8	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

Zone 3 (4500 mm - 6000 mm) Negative moment - section 6.1

Design bending moment	$M = \max(\beta_1 \times \text{abs}(M_{m1_s1_max_red}), \text{abs}(M_{m1_s1_z3_min_red})) = 26.1 \text{ kNm}$
Effective depth of tension reinforcement	$d = 448 \text{ mm}$
Redistribution ratio	$\delta = 1 = 1.000$
	$K = M / (b \times d^2 \times f_{ck}) = 0.017$
	$K' = (2 \times \eta \times \alpha_{cc} / \gamma_c) \times (1 - \lambda \times (\delta - k_1) / (2 \times k_2)) \times (\lambda \times (\delta - k_1) / (2 \times k_2)) = 0.196$
	$K' > K$ - No compression reinforcement is required
Lever arm	$z = \min(0.5 \times d \times [1 + (1 - 2 \times K / (\eta \times \alpha_{cc} / \gamma_c))^{0.5}], 0.95 \times d) = 425 \text{ mm}$
Depth of neutral axis	$x = 2 \times (d - z) / \lambda = 56 \text{ mm}$
Area of tension reinforcement required	$A_{s,req} = M / (f_{yd} \times z) = 141 \text{ mm}^2$
Tension reinforcement provided	$2 \times 20\phi + 2 \times 16\phi$
Area of tension reinforcement provided	$A_{s,prov} = 1030 \text{ mm}^2$
Minimum area of reinforcement - exp.9.1N	$A_{s,min} = \max(0.26 \times f_{ctm} / f_{yk}, 0.0013) \times b \times d = 179 \text{ mm}^2$
Maximum area of reinforcement - cl.9.2.1.1(3)	$A_{s,max} = 0.04 \times b \times h = 6000 \text{ mm}^2$

PASS - Area of reinforcement provided is greater than area of reinforcement required

Crack control - Section 7.3

Maximum crack width	$w_k = 0.30 \text{ mm}$
Design value modulus of elasticity reinf - 3.2.7(4)	$E_s = 200000 \text{ N/mm}^2$
Mean value of concrete tensile strength	$f_{ct,eff} = f_{ctm} = 2.6 \text{ N/mm}^2$
Stress distribution coefficient	$K_c = 0.4$
Non-uniform self-equilibrating stress coefficient	$k = \min(\max(1 + (300 \text{ mm} - \min(h, b)) \times 0.35 / 500 \text{ mm}, 0.65), 1) = 1.00$
Actual tension bar spacing	$S_{bar} = (b - (2 \times (C_{nom_s} + \phi_{m1_s1_z3_v}) + \phi_{m1_s1_z3_t_L1} \times N_{m1_s1_z3_t_L1} + \phi_{m1_s1_z2r_t_L1} \times N_{m1_s1_z2r_t_L1})) / ((N_{m1_s1_z3_t_L1} + N_{m1_s1_z2r_t_L1}) - 1) + \phi_{m1_s1_z3_t_L1} = 67.3 \text{ mm}$
Maximum stress permitted - Table 7.3N	$\sigma_s = 346 \text{ N/mm}^2$
Steel to concrete modulus of elast. ratio	$\alpha_{cr} = E_s / E_{cm} = 6.35$
Distance of the Elastic NA from bottom of beam	$y = (b \times h^2 / 2 + A_{s,prov} \times (\alpha_{cr} - 1) \times (h - d)) / (b \times h + A_{s,prov} \times (\alpha_{cr} - 1)) = 243 \text{ mm}$
Area of concrete in the tensile zone	$A_{ct} = b \times y = 72895 \text{ mm}^2$
Minimum area of reinforcement required - exp.7.1	$A_{s,min} = K_c \times k \times f_{ct,eff} \times A_{ct} / \sigma_s = 216 \text{ mm}^2$

PASS - Area of tension reinforcement provided exceeds minimum required for crack control

Quasi-permanent moment	$M_{QP} = \max(\beta_1 \times \text{abs}(M_{m1_s1_z2_pos_quasi}), \text{abs}(M_{m1_s1_z3_neg_quasi})) = 19.4 \text{ kNm}$
Permanent load ratio	$R_{PL} = M_{QP} / M = 0.74$
Service stress in reinforcement	$\sigma_{sr} = f_{yd} \times A_{s,req} / A_{s,prov} \times R_{PL} = 44 \text{ N/mm}^2$
Maximum bar spacing - Tables 7.3N	$S_{bar,max} = 300 \text{ mm}$

PASS - Maximum bar spacing exceeds actual bar spacing for crack control

Minimum bar spacing (Section 8.2)

Top bar spacing	$S_{top} = (b - (2 \times (C_{nom_s} + \phi_{m1_s1_z3_v}) + \phi_{m1_s1_z3_t_L1} \times N_{m1_s1_z3_t_L1} + \phi_{m1_s1_z2r_t_L1} \times N_{m1_s1_z2r_t_L1})) / ((N_{m1_s1_z3_t_L1} + N_{m1_s1_z2r_t_L1}) - 1) = 47.3 \text{ mm}$
Minimum allowable top bar spacing	$S_{top,min} = \max(\phi_{m1_s1_z3_t_L1} \times k_{s1}, h_{agg} + k_{s2}, 20 \text{ mm}) = 25.0 \text{ mm}$

PASS - Actual bar spacing exceeds minimum allowable

Project				Job no.	
Calcs for				Start page no./Revision	
				9	
Calcs by	Calcs date	Checked by	Checked date	Approved by	Approved date
J	29/05/2024				

Bottom bar spacing

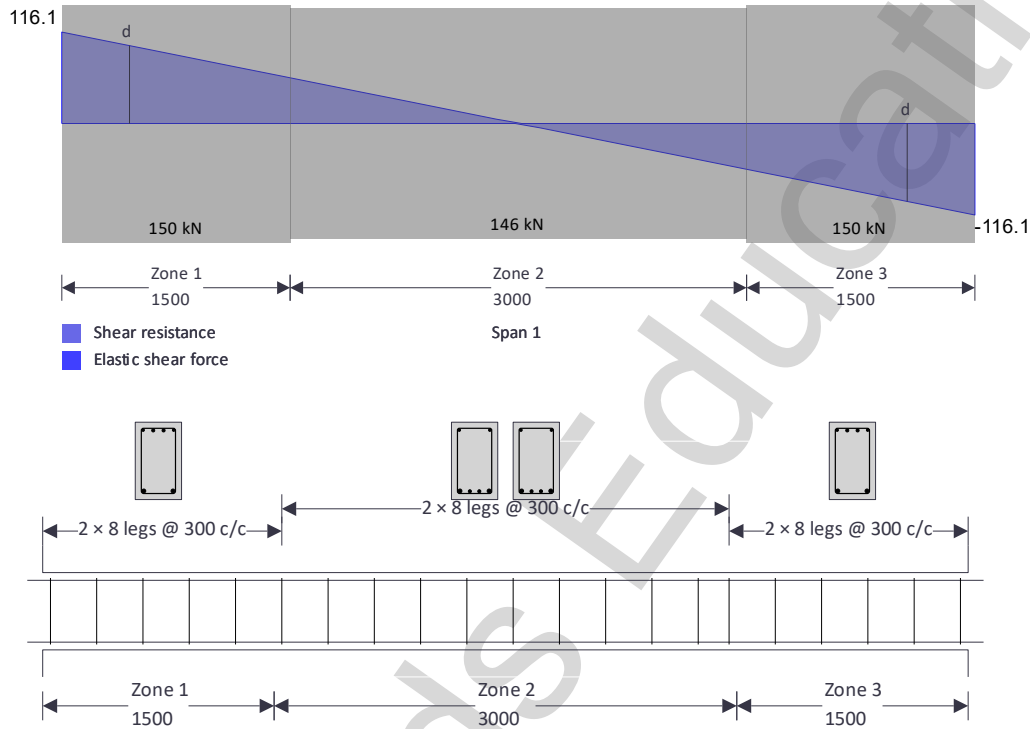
$$s_{bot} = (b - (2 \times (c_{nom_s} + \phi_{m1_s1_z3_v}) + \phi_{m1_s1_z3_b_L1} \times N_{m1_s1_z3_b_L1})) / (N_{m1_s1_z3_b_L1} - 1) = 164.0 \text{ mm}$$

Minimum allowable bottom bar spacing

$$s_{bot,min} = \max(\phi_{m1_s1_z3_b_L1} \times k_{s1}, h_{agg} + k_{s2}, 20\text{mm}) = 25.0 \text{ mm}$$

PASS - Actual bar spacing exceeds minimum allowable

Shear design



Angle of comp. shear strut for maximum shear

$$\theta_{max} = 45 \text{ deg}$$

Strength reduction factor - cl.6.2.3(3)

$$v_1 = 0.6 \times (1 - f_{ck} / 250 \text{ N/mm}^2) = 0.540$$

Compression chord coefficient - cl.6.2.3(3)

$$\alpha_{cw} = 1.00$$

Minimum area of shear reinforcement - exp.9.5N

$$A_{sv,min} = 0.08 \text{ N/mm}^2 \times b \times (f_{ck} / 1 \text{ N/mm}^2)^{0.5} / f_{yk} = 240 \text{ mm}^2/\text{m}$$

