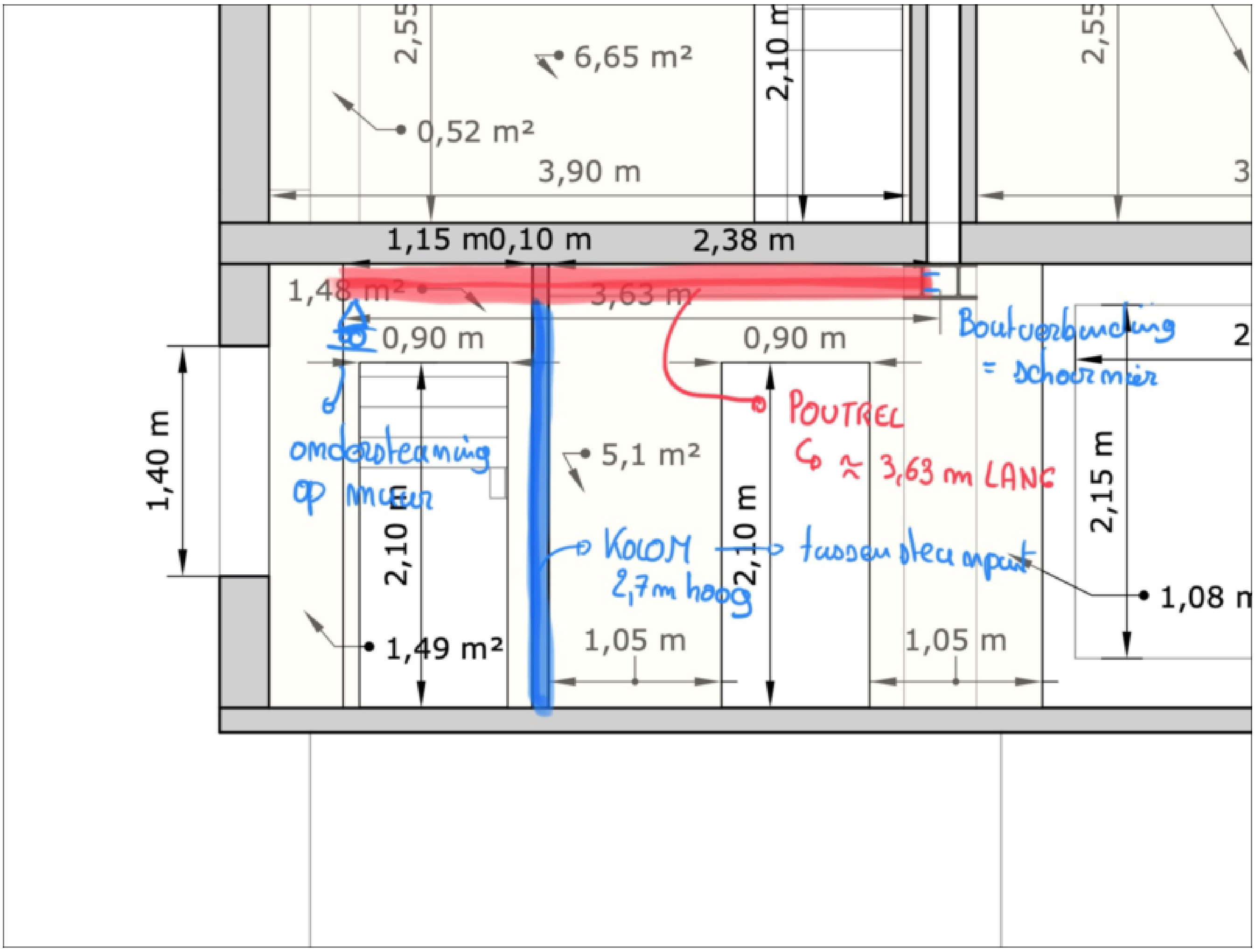
Berekening Kolom 2 - SHS 90/4

Berekening van **Kolom 2**, de kolom die de liggers **Profiel 2** ondersteunt en staat op de rand van de kelderwand. Geen krachten van het dak worden afgeleid naar deze kolom.

`situatieschets =`



Berekening Kolom 2

Conclussie

Een controle van de draagkracht van de kolom is uitgevoerd, dit zowel in **GGT** als **UGT**. Onderstaande controles zijn uitgevoerd.