Zone 1 (0 mm - 1500 mm) shear - section 6.2

Design shear force at support

$$V_{Ed,max} = \max(\text{abs}(V_{z1_max}), \text{abs}(V_{z1_red_max})) = 116 \text{ kN}$$

Min lever arm in shear zone

$$z = 413 \text{ mm}$$

Maximum design shear resistance - exp.6.9

$$V_{Rd,max} = \alpha_{cw} \times b \times z \times v_1 \times f_{cwd} / (\cot(\theta_{max}) + \tan(\theta_{max})) = 557 \text{ kN}$$

PASS - Design shear force at support is less than maximum design shear resistance

Design shear force at 445mm from support

$$V_{Ed} = 99 \text{ kN}$$

Design shear stress

$$v_{Ed} = V_{Ed} / (b \times z) = 0.799 \text{ N/mm}^2$$

Angle of concrete compression strut - cl.6.2.3

$$\theta = \min(\max(0.5 \times \text{Asin}(\min(2 \times v_{Ed} / (\alpha_{cw} \times f_{cwd} \times v_1)), 1)), 21.8 \text{ deg}), 45\text{deg}) = 21.8 \text{ deg}$$

Area of shear reinforcement required - exp.6.8

$$A_{sv,des} = v_{Ed} \times b / (f_{yd} \times \cot(\theta)) = 220 \text{ mm}^2/\text{m}$$

Area of shear reinforcement required

$$A_{sv,req} = \max(A_{sv,min}, A_{sv,des}) = 240 \text{ mm}^2/\text{m}$$

Shear reinforcement provided

$$2 \times 8 \text{ legs @ } 300 \text{ c/c}$$

Area of shear reinforcement provided

$$A_{sv,prov} = 335 \text{ mm}^2/\text{m}$$

PASS - Area of shear reinforcement provided exceeds minimum required

Maximum longitudinal spacing - exp.9.6N

$$s_{vl,max} = 0.75 \times d = 333 \text{ mm}$$

PASS - Longitudinal spacing of shear reinforcement provided is less than maximum

Project				Job no.	
Calcs for				Start page no./Revision 10	
Calcs by J	Calcs date 29/05/2024	Checked by	Checked date	Approved by	Approved date

Zone 2 (1500 mm - 4500 mm) shear - section 6.2

Design shear force at support

$$V_{Ed,max} = \max(\text{abs}(V_{z2_max}), \text{abs}(V_{z2_red_max})) = 58 \text{ kN}$$

Min lever arm in shear zone

$$z = 402 \text{ mm}$$

Maximum design shear resistance - exp.6.9

$$V_{Rd,max} = \alpha_{cw} \times b \times z \times v_1 \times f_{cwd} / (\cot(\theta_{max}) + \tan(\theta_{max})) = 543 \text{ kN}$$

PASS - Design shear force at support is less than maximum design shear resistance

Design shear force within zone

$$V_{Ed} = 58 \text{ kN}$$

Design shear stress

$$v_{Ed} = V_{Ed} / (b \times z) = 0.481 \text{ N/mm}^2$$

Angle of concrete compression strut - cl.6.2.3

$$\theta = \min(\max(0.5 \times \text{Asin}(\min(2 \times v_{Ed} / (\alpha_{cw} \times f_{cwd} \times v_1), 1)), 21.8 \text{ deg}), 45 \text{ deg}) = 21.8 \text{ deg}$$

Area of shear reinforcement required - exp.6.8

$$A_{sv,des} = V_{Ed} \times b / (f_{yd} \times \cot(\theta)) = 133 \text{ mm}^2/\text{m}$$

Area of shear reinforcement required

$$A_{sv,req} = \max(A_{sv,min}, A_{sv,des}) = 240 \text{ mm}^2/\text{m}$$

Shear reinforcement provided

$$2 \times 8 \text{ legs @ } 300 \text{ c/c}$$

Area of shear reinforcement provided

$$A_{sv,prov} = 335 \text{ mm}^2/\text{m}$$

PASS - Area of shear reinforcement provided exceeds minimum required

Maximum longitudinal spacing - exp.9.6N

$$s_{vl,max} = 0.75 \times d = 334 \text{ mm}$$

PASS - Longitudinal spacing of shear reinforcement provided is less than maximum

Zone 3 (4500 mm - 6000 mm) shear - section 6.2

Design shear force at support

$$V_{Ed,max} = \max(\text{abs}(V_{z3_max}), \text{abs}(V_{z3_red_max})) = 116 \text{ kN}$$

Min lever arm in shear zone

$$z = 413 \text{ mm}$$

Maximum design shear resistance - exp.6.9

$$V_{Rd,max} = \alpha_{cw} \times b \times z \times v_1 \times f_{cwd} / (\cot(\theta_{max}) + \tan(\theta_{max})) = 557 \text{ kN}$$

PASS - Design shear force at support is less than maximum design shear resistance

Design shear force at 445mm from support

$$V_{Ed} = 99 \text{ kN}$$

Design shear stress

$$v_{Ed} = V_{Ed} / (b \times z) = 0.799 \text{ N/mm}^2$$

Angle of concrete compression strut - cl.6.2.3

$$\theta = \min(\max(0.5 \times \text{Asin}(\min(2 \times v_{Ed} / (\alpha_{cw} \times f_{cwd} \times v_1), 1)), 21.8 \text{ deg}), 45 \text{ deg}) = 21.8 \text{ deg}$$

Area of shear reinforcement required - exp.6.8

$$A_{sv,des} = V_{Ed} \times b / (f_{yd} \times \cot(\theta)) = 220 \text{ mm}^2/\text{m}$$

Area of shear reinforcement required

$$A_{sv,req} = \max(A_{sv,min}, A_{sv,des}) = 240 \text{ mm}^2/\text{m}$$

Shear reinforcement provided

$$2 \times 8 \text{ legs @ } 300 \text{ c/c}$$

Area of shear reinforcement provided

$$A_{sv,prov} = 335 \text{ mm}^2/\text{m}$$

PASS - Area of shear reinforcement provided exceeds minimum required

Maximum longitudinal spacing - exp.9.6N

$$s_{vl,max} = 0.75 \times d = 333 \text{ mm}$$

PASS - Longitudinal spacing of shear reinforcement provided is less than maximum

Bijlage C - Eurocode 2 regels die niet van toepassing zijn

Deze bijlage biedt een opsomming van de hoofdstukken, beginselen en toepassingsregels van Eurocode 2 die relevant zijn voor het ontwerpen van betonnen balken, maar die in deze masterproef buiten beschouwing zijn gelaten:

- NBN EN 1992-1-1: 2.3.1.3;
- NBN EN 1992-1-1: 2.3.1.4;
- NBN EN 1992-1-1: 2.4.2.1;
- NBN EN 1992-1-1: 2.4.2.2;
- NBN EN 1992-1-1: 2.4.2.3;
- NBN EN 1992-1-1: 2.4.2.5;
- NBN EN 1992-1-1: 2.5;
- NBN EN 1992-1-1: 2.6;
- NBN EN 1992-1-1: 2.7;
- NBN EN 1992-1-1: 3.1.5;
- NBN EN 1992-1-1: 3.1.9;
- NBN EN 1992-1-1: 3.2.6;
- NBN EN 1992-1-1: 3.3;
- NBN EN 1992-1-1: 3.4;
- NBN EN 1992-1-1: 5.1.2;
- NBN EN 1992-1-1: 5.1.3 (1)P (a);
- NBN EN 1992-1-1: 5.1.3 (1)P (b);
- NBN EN 1992-1-1: 5.1.4;
- NBN EN 1992-1-1: 5.2;
- NBN EN 1992-1-1: 5.3.2.2;
- NBN EN 1992-1-1: 5.6;

- NBN EN 1992-1-1: 5.8;
- NBN EN 1992-1-1: 5.9;
- NBN EN 1992-1-1: 5.10;
- NBN EN 1992-1-1: 6.1 (4);
- NBN EN 1992-1-1: 6.1 (5);
- NBN EN 1992-1-1: 6.1 (7);
- NBN EN 1992-1-1: 6.1 (8);
- NBN EN 1992-1-1: 6.2.4;
- NBN EN 1992-1-1: 6.2.5;
- NBN EN 1992-1-1: 6.3;
- NBN EN 1992-1-1: 6.5;
- NBN EN 1992-1-1: 6.7;
- NBN EN 1992-1-1: 6.8;
- NBN EN 1992-1-1: 8.5;
- NBN EN 1992-1-1: 8.6;
- NBN EN 1992-1-1: 8.7;
- NBN EN 1992-1-1: 8.8;
- NBN EN 1992-1-1: 8.9;
- NBN EN 1992-1-1: 8.10;
- NBN EN 1992-1-1: 9.7;
- NBN EN 1992-1-1: 9.9;
- NBN EN 1992-1-1: 10;
- NBN EN 1992-1-1: 11;
- NBN EN 1992-1-1: 12;
- NBN EN 1992-1-2;
- NBN EN 1992-2;
- NBN EN 1992-3.

Bijlage D - Tabel B4.2.3.7-a

Tabel B4.2.3.7-a Tabel voor ontwerpberekening in de UGT;
enkelgewapende rechthoekige doorsnede; enkelvoudige buiging

betonklassen $\leq C50/60$ - $\varepsilon_{ud} = 10\%$ - $f_{cd} = 0,85 \cdot f_{ck} / 1,5$

μ_d	ω	$\xi=x/d$	$\zeta=z/d$	ε_c (‰)	ε_s (‰)	ψ	δ_G	σ_s (MPa)			
								S220	S400	S500	S600
0,010	0,0102	0,046	0,984	0,48	10,00	0,2213	0,3407	191	348	435	522
0,020	0,0205	0,066	0,977	0,70	10,00	0,3109	0,3445	191	348	435	522
0,030	0,0309	0,082	0,972	0,89	10,00	0,3784	0,3478	191	348	435	522
0,040	0,0414	0,095	0,967	1,05	10,00	0,4342	0,3511	191	348	435	522
0,050	0,0520	0,108	0,962	1,21	10,00	0,4824	0,3543	191	348	435	522
0,060	0,0627	0,119	0,957	1,36	10,00	0,5248	0,3577	191	348	435	522
0,070	0,0735	0,131	0,953	1,50	10,00	0,5628	0,3611	191	348	435	522
0,080	0,0843	0,141	0,948	1,65	10,00	0,5970	0,3648	191	348	435	522
0,090	0,0953	0,152	0,944	1,79	10,00	0,6280	0,3688	191	348	435	522
0,100	0,1064	0,162	0,939	1,94	10,00	0,6559	0,3731	191	348	435	522
0,104	0,1111	0,167	0,937	2,00	10,00	0,6667	0,3750	191	348	435	522
0,110	0,1177	0,173	0,935	2,09	10,00	0,6809	0,3778	191	348	435	522
0,120	0,1291	0,184	0,930	2,25	10,00	0,7034	0,3830	191	348	435	522
0,130	0,1406	0,194	0,925	2,41	10,00	0,7236	0,3882	191	348	435	522
0,140	0,1523	0,205	0,919	2,58	10,00	0,7419	0,3934	191	348	435	522
0,150	0,1642	0,216	0,914	2,76	10,00	0,7586	0,3985	191	348	435	522
0,160	0,1762	0,228	0,908	2,95	10,00	0,7738	0,4034	191	348	435	522
0,170	0,1884	0,239	0,902	3,14	10,00	0,7879	0,4082	191	348	435	522
0,180	0,2008	0,251	0,897	3,35	10,00	0,8008	0,4128	191	348	435	522
0,187	0,2099	0,259	0,892	3,50	10,00	0,8095	0,4160	191	348	435	522
0,190	0,2134	0,264	0,890	3,50	9,78	0,8095	0,4160	191	348	435	522
0,200	0,2263	0,280	0,884	3,50	9,02	0,8095	0,4160	191	348	435	522
0,210	0,2395	0,296	0,877	3,50	8,33	0,8095	0,4160	191	348	435	522
0,220	0,2529	0,312	0,870	3,50	7,71	0,8095	0,4160	191	348	435	522
0,230	0,2665	0,329	0,863	3,50	7,13	0,8095	0,4160	191	348	435	522
0,240	0,2804	0,346	0,856	3,50	6,60	0,8095	0,4160	191	348	435	522
0,250	0,2946	0,364	0,849	3,50	6,12	0,8095	0,4160	191	348	435	522
0,260	0,3091	0,382	0,841	3,50	5,67	0,8095	0,4160	191	348	435	522
0,270	0,3239	0,400	0,834	3,50	5,25	0,8095	0,4160	191	348	435	522
0,280	0,3391	0,419	0,826	3,50	4,86	0,8095	0,4160	191	348	435	522
0,290	0,3546	0,438	0,818	3,50	4,49	0,8095	0,4160	191	348	435	522
0,300	0,3706	0,458	0,810	3,50	4,15	0,8095	0,4160	191	348	435	522
0,310	0,3869	0,478	0,801	3,50	3,82	0,8095	0,4160	191	348	435	522
0,320	0,4038	0,499	0,793	3,50	3,52	0,8095	0,4160	191	348	435	522
0,330	0,4211	0,520	0,784	3,50	3,23	0,8095	0,4160	191	348	435	522
0,340	0,4391	0,542	0,774	3,50	2,95	0,8095	0,4160	191	348	435	522
0,350	0,4576	0,565	0,765	3,50	2,69	0,8095	0,4160	191	348	435	522
0,353	0,4638	0,573	0,762	3,50	2,61	0,8095	0,4160	191	348	435	522
0,360	0,4768	0,589	0,755	3,50	2,44	0,8095	0,4160	191	348	435	488
0,370	0,4968	0,614	0,745	3,50	2,20	0,8095	0,4160	191	348	435	441
0,371	0,4994	0,617	0,743	3,50	2,17	0,8095	0,4160	191	348	435	435
0,380	0,5177	0,640	0,734	3,50	1,97	0,8095	0,4160	191	348	395	395
0,391	0,5408	0,668	0,722	3,50	1,74	0,8095	0,4160	191	348	348	348
0,400	0,5627	0,695	0,711	3,50	1,54	0,8095	0,4160	191	307	307	307
0,410	0,5871	0,725	0,698	3,50	1,33	0,8095	0,4160	191	265	265	265
0,420	0,6132	0,758	0,685	3,50	1,12	0,8095	0,4160	191	224	224	224
0,428	0,6358	0,785	0,673	3,50	0,96	0,8095	0,4160	191	191	191	191
0,430	0,6414	0,792	0,670	3,50	0,92	0,8095	0,4160	184	184	184	184
0,440	0,6721	0,830	0,655	3,50	0,72	0,8095	0,4160	143	143	143	143
0,450	0,7064	0,873	0,637	3,50	0,51	0,8095	0,4160	102	102	102	102
0,460	0,7458	0,921	0,617	3,50	0,30	0,8095	0,4160	60	60	60	60
0,470	0,7937	0,980	0,592	3,50	0,07	0,8095	0,4160	14	14	14	14
0,473	0,8095	1,000	0,584	3,50	-	0,8095	0,4160	0	0	0	0

Bijlage E - Eurocode 3 regels die niet van toepassing zijn

Deze bijlage biedt een opsomming van de hoofdstukken, beginselen en toepassingsregels van Eurocode 3 die relevant zijn voor het ontwerpen van stalen balken, maar die in deze masterproef buiten beschouwing zijn gelaten:

- NBN EN 1993-1-1: 3.2.4;
- NBN EN 1993-1-1: 3.3;
- NBN EN 1993-1-1: 3.4;
- NBN EN 1993-1-1: 5.1.2;
- NBN EN 1993-1-1: 5.1.3;
- NBN EN 1993-1-1: 5.2.1;
- NBN EN 1993-1-1: 5.2.2;
- NBN EN 1993-1-1: 5.3.2;
- NBN EN 1993-1-1: 5.3.3;
- NBN EN 1993-1-1: 6.2.1 (2);
- NBN EN 1993-1-1: 6.2.2.2;
- NBN EN 1993-1-1: 6.2.2.3;
- NBN EN 1993-1-1: 6.2.2.4;
- NBN EN 1993-1-1: 6.2.2.5;
- NBN EN 1993-1-1: 6.2.3;
- NBN EN 1993-1-1: 6.2.4;
- NBN EN 1993-1-1: 6.2.5 (3);
- NBN EN 1993-1-1: 6.2.5 (4);
- NBN EN 1993-1-1: 6.2.5 (5);
- NBN EN 1993-1-1: 6.2.5 (6);
- NBN EN 1993-1-1: 6.2.6 (4);

- NBN EN 1993-1-1: 6.2.6 (5);
- NBN EN 1993-1-1: 6.2.6 (6);
- NBN EN 1993-1-1: 6.2.6 (7);
- NBN EN 1993-1-1: 6.2.6 (8);
- NBN EN 1993-1-1: 6.2.7;
- NBN EN 1993-1-1: 6.2.8 (6);
- NBN EN 1993-1-1: 6.2.9;
- NBN EN 1993-1-1: 6.2.10;
- NBN EN 1993-1-1: 6.3.1;
- NBN EN 1993-1-1: 6.3.2.1;
- NBN EN 1993-1-1: 6.3.2.3;
- NBN EN 1993-1-1: 6.3.2.4;
- NBN EN 1993-1-1: 6.3.3;
- NBN EN 1993-1-1: 6.3.4;
- NBN EN 1993-1-1: 6.3.5;
- NBN EN 1993-1-1: 7.1 (4);
- NBN EN 1993-1-1: 7.2.2;
- NBN EN 1993-1-1: 7.2.3;
- NBN EN 1993-1-2;
- NBN EN 1993-1-3;
- NBN EN 1993-1-4;
- NBN EN 1993-1-9;
- NBN EN 1993-1-10;
- NBN EN 1993-1-11;
- NBN EN 1993-1-12;
- NBN EN 1993-2;
- NBN EN 1993-3;
- NBN EN 1993-4;
- NBN EN 1993-6.