Controles in **GGT**

$\longrightarrow$ Controle der vervormingen - zie de controles onder het hoofdstuk *Controle*

Controles in **UGT**

$\longrightarrow$ Doorsnede- en stabiliteitscontrole - zie de controles onder het hoofdstuk *Controle*

Algemeen (conservatief):

- **Elastische toetsing** aan de hand van het vloeicriterium van *Maxwell–Huber–Hencky–von Mises*, met $\sigma_{x,Ed}$ de spanning in de lengterichting, $\sigma_{z,Ed}$ de spanning in de dwarsrichting en τ_{Ed} de schuifspanning in een punt, maar **controle op basis van weerstanden** en interactie tussen N_{Rd}, V_{Rd} en M_{Rd} geniet voorkeur.

$$\left(\frac{\sigma_{x,Ed}}{f_y/\gamma_{M0}}\right)^2 + \left(\frac{\sigma_{z,Ed}}{f_y/\gamma_{M0}}\right)^2 - \left(\frac{\sigma_{x,Ed}}{f_y/\gamma_{M0}}\right)\left(\frac{\sigma_{z,Ed}}{f_y/\gamma_{M0}}\right) + 3\left(\frac{\tau_{Ed}}{f_y/\gamma_{M0}}\right) \leq 1$$

- **Conservatieve benadering** door het **lineair optellen** verhouding rekenwaarden belastingseffecten en hun weerstand kan ook.

$$\frac{N_{Ed}}{N_{Rd}} + \frac{M_{y,Ed}}{M_{y,Rd}} + \frac{M_{z,Ed}}{M_{z,Rd}} <= 1$$

Een **controle op basis van weerstanden** en met interactie tussen N_{Rd}, V_{Rd} en M_{Rd} is beschouwd in onderhavige rekennota.

Doorsnedecontrole `:UGT`

Axiale druk $\mathrm{UC} = \frac{N_{Ed}}{N_{c,Rd}} = \frac{96.6}{319.4} = 0.3 \rightarrow$ OK

Buigend moment $\mathrm{UC} = \frac{M_{Ed}}{M_{N,Rd}} = \frac{0.8}{9.3} = 0.09 \rightarrow$ OK

Dwarskracht $\mathrm{UC} = \frac{V_{Ed}}{V_{c,Rd}} = \frac{1.0}{159.6} = 0.01 \rightarrow$ OK

Wringing $\mathrm{UC} = \frac{V_{Ed}}{V_{c,Rd}} = \frac{1.0}{159.6} = 0.01 \rightarrow$ OK

Stabiliteitscontrole `:UGT`

Knikstabiliteit (*Axiale druk*) $\mathrm{UC} = \frac{N_{Ed}}{N_{b,Rd}} = \frac{96.6}{108.4} = 0.89 \rightarrow$ OK

Kipstabiliteit (*Buigend moment*) $\mathrm{UC} = \frac{M_{Ed}}{M_{b,Rd}} = \frac{0.8}{10.2} = 0.08 \rightarrow$ OK

Gecombineerd effect $\mathrm{UC} = \frac{N_{Ed}}{N_{b,Rd}} + \frac{\mu_y \, C_{my} \, \chi_{LT}}{\left(1 - \frac{N_{Ed}}{N_{cr,y}}\right) C_{yy}} \cdot \frac{M_{Ed}}{M_{b,Rd}} = \frac{1016.9}{1111.3} = 0.92 \rightarrow$ OK

Vervormingen `:GGT`

Uitzicht (9) $\mathrm{UC} = \frac{v_{max}}{h_1 \;/\; 250} = \frac{1.4}{10.4} = 0.14 \rightarrow$ OK

Geometrie en materiaal

De geometrie is af te lezen op de *situatieschets*.

geom = (L = 2.6)

```
geom = (
    L = 2.6, # m - verdiepingshoogte = 2.7m - 0.18 voor het profiel + 0.08cm uitgraving.
)
```