Bijlage F - Eurocode 5 regels die niet van toepassing zijn

Deze bijlage biedt een opsomming van de hoofdstukken, beginselen en toepassingsregels van Eurocode 5 die relevant zijn voor het ontwerpen van houten balken, maar die in deze masterproef buiten beschouwing zijn gelaten:

- NBN EN 1995-1-1: 2.2.2 (2);
- NBN EN 1995-1-1: 2.2.3 (3);
- NBN EN 1995-1-1: 2.2.3 (4);
- NBN EN 1995-1-1: 2.2.3 (6);
- NBN EN 1995-1-1: 2.2.1.1 (3)P;
- NBN EN 1995-1-1: 2.3.2.1 (2);
- NBN EN 1995-1-1: 2.3.2.2 (2);
- NBN EN 1995-1-1: 3.1.2 (2);
- NBN EN 1995-1-1: 3.2 (4);
- NBN EN 1995-1-1: 3.2 (5)P;
- NBN EN 1995-1-1: 3.3 (4)P;
- NBN EN 1995-1-1: 3.3 (5)P;
- NBN EN 1995-1-1: 3.4;
- NBN EN 1995-1-1: 4;
- NBN EN 1995-1-1: 5.1 (3);
- NBN EN 1995-1-1: 5.1 (4)P;
- NBN EN 1995-1-1: 5.1 (5)P;
- NBN EN 1995-1-1: 5.2 (2)P;
- NBN EN 1995-1-1: 5.2 (3);
- NBN EN 1995-1-1: 5.2 (4);
- NBN EN 1995-1-1: 5.4;

- NBN EN 1995-1-1: 6.1.2;
- NBN EN 1995-1-1: 6.1.3;
- NBN EN 1995-1-1: 6.1.4;
- NBN EN 1995-1-1: 6.1.5;
- NBN EN 1995-1-1: 6.1.7 (2);
- NBN EN 1995-1-1: 6.1.8;
- NBN EN 1995-1-1: 6.2;
- NBN EN 1995-1-1: 6.3.1 (1)P;
- NBN EN 1995-1-1: 6.3.3 (6);
- NBN EN 1995-1-1: 6.3.4;
- NBN EN 1995-1-1: 6.5;
- NBN EN 1995-1-1: 6.6;
- NBN EN 1995-1-1: 7.3;
- NBN EN 1995-1-1: 9;
- NBN EN 1995-1-1: 10;
- NBN EN 1995-1-1: Bijlage B;
- NBN EN 1995-1-2;
- NBN EN 1995-2.

Bijlage G - Formulier berekeningen beton

Masterproof Sanen

FileEdit

Geometry

Loads

Material

Results

Calculations

Analys and control beam

Optimization Functions

Concrete

Steel

Timber

Generate calculations

Calculations

Concrete Calculations:

Section Details:

Partial Factor for Concrete, $\gamma_c = 1.5$

Partial Factor for Steel $\gamma_s = 1.15$

Longterm factor for compressive strength concrete for Bending $\alpha_{cc} = 0.85$

Longterm factor for compressive strength for Shear $\alpha_{cc} = 1.0$

Cover to Reinforcement $c = 25\text{ mm}$

Creep Factor $\varphi_0 = 2.15$

Exposure ClassXD1

Height of Section $h = 500\text{ mm}$

Width of Section $b = 250\text{ mm}$

Strength ClassC25/30

Characteristic Strength of Concrete $f_{ck} = 25\text{ N/mm}^2$

Mean Strength of Concrete $f_{cm} = 33\text{ N/mm}^2$

Mean Tensile Strength of Concrete $f_{ctm} = 2.6\text{ N/mm}^2$

Mean Modulus of Elasticity of Concrete $E_{cm} = 31000\text{ N/mm}^2$

Strength Class of ReinforcementBE500S

Characteristic Strength of Reinforcement $f_{yk} = 500\text{ N/mm}^2$

Modulus of Elasticity of Reinforcement $E_s = 200000\text{ N/mm}^2$

Modulus of Elasticity of Concrete without creep $E_c = 31000\text{ N/mm}^2$

Alpha factor without creep $\alpha = \frac{E_c}{E_c} = \frac{200000\text{ N/mm}^2}{31000\text{ N/mm}^2} = 6.45$

Modulus of Elasticity of Concrete with creep $E_{c,creep} = \frac{E_c}{1+\varphi_0} = \frac{31000\text{ N/mm}^2}{1+2.15} = 9841\text{ N/mm}^2$

Alpha factor with creep $\alpha_{creep} = \frac{E_c}{E_{c,creep}} = \frac{200000\text{ N/mm}^2}{9841\text{ N/mm}^2} = 20.32$

Coefficient for stress distribution $k_c = 0.4$

Factor for load duration $k_t = 0.4\text{ (longterm)}$

Factor for adhesion reinforcement $k_1 = 0.8$

Factor for strain distribution $k_2 = 0.5\text{ (bending)}$

Factor according to NBN EN 1992-1-1 $k_3 = 3.4$

Factor according to NBN EN 1992-1-1 $k_4 = 0.425$

Factor for effect load duration on strain $\beta = 0.5\text{ (longterm)}$

Angle of compression strut $\theta = 26.6^\circ$

Maximum deformation factor for SLS-C	$\delta_{max,SLS-C} = \frac{f_{yk}}{300}$
Maximum deformation factor for SLS-F	$\delta_{max,SLS-F} = \frac{f_{yk}}{300}$
Maximum deformation factor for SLS-Q	$\delta_{max,SLS-Q} = \frac{f_{yk}}{300}$
Maximum deformation factor with creep	$\delta_{max,creep} = \frac{f_{yk}}{250}$
Maximum concrete stress for SLS-C	$\sigma_{c,max,SLS-C} = 0.60 \times f_{ck} = 0.60 \times 25 \text{ N/mm}^2 = 15 \text{ N/mm}^2$
Maximum concrete stress for SLS-F	$\sigma_{c,max,SLS-F} = 0.50 \times f_{ck} = 0.50 \times 25 \text{ N/mm}^2 = 12.5 \text{ N/mm}^2$
Maximum concrete stress for SLS-Q	$\sigma_{c,max,SLS-Q} = 0.45 \times f_{ck} = 0.45 \times 25 \text{ N/mm}^2 = 11.25 \text{ N/mm}^2$
Maximum steel stress	$\sigma_{s,max} = 0.80 \times f_{yk} = 0.80 \times 500 \text{ N/mm}^2 = 400 \text{ N/mm}^2$
Maximum crack width	$w_{max} = 0.4 \text{ mm}$
<u>Determine Minimum Longitudinal Reinforcement:</u>	
Greatst value of effective depth at 0 m	$d = 457 \text{ mm}$
Absolute minimum reinforcement	$A_{s,min,abs} = 0.0013 \times b \times d = 0.0013 \times 250 \text{ mm} \times 457 \text{ mm} = 149 \text{ mm}^2$
Minimum reinforcement against brittle fracture	$A_{s,min,brittle} = 0.26 \times \frac{f_{ctm}}{f_{yk}} \times b \times d = 0.26 \times \frac{2.6 \text{ N/mm}^2}{500 \text{ N/mm}^2} \times 250 \text{ mm} \times 457 \text{ mm} = 154 \text{ mm}^2$
Area of concrete in tension	$A_{ct} = b \times \frac{h}{2} = 250 \text{ mm} \times \frac{500 \text{ mm}}{2} = 62500 \text{ mm}^2$
Factor for height	$k = 1.21 - 0.7 \times \frac{h}{1000} = 1.21 - 0.7 \times \frac{500}{1000} = 0.86$
Minimum reinforcement for crack control	$A_{s,min,crackcontrol} = \frac{k \times k \times f_{ctm} \times A_{ct}}{f_{yk}} = \frac{0.4 \times 0.86 \times 2.6 \text{ N/mm}^2 \times 62500 \text{ mm}^2}{500 \text{ N/mm}^2} = 112 \text{ mm}^2$
Minimum reinforcement	$A_{s,min} = \max(A_{s,min,abs}; A_{s,min,brittle}; A_{s,min,crackcontrol}) = \max(149 \text{ mm}^2; 154 \text{ mm}^2; 112 \text{ mm}^2) = 154 \text{ mm}^2$
<u>Control of Longitudinal Reinforcement:</u>	
Design value of moment at 2.5 m	$M_{Ed} = 106.93 \text{ kNm}$
Effective depth at 2.5 m	$d = 457 \text{ mm}$
Provided reinforcement at 2.5 m	$A_{s,prov} = 628 \text{ mm}^2$
Design value of concrete compressive strength	$f_{cd} = \alpha_{cc} \times \frac{f_{ck}}{\gamma_c} = 0.85 \times \frac{25 \text{ N/mm}^2}{1.5} = 14.17 \text{ N/mm}^2$
Design value of reduced bending moment	$\mu_d = \frac{M_{Ed}}{b \times d^2 \times f_{cd}} = \frac{106.93 \times 10^6 \text{ Nmm}}{250 \text{ mm} \times (457 \text{ mm})^2 \times 14.17 \text{ N/mm}^2} = 0.1446$
Ratio lever arm to effective depth	$\zeta = 0.9167$ (Interpolation in Table 4.2.3.7a)
Design value of stress in reinforcement	$\sigma_s = 435 \text{ N/mm}^2$ (Interpolation in Table 4.2.3.7a)
Calculated reinforcement	$A_{s,calculated} = \frac{M_{Ed}}{\zeta \times d \times \sigma_s} = \frac{106.93 \times 10^6 \text{ Nmm}}{0.9167 \times 457 \text{ mm} \times 435 \text{ N/mm}^2} = 586 \text{ mm}^2$
Required reinforcement	$A_{s,req} = \max(A_{s,min}; A_{s,calculated}) = \max(154 \text{ mm}^2; 586 \text{ mm}^2) = 586 \text{ mm}^2$
Check if provided reinforcement is sufficient	$A_{s,prov} \geq A_{s,req} = 628 \text{ mm}^2 \geq 586 \text{ mm}^2 \Rightarrow \text{OK}$
<u>Determine Minimum Shear Reinforcement:</u>	
Minimum shear reinforcement	$\left(\frac{A_{sk}}{s}\right)_{min} = \frac{0.08 \times \sqrt{f_{ck}} \times b}{f_{yk}} = \frac{0.08 \times \sqrt{25 \text{ N/mm}^2} \times 250 \text{ mm}}{500 \text{ N/mm}^2} = 0.200 \text{ mm}^2/\text{mm}$
<u>Control of Shear Reinforcement:</u>	
Design value of shear force at 0.5 m	$V_{Ed} = 66.25 \text{ kN}$
Effective depth at 0.5 m	$d = 457 \text{ mm}$
Provided shear reinforcement at 0.5 m	$\left(\frac{A_{sk}}{s}\right)_{prov} = 0.335 \text{ mm}^2/\text{mm}$
Design value of tensile strength of shear reinforcement	$f_{yk,d} = \frac{f_{yk}}{\gamma_s} = \frac{500 \text{ N/mm}^2}{1.15} = 435 \text{ N/mm}^2$
Calculated shear reinforcement	$\left(\frac{A_{sk}}{s}\right)_{calculated} = \frac{V_{Ed}}{0.9 \times d \times f_{yk,d} \times \cot(\theta)} = \frac{66.25 \times 10^3 \text{ N}}{0.9 \times 457 \text{ mm} \times 435 \text{ N/mm}^2 \times \cot(26.6^\circ)} = 0.185 \text{ mm}^2/\text{mm}$
Required shear reinforcement	$\left(\frac{A_{sk}}{s}\right)_{req} = \max\left(\left(\frac{A_{sk}}{s}\right)_{min}; \left(\frac{A_{sk}}{s}\right)_{calculated}\right) = \max(0.200 \text{ mm}^2/\text{mm}; 0.185 \text{ mm}^2/\text{mm}) = 0.200 \text{ mm}^2/\text{mm}$
Check if provided shear reinforcement is sufficient	$\left(\frac{A_{sk}}{s}\right)_{prov} \geq \left(\frac{A_{sk}}{s}\right)_{req} = 0.335 \text{ mm}^2/\text{mm} \geq 0.200 \text{ mm}^2/\text{mm} \Rightarrow \text{OK}$
<u>Section Parameters without Creep:</u>	
Effective depth at 2.5 m	$d = 457 \text{ mm}$
Distance to center of gravity secondary reinforcement at 2.5 m	$d_2 = 28 \text{ mm}$
Amount of primary reinforcement at 2.5 m	$A_{s1} = 628 \text{ mm}^2$
Amount of secondary reinforcement at 2.5 m	$A_{s2} = 226 \text{ mm}^2$
Height of compression zone in uncracked section at 2.5 m	$x_{uncracked} = \frac{\frac{b \times d^2}{4} + \alpha \times (A_{s1} \times d + A_{s2} \times d_2)}{A_{s1} \times \alpha + A_{s2} \times \alpha + b \times h}$
	$x_{uncracked} = \frac{\frac{250 \text{ mm} \times (500 \text{ mm})^2}{4} + 6.45 \times (628 \text{ mm}^2 \times 457 \text{ mm} + 226 \text{ mm}^2 \times 28 \text{ mm})}{628 \text{ mm}^2 \times 6.45 + 226 \text{ mm}^2 \times 6.45 + 250 \text{ mm} \times 500 \text{ mm}} = 254 \text{ mm}$
Moment of inertia of uncracked section at 2.5 m	$I_{uncracked} = \frac{b \times x^3}{3} + \frac{b \times (h - x_{uncracked})^3}{3} + \alpha \times A_{s1} \times (d - x_{uncracked})^2 + \alpha \times A_{s2} \times (x_{uncracked} - d_2)^2$
	$I_{uncracked} = \frac{250 \text{ mm} \times (254 \text{ mm})^3}{3} + \frac{250 \text{ mm} \times (500 \text{ mm} - 254 \text{ mm})^3}{3} + 6.45 \times 628 \text{ mm}^2 \times (457 \text{ mm} - 254 \text{ mm})^2 + 6.45 \times 226 \text{ mm}^2 \times (254 \text{ mm} - 28 \text{ mm})^2$
	$I_{uncracked} = 2.85 \times 10^9 \text{ mm}^4$
Critical moment at 2.5 m	$M_{cr} = \frac{I_{uncracked} \times f_{ctm}}{h - x_{uncracked}} = \frac{2.85 \times 10^9 \text{ mm}^4 \times 2.6 \text{ N/mm}^2}{500 \text{ mm} - 254 \text{ mm}} = 30 \text{ kNm}$
Height of compression zone in cracked section at 2.5 m	$x_{cracked} = \frac{-\alpha \times (A_{s1} + A_{s2})}{b} + \sqrt{\left(\frac{\alpha \times (A_{s1} + A_{s2})}{b}\right)^2 + \frac{2 \times \alpha \times (A_{s1} \times d + A_{s2} \times d_2)}{b}}$
	$x_{cracked} = \frac{-6.45 \times (628 \text{ mm}^2 + 226 \text{ mm}^2)}{250 \text{ mm}} + \sqrt{\left(\frac{6.45 \times (628 \text{ mm}^2 + 226 \text{ mm}^2)}{250 \text{ mm}}\right)^2 + \frac{2 \times 6.45 \times (628 \text{ mm}^2 \times 457 \text{ mm} + 226 \text{ mm}^2 \times 28 \text{ mm})}{250 \text{ mm}}} = 103 \text{ mm}$