Profieldoorsnede

kolom = (naam = "SHS 90/4", kwaliteit = "S235", beschrijving = "Kolom 2", knikkromme = :a)

```
kolom = (
    naam = "SHS 90/4",
    kwaliteit = "S235",
    beschrijving = "Kolom 2",
    knikkromme = :a
)
```

	name	G	b	t	A	Av	I	Wel
1	"SHS 80/3.2"	7.63	80	3.2	972	486	0.9495	23.74
2	"SHS 70/6.3"	12.3	70	6.3	1563	781	1.038	29.67
3	"SHS 80/4"	9.41	80	4.0	1199	599	1.145	28.61
4	"SHS 70/8"	15.0	70	8.0	1915	958	1.198	34.22
5	"SHS 80/5"	11.6	80	5.0	1473	737	1.366	34.15
6	"SHS 80/6.3"	14.2	80	6.3	1815	907	1.619	40.47
7	"SHS 90/4"	10.7	90	4.0	1359	679	1.663	36.95
8	"SHS 80/8"	17.5	80	8.0	2235	1118	1.893	47.32
9	"SHS 90/5"	13.1	90	5.0	1673	837	1.996	44.35
10	"SHS 100/4"	11.9	100	4.0	1519	759	2.318	46.36

Haal de **eigenschappen** op van het gekozen profiel

`eig =`

DataFrameRow (16 columns)

	name	b	t	ro	ri	G	P	A	Av	I	Wel
	String	Int64	Int64	Int64	Int64	Float64	Float64	Int64	Int64	Float64	Float64
1	SHS 90/4	90	4	6	4	10.7	0.35	1359	679	1.663	36.95

Nominale waarde volgen *NBN EN 1993-1-1 §3.2*

`f_yk = 235`

```
f_yk = parse(Int64, kolom[:kwaliteit][2:end])
```

Rekenwaarden van materiaaleigenschappen volgens *NBN EN 1991-1-1 §3.2.6*

`E = 210000`

```
E = 210_000 # MPa
```

`G = 80769.23076923077`

```
G = E / (2 * (1 + ν)) # MPa
```

`ν = 0.3`

```
ν = 0.3 # Coëfficiënt van Poisson in het elastisch gebied
```

`z_max = 0.045`

```
z_max = +(eig.b / 1000) / 2 # m - Halve hoogte, van neutrale lijn tot uiterste vezel
```

```
z_min = -0.045
```

```
z_min = -(eig.b / 1000) / 2 # m - Halve hoogte, van neutrale lijn tot uiterste vezel
```

Belastingen

Overzicht van de aangrijpende belastingen

gevallen =

	naam	waarde	beschrijving
1	"GGT1"	67.3065	"Afdracht profiel 3 - lasten GGT Frequ
2	"GGT_K1"	70.7605	"Afdracht profiel 3 - lasten GGT Karak
3	"UGT1"	96.5629	"Afdracht profiel 3 - lasten UGT"

combinaties =

	check	naam	formule	uitkomst
1	:GGT	"F"	"GGT1"	67.3065
2	:GGT_K	"F"	"GGT_K1"	70.7605
3	:UGT	"F"	"UGT1"	96.5629

```
replacer = r"GGT1|GGT_K1|UGT1" => #1
```

maatgevend =

	check	F
1	:GGT	67.3065
2	:GGT_K	70.7605
3	:UGT	96.5629

rvw =

	check	F	L
1	:GGT	67.3065	2.6
2	:GGT_K	70.7605	2.6
3	:UGT	96.5629	2.6

Berekening

Berekening van de aangrijpende krachtswerking

Controle toepasbaarheid *1ste orderberekening* waar F_{cr} de elastiche kritieke (knik)belasting en F_{Ed} de rekenwaarde van de belastingen