Moment of inertia of cracked section at 2.5 m	$x_{cracked} = \frac{-6.45 \times (628 \text{ mm}^2 + 226 \text{ mm}^2)}{250 \text{ mm}} + \sqrt{\left(\frac{6.45 \times (628 \text{ mm}^2 + 226 \text{ mm}^2)}{250 \text{ mm}}\right)^2 + \frac{2 \times 6.45 \times (628 \text{ mm}^2 \times 457 \text{ mm} + 226 \text{ mm}^2 \times 28 \text{ mm})}{250 \text{ mm}}} = 103 \text{ mm}$ $I_{cracked} = \frac{b \times x_{cracked}^3}{3} + \alpha \times A_{s1} \times (d - x_{cracked})^2 + \alpha \times A_{s2} \times (x_{cracked} - d_2)^2$ $I_{cracked} = \frac{250 \text{ mm} \times (35 \text{ mm})^3}{3} + 6.45 \times 628 \text{ mm}^2 \times (457 \text{ mm} - 35 \text{ mm})^2 + 6.45 \times 226 \text{ mm}^2 \times (35 \text{ mm} - 28 \text{ mm})^2 = 6.07 \times 10^8 \text{ mm}^4$
<u>Control of Maximum Concrete Stress for SLS-C:</u>	
Design value of moment at 2.5 m	$M_{Ed} = 72.27 \text{ kNm}$
Check if section is cracked	$M_{Ed} > M_{cr} = 72.27 \text{ kNm} > 30 \text{ kNm} \Rightarrow \text{CRACKED}$
Concrete stress	$\sigma_c = \frac{M_{Ed} \times x_{cracked}}{I_{cracked}} = \frac{72.27 \times 10^6 \text{ Nmm} \times 103 \text{ mm}}{6.07 \times 10^8 \text{ mm}^4} = 12.3 \text{ N/mm}^2$
Stress in primary reinforcement	$\sigma_{s1} = \frac{d - x_{cracked}}{x_{cracked}} \times \sigma_c \times \alpha = \frac{457 \text{ mm} - 103 \text{ mm}}{103 \text{ mm}} \times 12.3 \text{ N/mm}^2 \times 6.45 = 273 \text{ N/mm}^2$
Stress in secondary reinforcement	$\sigma_{s2} = \frac{x_{cracked} - d_2}{x_{cracked}} \times \sigma_c \times \alpha = \frac{103 \text{ mm} - 28 \text{ mm}}{103 \text{ mm}} \times 12.3 \text{ N/mm}^2 \times 6.45 = 49 \text{ N/mm}^2$
Check concrete stress	$\sigma_c \leq \sigma_{c,max,SLS-C} = 12.3 \text{ N/mm}^2 \leq 15 \text{ N/mm}^2 \Rightarrow \text{OK}$
Check stress in primary reinforcement	$\sigma_{s1} \leq \sigma_{s,max} = 273 \text{ N/mm}^2 \leq 400 \text{ N/mm}^2 \Rightarrow \text{OK}$
Check stress in secondary reinforcement	$\sigma_{s2} \leq \sigma_{s,max} = 49 \text{ N/mm}^2 \leq 400 \text{ N/mm}^2 \Rightarrow \text{OK}$
<u>Control of Maximum Concrete Stress for SLS-F:</u>	
Design value of moment at 2.5 m	$M_{Ed} = 53.52 \text{ kNm}$
Check if section is cracked	$M_{Ed} > M_{cr} = 53.52 \text{ kNm} > 30 \text{ kNm} \Rightarrow \text{CRACKED}$
Concrete stress	$\sigma_c = \frac{M_{Ed} \times x_{cracked}}{I_{cracked}} = \frac{53.52 \times 10^6 \text{ Nmm} \times 103 \text{ mm}}{6.07 \times 10^8 \text{ mm}^4} = 9.1 \text{ N/mm}^2$
Stress in primary reinforcement	$\sigma_{s1} = \frac{d - x_{cracked}}{x_{cracked}} \times \sigma_c \times \alpha = \frac{457 \text{ mm} - 103 \text{ mm}}{103 \text{ mm}} \times 9.1 \text{ N/mm}^2 \times 6.45 = 202 \text{ N/mm}^2$
Stress in secondary reinforcement	$\sigma_{s2} = \frac{x_{cracked} - d_2}{x_{cracked}} \times \sigma_c \times \alpha = \frac{103 \text{ mm} - 28 \text{ mm}}{103 \text{ mm}} \times 9.1 \text{ N/mm}^2 \times 6.45 = 36 \text{ N/mm}^2$
Check concrete stress	$\sigma_c \leq \sigma_{c,max,SLS-F} = 9.1 \text{ N/mm}^2 \leq 12.5 \text{ N/mm}^2 \Rightarrow \text{OK}$
Check stress in primary reinforcement	$\sigma_{s1} \leq \sigma_{s,max} = 202 \text{ N/mm}^2 \leq 400 \text{ N/mm}^2 \Rightarrow \text{OK}$
Check stress in secondary reinforcement	$\sigma_{s2} \leq \sigma_{s,max} = 36 \text{ N/mm}^2 \leq 400 \text{ N/mm}^2 \Rightarrow \text{OK}$
<u>Section Parameters with Creep:</u>	
Height of compression zone in uncracked section at 2.5 m	$x_{uncracked,creep} = \frac{\frac{b \times h^2}{2} + \alpha_{creep} \times (A_{s1} \times d + A_{s2} \times d_2)}{A_{s1} \times \alpha_{creep} + A_{s2} \times \alpha_{creep} + b \times h}$ $x_{uncracked,creep} = \frac{\frac{250 \text{ mm} \times (500 \text{ mm})^2}{2} + 20.32 \times (628 \text{ mm}^2 \times 457 \text{ mm} + 226 \text{ mm}^2 \times 28 \text{ mm})}{628 \text{ mm}^2 \times 20.32 + 226 \text{ mm}^2 \times 20.32 + 250 \text{ mm} \times 500 \text{ mm}} = 261 \text{ mm}$
Moment of inertia of uncracked, creep section at 2.5 m	$I_{uncracked,creep} = \frac{b \times x_{uncracked,creep}^3}{3} + \frac{b \times (h - x_{uncracked,creep})^3}{3} + \alpha_{creep} \times A_{s1} \times (d - x_{uncracked,creep})^2 + \alpha_{creep} \times A_{s2} \times (x_{uncracked,creep} - d_2)^2$ $I_{uncracked,creep} = \frac{250 \text{ mm} \times (261 \text{ mm})^3}{3} + \frac{250 \text{ mm} \times (500 \text{ mm} - 261 \text{ mm})^3}{3} + 20.32 \times 628 \text{ mm}^2 \times (457 \text{ mm} - 261 \text{ mm})^2 + 20.32 \times 226 \text{ mm}^2 \times (261 \text{ mm} - 28 \text{ mm})^2$ $I_{uncracked,creep} = 3.34 \times 10^9 \text{ mm}^4$
Critical moment at 2.5 m	$M_{cr} = \frac{I_{uncracked,creep} \times f_{ctm}}{h - x_{uncracked,creep}} = \frac{3.34 \times 10^9 \text{ mm}^4 \times 2.6 \text{ N/mm}^2}{500 \text{ mm} - 261 \text{ mm}} = 36 \text{ kNm}$
Height of compression zone in cracked section at 2.5 m	$x_{cracked,creep} = \frac{-\alpha_{creep} \times (A_{s1} + A_{s2})}{b} + \sqrt{\left(\frac{\alpha_{creep} \times (A_{s1} + A_{s2})}{b}\right)^2 + \frac{2 \times \alpha_{creep} \times (A_{s1} \times d + A_{s2} \times d_2)}{b}}$ $x_{cracked,creep} = \frac{-20.32 \times (628 \text{ mm}^2 + 226 \text{ mm}^2)}{250 \text{ mm}} + \sqrt{\left(\frac{20.32 \times (628 \text{ mm}^2 + 226 \text{ mm}^2)}{250 \text{ mm}}\right)^2 + \frac{2 \times 20.32 \times (628 \text{ mm}^2 \times 457 \text{ mm} + 226 \text{ mm}^2 \times 28 \text{ mm})}{250 \text{ mm}}} = 160 \text{ mm}$
Moment of inertia of cracked, creep section at 2.5 m	$I_{cracked,creep} = \frac{b \times x_{cracked,creep}^3}{3} + \alpha_{creep} \times A_{s1} \times (d - x_{cracked,creep})^2 + \alpha_{creep} \times A_{s2} \times (x_{cracked,creep} - d_2)^2$ $I_{cracked,creep} = \frac{250 \text{ mm} \times (160 \text{ mm})^3}{3} + 20.32 \times 628 \text{ mm}^2 \times (457 \text{ mm} - 160 \text{ mm})^2 + 20.32 \times 226 \text{ mm}^2 \times (160 \text{ mm} - 28 \text{ mm})^2 = 1.53 \times 10^9 \text{ mm}^4$
<u>Control of Maximum Concrete Stress for SLS-Q:</u>	
Design value of moment at 2.5 m	$M_{Ed} = 47.27 \text{ kNm}$
Check if section is cracked	$M_{Ed} > M_{cr} = 47.27 \text{ kNm} > 36 \text{ kNm} \Rightarrow \text{CRACKED}$
Concrete stress	$\sigma_c = \frac{M_{Ed} \times (x_{cracked,creep})}{I_{cracked,creep}} = \frac{47.27 \times 10^6 \text{ Nmm} \times 160 \text{ mm}}{1.53 \times 10^9 \text{ mm}^4} = 4.9 \text{ N/mm}^2$
Stress in primary reinforcement	$\sigma_{s1} = \frac{d - x_{cracked,creep}}{x_{cracked,creep}} \times \sigma_c \times \alpha_{creep} = \frac{457 \text{ mm} - 160 \text{ mm}}{160 \text{ mm}} \times 4.9 \text{ N/mm}^2 \times 20.32 = 186 \text{ N/mm}^2$