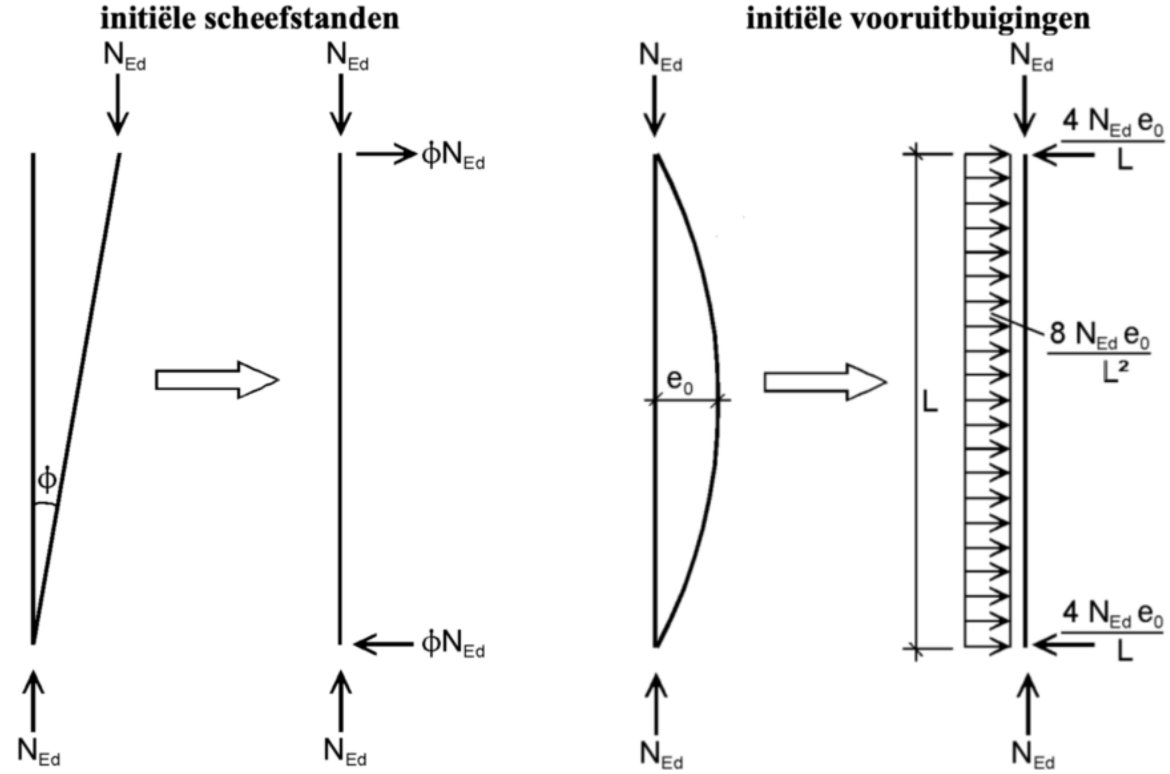
$$\alpha_{cr} = \frac{F_{cr}}{F_{Ed}} \geq 10 - \text{elastische berekening}$$

$$\alpha_{cr} = \frac{F_{cr}}{F_{Ed}} \geq 15 - \text{plastische berekening}$$

Geometrische imperfectie kolom

Gemetrische imperfectie volgen *NBN EN 1993-1-1 §5.3.2* is afhankelijk van de knikkromme (bepaald volgens tabel 6.2)

```
figuur5_4 =
```



Duiding van de scheefstand en vooruitbuiging en mogelijke **equivalente krachten** ter vervanging van de geometrische imperfecties

Initiële scheefstand

Draagt bij tot bijkomende knikgevoeligheid

```
ϕ = 1//200
```

```
ϕ = ϕ₀ * αₕ * αₘ |> rationalize
```

met

```
ϕ₀ = 1//200
```
```
ϕ₀ = 1//200 # Basiswaarde van de scheefstand
```

```
αₕ = 1.0
```
```
αₕ = max(min(2 / sqrt(geom[:L]), 1), 2//3)  # Reductiefactor voor de hoogte h
```

```
αₘ = 1.0
```
```
αₘ = sqrt( 0.5 * (1 + 1 // (m = 1; m))) # Reductiefactor meerdere kolommen
```

Initiële vooruitbuiging

Ter bepaling van de bijkomende buigingsknik - Bepaling van de verhouding e_0/L in functie van de knikkromme van de doorsnede, zie onderstaande tabel

`tabel5_1 =`

	knikkromme	elastisch	plastisch
1	:a0	1//350	1//300
2	:a	1//300	1//250
3	:b	1//250	1//200
4	:c	1//200	1//150
5	:d	1//150	1//100

```
e₀ = 1//300
```
```
e₀ = tabel5_1.elastisch[tabel5_1.knikkromme .== kolom[:knikkromme]] |> first # e_0
```

Belastingschema

Samenstel van de belastingseffecten van enerzijds de **initiële vervorming** en anderzijds de **intiële vooruitbuiging**

`N =`

$$F$$

```
N = F
```

`V =`

$$-F\phi - \frac{4Fe_0}{L} + \frac{8Fe_0t}{L^2}$$

```
V = V1 + V2
```

```
M =
```

$$-\frac{FL\phi}{2}+F\phi t+\frac{4Fe_0t}{L}-\frac{4Fe_0}{L}-\frac{4Fe_0t^2}{L^2}+\frac{-2FLe_0+12Fe_0}{3L}$$

```
M = M1 + M2
```

```
α =
```

$$\frac{-\frac{FL\phi t}{2}+\frac{F\phi t^2}{2}}{EI}+\frac{-\frac{2Fe_0t}{3}+\frac{2Fe_0t^2}{L}-\frac{4Fe_0t^3}{3L^2}}{EI}$$

```
α = α1 + α2
```

```
v =
```

$$\frac{-\frac{FL\phi t^2}{4}+\frac{F\phi t^3}{6}}{EI}+\frac{-\frac{2Fe_0t}{3}+\frac{2Fe_0t^2}{L}-\frac{4Fe_0t^3}{3L^2}}{EI}$$

```
v = v1 + v2
```

```
σ =
```

$$\frac{z\left(-\frac{FL\phi}{2}+F\phi t+\frac{4Fe_0t}{L}-\frac{4Fe_0}{L}-\frac{4Fe_0t^2}{L^2}+\frac{-2FLe_0+12Fe_0}{3L}\right)}{I}+\frac{F}{A}$$

```
σ = M * z / I + N / A
```

Interne krachtswerking

De interne krachtswerking heeft invloed op de bepaling van de doorsnede, alsook op de verdere stabiliteitscontrole (knik problematiek)

```
rvw_compleet =
```

	check	F	L	phi	e_o	I	A	EI
1	:GGT	67.3065	2.6	1//200	1//300	1.663e-6	0.001359	349.23
2	:GGT_K	70.7605	2.6	1//200	1//300	1.663e-6	0.001359	349.23
3	:UGT	96.5629	2.6	1//200	1//300	1.663e-6	0.001359	349.23

```
rvw_compleet = begin
    rvw_compleet = copy(rvw)
    rvw_compleet[!, :phi]   .= ϕ
    rvw_compleet[!, :e_0]   .= e₀
    rvw_compleet[!, :I]     .= eig.I * 10^-6 # m⁴
    rvw_compleet[!, :A]     .= eig.A * 10^-6 # m²
    rvw_compleet[!, :EI]    .= eig.I * E * 10^-3 # kNm²
    rvw_compleet
end
```

```
opl =
```

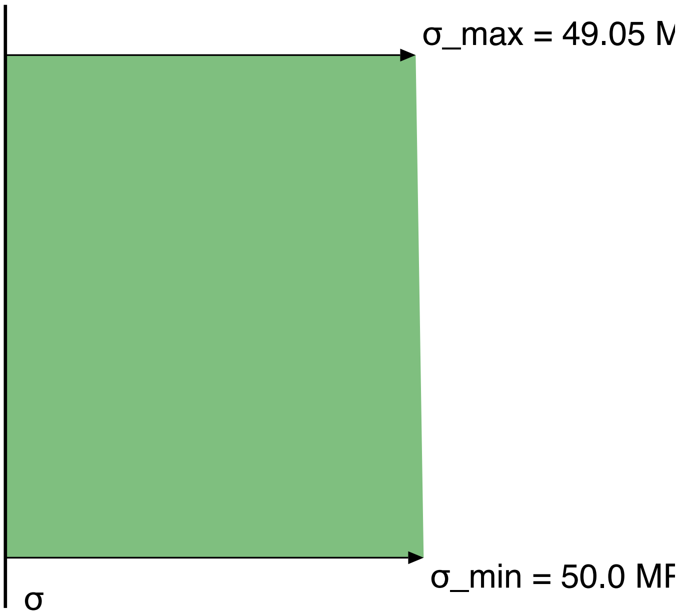
	check	N	V	M	α	v	σ
1	:GGT	#101	#101	#101	#101	#101	#101
2	:GGT_K	#101	#101	#101	#101	#101	#101
3	:UGT	#101	#101	#101	#101	#101	#101

Spanningsverdeling in de doorsnede

Voor een correcte inschatting van de classificatie van het profiel, moeten we de drukzone van de doorsnede kennen

Kies een locatie voor t: 1.05

Kies een grenstoestand: GGT



Classificatie van de doorsnede

Classificatie volgens doorsnede in categoriën die aangeven hoezeer de **weerstand** en **rotatiecapaciteit** zijn beperkt door de **plooiweerstand**. Voor een **klasse 1** bijvoorbeeld mag er zich een *plastisch scharnier* vormen zonder weerstandsverlies.