Stress in secondary reinforcement	$\sigma_{s2} = \frac{x_{cracked,creep}-d_2}{x_{cracked,creep}} \times \sigma_c \times \alpha_{creep} = \frac{160\text{ mm}-28\text{ mm}}{160\text{ mm}} \times 4.9\text{ N/mm}^2 \times 20.32 = 76\text{ N/mm}^2$
Check concrete stress	$\sigma_c \leq \sigma_{c,max,SLS-Q} = 4.9\text{ N/mm}^2 \leq 11.25\text{ N/mm}^2 \Rightarrow \text{OK}$
Check stress in primary reinforcement	$\sigma_{s1} \leq \sigma_{s,max} = 186\text{ N/mm}^2 \leq 400\text{ N/mm}^2 \Rightarrow \text{OK}$
Check stress in secondary reinforcement	$\sigma_{s2} \leq \sigma_{s,max} = 76\text{ N/mm}^2 \leq 400\text{ N/mm}^2 \Rightarrow \text{OK}$
<u>Control of Maximum Crack Width:</u>	
Effective area of concrete at 2.5 m	$A_{c,eff} = b \times \min\left(\frac{h-x_{cracked,creep}}{3}; 2.5 \times (h-d); \frac{h}{3}\right) = 250\text{ mm} \times \min\left(\frac{500\text{ mm}-160\text{ mm}}{3}; 2.5 \times (500\text{ mm} - 457\text{ mm}); \frac{500\text{ mm}}{2}\right) = 250\text{ mm} \times 107.5\text{ mm}$
Reinforcement ratio	$\rho_{eff} = \frac{A_{st}}{A_{c,eff}} = \frac{628\text{ mm}^2}{26875\text{ mm}^2} = 0.0234$
Equivalent diameter of reinforcement	$\phi_{eq} = 20\text{ mm}$
Maximum crack width	$S_{r,max} = k_3 \times c + k_1 \times k_2 \times k_4 \times \frac{\sigma_{st}}{\rho_{eff}} = 3.4 \times 25 + 0.8 \times 0.5 \times 0.425 \times \frac{20\text{ mm}}{0.0234} = 230\text{ mm}$
Minimum difference in strain between reinforcement and concrete	$(\varepsilon_{sm} - \varepsilon_{cm})_{min} = 0.6 \times \frac{\sigma_s}{E_s} = 0.6 \times \frac{186\text{ N/mm}^2}{200000\text{ N/mm}^2} = 5.58 \times 10^{-4}$
Calculated difference in strain between reinforcement and concrete	$(\varepsilon_{sm} - \varepsilon_{cm})_{calculated} = \frac{\sigma_s - k_1 \times \frac{k_{creep}}{\rho_{eff}} \times (1 - \alpha_{creep} \times \rho_{eff})}{E_s} = \frac{186\text{ N/mm}^2 - 1.0 \times \frac{2.5\text{ N/mm}^2}{0.0234} \times (1 - 20.32 \times 0.0234)}{200000\text{ N/mm}^2} = 6.01 \times 10^{-4}$
Difference in strain between reinforcement and concrete	$(\varepsilon_{sm} - \varepsilon_{cm}) = \max((\varepsilon_{sm} - \varepsilon_{cm})_{min}; (\varepsilon_{sm} - \varepsilon_{cm})_{calculated}) = \max(5.58 \times 10^{-4}; 6.01 \times 10^{-4}) = 6.01 \times 10^{-4}$
Crack width	$w_k = S_{r,max} \times (\varepsilon_{sm} - \varepsilon_{cm}) = 230\text{ mm} \times 6.01 \times 10^{-4} = 0.138\text{ mm}$
Check crack width	$w_k \leq w_{max} = 0.138\text{ mm} \leq 0.4\text{ mm} \Rightarrow \text{OK}$
<u>Control of Maximum Deflection with Creep:</u>	
Design value of moment at 2.5 m	$M_{Ed} = 47.27\text{ kNm}$
Deflection of uncracked section at 2.5 m	$\alpha_I = 1.19\text{ mm}$
Deflection of cracked section at 2.5 m	$\alpha_{II} = 2.59\text{ mm}$
Distribution factor	$\zeta = 1 - \beta \times \left(\frac{M_{cr}}{M_{Ed}}\right)^2 = 1 - 0.5 \times \left(\frac{36\text{ kNm}}{47.27\text{ kNm}}\right)^2 = 0.71$
Deflection with creep	$\alpha = \zeta \times \alpha_{II} + (1 - \zeta) \times \alpha_I = 0.71 \times 2.59\text{ mm} + (1 - 0.71) \times 1.19\text{ mm} = 2.18\text{ mm}$
Check deflection with creep	$\alpha \leq \frac{L}{\delta_{max,creep}} = 2.18\text{ mm} \leq \frac{5000\text{ mm}}{250} = 2.18\text{ mm} \leq 20\text{ mm} \Rightarrow \text{OK}$

Bijlage H- Formulier berekeningen staal

Steel Calculations:	
<u>Section Details:</u>	
Partial factor for resistance strength	$\gamma_{M0} = 1.0$
Partial factor for resistance stability	$\gamma_{M1} = 1.0$
Elasticity modulus	$E = 210000 \text{ MPa}$
Shear modulus	$G = 81000 \text{ MPa}$
Section name	IPE 200
Height	$h = 200 \text{ mm}$
Width	$b = 100 \text{ mm}$
Flange thickness	$t_f = 8.5 \text{ mm}$
Web thickness	$t_w = 5.6 \text{ mm}$
Radius	$r = 12 \text{ mm}$
Plastic section modulus	$W_{pl,y} = 220600 \text{ mm}^3$
Area of effective shear plane	$A_v = 1400 \text{ mm}^2$
Moment of inertia about y-axis	$I_y = 19430000 \text{ mm}^4$
Moment of inertia about z-axis	$I_z = 1423000 \text{ mm}^4$
Torsion constant	$I_t = 69150 \text{ mm}^4$
Warping constant	$I_w = 12980000000 \text{ mm}^6$
Strength class	$f_y = 275 \text{ N/mm}^2$
Maximum deflection factor at SLS-C	$\delta_{max,SLS-C} = \frac{f_y}{300}$
Maximum deflection factor at SLS-F	$\delta_{max,SLS-F} = \frac{f_y}{400}$
Maximum deflection factor at SLS-Q	$\delta_{max,SLS-Q} = \frac{f_y}{500}$
<u>Section Classification:</u>	
Strength class factor	$\epsilon = \sqrt{\frac{235}{f_y}} = \sqrt{\frac{235}{275}} = 0.92$
Web classification	$\frac{c}{t_w} = \frac{h-2 \times t_f-2 \times r}{t_w} = \frac{200-2 \times 8.5-2 \times 12}{5.6} = 28.39 \rightarrow (28.39 \leq 72\epsilon = \text{class } 1)$
Flange classification	$\frac{c}{t_f} = \frac{b-t_w-r}{t_f} = \frac{100-5.6-12}{8.5} = 4.14 \rightarrow (4.14 \leq 9\epsilon = \text{class } 1)$
Section classification	$\text{class} = \max(1, 1) = 1$
<u>Control of Bending in ULS-P:</u>	
Design value of applied moment at 3.0 m	$M_{Ed} = -79.76 \text{ kNm}$
Design value of resistance moment	$M_{c,Rd} = \frac{W_{pl,y} \times f_y}{\gamma_{M0}} = \frac{220600 \text{ mm}^3 \times 275 \text{ N/mm}^2}{1.0} = 60.66 \text{ kNm}$
Check of bending resistance	$\frac{M_{Ed}}{M_{c,Rd}} = \frac{-79.76 \text{ kNm}}{60.66 \text{ kNm}} = 1.31 \leq 1 \rightarrow \text{NOT OK}$
<u>Control of Shear in ULS-P:</u>	
Design value of applied shear force at 3.0 m	$V_{Ed} = 105.85 \text{ kN}$
Design value of resistance shear force	$V_{c,Rd} = \frac{A_{v,z} \times f_{t,k}}{\sqrt{3} \times \gamma_{M0}} = \frac{1400 \text{ mm}^2 \times 275 \text{ N/mm}^2}{\sqrt{3} \times 1.0} = 222.29 \text{ kN}$
Check of shear resistance	$\frac{V_{Ed}}{V_{c,Rd}} = \frac{105.85 \text{ kN}}{222.29 \text{ kN}} = 0.48 \leq 1 \rightarrow \text{OK}$
<u>Control of combination Shear and Bending in ULS-P:</u>	
Design value of applied shear force at 3.0 m	$V_{Ed} = 105.69 \text{ kN}$
Check of combination is necessary	$V_{Ed} \leq 0.5 \times V_{c,Rd} = 105.69 \text{ kN} \leq 0.5 \times 222.28 \text{ kN} \rightarrow \text{Control not necessary}$
Reduction factor for yield strength	Control not necessary
Reduced yield strength	Control not necessary
Design value of applied moment at 3.0 m	Control not necessary
Design value of resistance moment	Control not necessary
Check of bending resistance	Control not necessary

Control of Stability in ULS-P:

Interval between 2.00 m and 3.00 m

Critical moment for lateral-torsional buckling

$$L = 1.00 \text{ m}$$

$$M_{cr} = C_1 \times \frac{\pi^2 \times E \times I_z}{L^2} \times \left(\sqrt{\frac{I_y}{I_z} + \frac{I_z^2 \times G \times I_x}{\pi^4 \times E \times I_z^2} + (C_2 \times z_g)^2} - C_2 \times z_g \right)$$

$$M_{cr} = 1.127 \times \frac{\pi^2 \times 210000 \times 1423000}{1.00^2} \times \left(\sqrt{\frac{1298000000}{1423000} + \frac{1.00^2 \times 81000 \times 69150}{\pi^4 \times 210000 \times 1423000} + (0.454 \times -100.0)^2} - 0.454 \times -100.0 \right)$$

$$M_{cr} = 531.00 \text{ kNm}$$

Slenderness factor

$$\lambda_{LT} = \sqrt{\frac{W_{pl,y} \times L}{M_{cr}}} = \sqrt{\frac{220600 \text{ mm}^3 \times 275 \text{ N/mm}^2}{531.00 \text{ kNm}}} = 0.34$$

Imperfection factor

$$h/b \leq 2 \rightarrow \alpha_{LT} = 0.21$$

Factor Φ

$$\Phi_{LT} = 0.5 \times (1 + \alpha_{LT} \times (\lambda_{LT} - 0.2) + \lambda_{LT}^2) = 0.5 \times (1 + 0.21 \times (0.34 - 0.2) + 0.34^2) = 0.57$$

Reduction factor

$$\chi_{LT} = \frac{1}{\Phi_{LT} + \sqrt{\Phi_{LT}^2 - \lambda_{LT}^2}} = \frac{1}{0.57 + \sqrt{0.57^2 - 0.34^2}} = 0.97$$

Design lateral buckling resistance

$$M_{b,Rd} = \chi_{LT} \times \frac{f_y \times W_{pl,y}}{\gamma_{M1}} = 0.97 \times \frac{275 \text{ N/mm}^2 \times 220600 \text{ mm}^3}{1.0} = 58.75 \text{ kNm}$$