classificatie = $f(\text{deel onder druk onder beschouwde belastingscombinatie})$

Gebasseerd op het spanningsverloop langsheen de kolom (=$f(t)$) wordt de knikkromme bepaald via Tabel 5.2 in *NBN EN 1993-1-1*

```
classificatie (generic function with 1 method)
```

```
function classificatie(σ; f_y=235)
    # Berekening α
    σ_pos, σ_neg = σ(z_max), σ(z_min)
    sigma = LinearInterpolation(
        sort([σ_pos, σ_neg]), [1, 0], extrapolation_bc=Line())
    α = max(min(sigma(0), 1), 0) # Bepaal α, de locatie waar de spanning = 0
    z_e = 1 - (σ_neg + 2 * σ_pos) / (σ_neg + σ_pos) * eig.b / 3 # zwaartepunt
    # Bepalen classificatie
    c = eig.b - 2 * (eig.t + eig.ri)
    ε = sqrt(235 / f_y)
    ψ = 1 - 1 / α
    bounds =
        α > 0.5 ?
        [ # α > 0.5
            0.0,
            396 * ε / (13 * α - 1),      # Max waarde klasse 1 - plastisch
            456 * ε / (13 * α - 1),      # Max waarde klasse 2 - plastisch
            42 * ε / ( 0.67 + 0.33 * ψ) # Max waarde klasse 3 - elastisch
        ] :
        [
            0.0,
            36 * ε / α,                  # Max waarde klasse 1 - plastisch
            41.5 * ε / α,                # Max waarde klasse 2 - plastisch
            62 * ε * (1 - ψ) * sqrt(-ψ) # Max waarde klasse 3 - elastisch
        ]
    # Interpoleer waardes
    itp = LinearInterpolation(bounds, 1:4, extrapolation_bc=Line())
    return itp(c/eig.t)
end
```

klasse =

	check	klasse
1	:GGT	1.56061
2	:GGT_K	1.56061
3	:UGT	1.56061

De kolom **Kolom 2** van het type **SHS 90/4** in staalkwaliteit *S235* is van **klasse 1 en 1 en 1** in respectievelijk **GGT en GGT_K en UGT**.

```
(naam = "SHS 90/4", kwaliteit = "S235", beschrijving = "Kolom 2", knikkromme = :a)
kolom
```

Controle

Aftoetsen van de interne krachten en vervormingen

Controles

Maak gebruik van *enumerate* `Check` met waarde *false* of *true*

```
(1.0, 1.0, 1.25)
```

```
γ_M0, γ_M1, γ_M2 = 1.00, 1.00, 1.25
```

Uiterste grenstoestanden :UGT

Selecteer de abscis t = 0.0

Doorsnede controle

UGT =
DataFrameRow (4 columns)

	N	V	M	v
	#101#10...	#101#10...	#101#10...	#101#10...
1	#101	#101	#101	#101

```
UGT = opl[opl.check .== :UGT, [:N, :V, :M, :v]] |> first
```

Axiale druk

Berekening volgens *NBN EN 1993-1-1 §6.2.4*

N_Ed = 96.5629

```
N_Ed = UGT.N() # kN
```

N_cRd = 319.365

```
N_cRd = eig.A * f_yk / γ_M0 / 1000 # kN
```

UC_N =

$$\mathrm{UC} = \dfrac{N_{Ed}}{N_{c,Rd}} = \dfrac{96.6}{319.4} = 0.3 \rightarrow \text{OK}$$

```
UC_N = UC(md"$\dfrac{N_{Ed}}{N_{c,Rd}}$", N_Ed, N_cRd)
```

Buigend moment

Berekening volgens *NBN EN 1993-1-1 §6.2.5* mits in acht name van factor ρ voor combinatie met *Dwarskracht* volgens *NBN EN 1993-1-1 §6.2.8*

M_Ed = 0.8422430722222224

```
M_Ed = abs(UGT.M(t_)) # kNm
```

M_cRd = 10.24835

```
M_cRd = eig.Wpl * 10^3 * (1 - ρ) * f_yk / γ_M0 / 10^6 # kNm
```

```
UC_M =
```

$$UC = \frac{M_{Ed}}{M_{N,Rd}} = \frac{0.8}{9.3} = 0.09 \rightarrow \text{OK}$$

```
UC_M = UC(md"$\dfrac{M_{Ed}}{M_{N,Rd}}$", M_Ed, M_NRd)
```