Check of lateral buckling resistance

$$\frac{M_{Ed}}{M_{b,Rd}} = \frac{79.62 \text{ kNm}}{58.75 \text{ kNm}} = 1.36 \leq 1 \rightarrow \text{NOT OK}$$

Control of Deflection in SLS-C:

Deflection at 5.5 m

$$\delta_{SLS-C} = 15.67 \text{ mm}$$

Maximum deflection

$$\delta_{max,SLS-C} = \frac{L}{300} = \frac{5000.0 \text{ mm}}{300} = 16.67 \text{ mm}$$

Check of deflection

$$\delta_{SLS-C} \leq \delta_{max,SLS-C} = 15.67 \text{ mm} \leq 16.67 \text{ mm} \rightarrow \text{OK}$$

Control of Deflection in SLS-F:

Deflection at 5.5 m

$$\delta_{SLS-F} = 7.91 \text{ mm}$$

Maximum deflection

$$\delta_{max,SLS-F} = \frac{L}{400} = \frac{5000.0 \text{ mm}}{400} = 12.50 \text{ mm}$$

Check of deflection

$$\delta_{SLS-F} \leq \delta_{max,SLS-F} = 7.91 \text{ mm} \leq 12.50 \text{ mm} \rightarrow \text{OK}$$

Control of Deflection in SLS-Q:

Deflection at 5.5 m

$$\delta_{SLS-Q} = 4.80 \text{ mm}$$

Maximum deflection

$$\delta_{max,SLS-Q} = \frac{L}{500} = \frac{5000.0 \text{ mm}}{500} = 10.00 \text{ mm}$$

Check of deflection

$$\delta_{SLS-Q} \leq \delta_{max,SLS-Q} = 4.80 \text{ mm} \leq 10.00 \text{ mm} \rightarrow \text{OK}$$

Bijlage I- Formulier berekeningen hout

Masterproef Sanen

File Edit

Geometry

Loads

Material

Results

Calculations

Analyse and control beam

Optimization Functions

Concrete

Steel

Timber

Generate calculations

Calculations

Timber Calculations:

Section Details:

Partial factor for material

$\gamma_M = 1.3$

Mean elasticity modulus

$E_{mean} = 7000 \text{ N/mm}^2$

5th percentile elasticity modulus

$E_{0.05} = 4700 \text{ N/mm}^2$

5th percentile shear modulus

$G_{0.05} = 295 \text{ N/mm}^2$

Height of the section

$h = 400 \text{ mm}$

Width of the section

$b = 200 \text{ mm}$

Material type

C14

Characteristic bending strength

$f_{m,k} = 14 \text{ N/mm}^2$

Characteristic shear strength

$f_{v,k} = 3.0 \text{ N/mm}^2$

Service class

1

Load duration class

medium

Modification factor

$k_{mod} = 0.80$

Deflection factor

$k_{def} = 0.60$

Height factor

$k_h = 1.0$

Modification factor for redistribution

$k_m = 0.70$

Modification factor for influence of cracks on shear strength

$k_{cr} = 0.67$

Maximum deflection factor for SLS-C

$\delta_{max,SLS-C} = \frac{L}{300}$

Maximum deflection factor for SLS-F

$\delta_{max,SLS-F} = \frac{L}{300}$

Maximum deflection factor for SLS-Q

$\delta_{max,SLS-Q} = \frac{L}{300}$

Maximum deflection factor for deflection with creep

$\delta_{max,creep} = \frac{L}{300}$

Control Bending:

Design value of applied moment at 2.5 m

$M_{Ed} = 48.06 \text{ kNm}$

Section modulus

$W_y = \frac{b \times h^2}{6} = \frac{200 \times 400^2}{6} = 5333333 \text{ mm}^3$

Design bending stress

$\sigma_{m,y,d} = \frac{M_{Ed}}{W_y} = \frac{48.06 \times 10^6 \text{ Nmm}}{5333333 \text{ mm}^3} = 9.01 \text{ N/mm}^2$

Design bending strength

$f_{m,d} = k_h \times k_{mod} \times \frac{f_{m,k}}{\gamma_M} = 1.0 \times 0.80 \times \frac{14 \text{ N/mm}^2}{1.3} = 8.62 \text{ N/mm}^2$

Check for bending

$\frac{\sigma_{m,y,d}}{f_{m,d}} = \frac{9.01 \text{ N/mm}^2}{8.62 \text{ N/mm}^2} = 1.05 \leq 1 \rightarrow \text{NOT OK!}$

Control Shear:

Design value of applied shear force at 0 m

$$V_{Ed} = 38.45 \text{ kN}$$

Design shear stress

$$\tau_d = \frac{3 \times V_{Ed}}{2 \times k_{cs} \times b \times h} = \frac{3 \times 38.45 \times 1000 \text{ N}}{2 \times 0.67 \times 200 \text{ mm} \times 400 \text{ mm}} = 1.076 \text{ N/mm}^2$$

Design shear strength

$$f_{v,d} = k_h \times k_{mod} \times \frac{f_{vk}}{\gamma_M} = 1.0 \times 0.80 \times \frac{3.0 \text{ N/mm}^2}{1.3} = 1.85 \text{ N/mm}^2$$

Check for shear

$$\frac{\tau_d}{f_{v,d}} = \frac{1.076 \text{ N/mm}^2}{1.85 \text{ N/mm}^2} = 0.58 \leq 1 \rightarrow \text{OK!}$$

Control Stability:

Interval between 0 and 5 m

$$L = 5000 \text{ mm}$$

Effective length

$$L_e = L + 2 \times h = 5000 \text{ mm} + 2 \times 400 \text{ mm} = 5800 \text{ mm}$$

Moment of inertia about the z-axis

$$I_z = \frac{b \times h^3}{12} = \frac{400 \text{ mm} \times 200^3 \text{ mm}^3}{12} = 2.67 \times 10^8 \text{ mm}^4$$

Torsion constant

$$I_{tor} = \frac{b \times h^3}{3} = \frac{400 \text{ mm} \times 200^3 \text{ mm}^3}{3} = 1.07 \times 10^9 \text{ mm}^4$$

Design stress

$$\sigma_{m,crit} = \frac{\pi \times \sqrt{E_{0.05} \times I_z \times G_{0.05} \times I_{tor}}}{L_e \times W_y} = \frac{\pi \times \sqrt{4700 \text{ N/mm}^2 \times 2.67 \times 10^8 \text{ mm}^4 \times 295 \text{ N/mm}^2 \times 1.07 \times 10^9 \text{ mm}^4}}{5800 \text{ mm} \times 533333 \text{ mm}^3} = 63.91 \text{ N/mm}^2$$

Slenderness ratio

$$\lambda_{rel,y} = \sqrt{\frac{f_{m,k}}{\sigma_{m,crit}}} = \sqrt{\frac{14 \text{ N/mm}^2}{63.91 \text{ N/mm}^2}} = 0.468$$

Reduction factor

$$\lambda_{rel,y} < 0.75 \rightarrow k_{crit} = 1.0$$

Check for stability

$$\frac{\sigma_{m,d}}{k_{crit} \times f_{m,d}} = \frac{9.01 \text{ N/mm}^2}{1.0 \times 8.62 \text{ N/mm}^2} = 1.05 \leq 1 \rightarrow \text{NOT OK!}$$

Control Deflection SLS-C:

Deflection at 2.5 m

$$\delta_{SLS-C} = 11.2 \text{ mm}$$

Maximum deflection

$$\delta_{max,SLS-C} = \frac{L}{300} = \frac{5000 \text{ mm}}{300} = 16.67 \text{ mm}$$

Check for deflection SLS-C

$$\delta_{SLS-C} \leq \delta_{max,SLS-C} = 11.2 \text{ mm} \leq 16.67 \text{ mm} \rightarrow \text{OK!}$$

Control Deflection SLS-F:

Deflection at 2.5 m

$$\delta_{SLS-F} = 5.75 \text{ mm}$$

Maximum deflection

$$\delta_{max,SLS-F} = \frac{L}{300} = \frac{5000 \text{ mm}}{300} = 16.67 \text{ mm}$$

Check for deflection SLS-F

$$\delta_{SLS-F} \leq \delta_{max,SLS-F} = 5.75 \text{ mm} \leq 16.67 \text{ mm} \rightarrow \text{OK!}$$

Control Deflection SLS-Q:

Deflection at 2.5 m

$$\delta_{SLS-Q} = 3.57 \text{ mm}$$

Maximum deflection

$$\delta_{max,SLS-Q} = \frac{L}{300} = \frac{5000 \text{ mm}}{300} = 16.67 \text{ mm}$$

Check for deflection SLS-Q

$$\delta_{SLS-Q} \leq \delta_{max,SLS-Q} = 3.57 \text{ mm} \leq 16.67 \text{ mm} \rightarrow \text{OK!}$$

Control Deflection with Creep:

Deflection at 2.5 m

$$\delta_{creep} = 13.35 \text{ mm}$$

Maximum deflection

$$\delta_{max,creep} = \frac{L}{300} = \frac{5000 \text{ mm}}{300} = 16.67 \text{ mm}$$

Check for deflection with creep

$$\delta_{creep} \leq \delta_{max,creep} = 13.35 \text{ mm} \leq 16.67 \text{ mm} \rightarrow \text{OK!}$$