Reductie M_{Rd} bij aanwezigheid van een normaalkracht

> **Toepasbaarheid**
>
> Onderstaande uiteenzetting geldt enkel voor profielen van **klasse 1 en 2**, zie *NBN EN 1993-1-1 §6.2.9.2-3* voor doorsneden van klasse 3 en 4. Ook is in onderstaande uitgegaan van één-assige buiging voor een kolom SHS (dus **stijfheid** in de **twee richting gelijk**)

```
M_NyRd = 9.347185744107744
M_NyRd = min(M_cRd * (1 - n) / (1 - 0.5 * a_w), M_cRd) # kNm - y-richting
```

```
M_NzRd = 9.347185744107744
M_NzRd = min(M_cRd * (1 - n) / (1 - 0.5 * a_f), M_cRd) # kNm - z-richting
```

```
M_NRd = 9.347185744107744
M_NRd = min(M_NyRd, M_NzRd) # Weerstand in de zwakke richting
```

```
a_w = 0.47019867549668876
a_w = min((eig.A - 2 * eig.b * eig.t) / eig.A, 0.5) # Gewalst buisprofiel
```

```
a_f = 0.47019867549668876
a_f = min((eig.A - 2 * eig.b * eig.t) / eig.A, 0.5) # Gewalst buisprofiel h=b
```

```
n = 0.30235905625225057
n = N_Ed / N_cRd
```

Dwarskracht (afschuiving)

Berekening volgens *NBN EN 1993-1-1 §6.2.6*

> **Toepasbaarheid**
>
> Onderstaande formule is enkel van toepassing voor **klasse 1 en 2**, voor de elastiche berekeningsmethode voor $V_{c,Rd}$ wordt voor het **kritiek punt** in de doorsnede de onderstaande vergelijking getest met S het statisch moment rond het beschouwde punt en t de dikte ervan.
>
> $$\frac{\tau_{Ed}}{f_y / \left(\sqrt(3) \gamma_{M0}\right)} \leq 1.0 \text{ met } \tau_{Ed} = \frac{V_{Ed}\, S}{I\, t}$$

```
V_Ed = 0.978008858974359
```

```
V_Ed = abs(UGT.V(t_)) # kN
```

```
V_plRd = 159.565
```

```
V_plRd = eig.Av * f_yk / γ_M0 / 1000 # kN
```

```
UC_V =
```

$$\mathrm{UC} = \dfrac{V_{Ed}}{V_{c,Rd}} = \dfrac{1.0}{159.6} = 0.01 \rightarrow \text{OK}$$

```
UC_V = UC(md"$\dfrac{V_{Ed}}{V_{c,Rd}}$", V_Ed, V_plRd)
```

```
ρ = 0
```

```
ρ = V_Ed / V_plRd < 0.5 ? 0 : ((2 * V_Ed) / V_plRd - 1)^2
```

Wringing (torsie)

Berekening volgens *NBN EN 1993-1-1 §6.2.7*

> **Geen wringing**
>
> Geen wringing (torsie) grijpt aan. De eenheidscontrole (`UC`) voldoet

```
UC_T =
```

$$\mathrm{UC} = \dfrac{N_{Ed}}{N_{c,Rd}} + \dfrac{M_{Ed}}{M_{N,Rd}} + \dfrac{V_{Ed}}{V_{c,Rd}} = \dfrac{189861.8}{476327.7} = 0.4 \rightarrow \text{OK}$$

```
UC_T = UC_N + UC_M + UC_V
```

Stabiliteit controle

Controle van de knikstabiliteit volgens *NBN EN 1993-1-1 §6.3.1*

Knikstabiliteit bij op druk belaste staven

Volgens *NBN EN 1993-1-1 §6.3.1.1*

> **Toepasbaarheid**
>
> Onderstaande uiteenzetting geldt enkel voor staven van **klasse 1, 2 en 3**

```
N_bRd = 108.4384030634186
```

```
N_bRd = χ_ * eig.A * f_yk / γ_M1 / 1000 # kN
```

Bepalen *reductiefactor* voor **knik**

```
χ_ = 0.3395437917850065
```

```
χ_ = min(1 / (ϕ_ + sqrt(ϕ_ ^ 2 - λ_ ^2)), 1) # reductiefactor knikvorm
```

```
ϕ_ = 1.8979161409846397
```

```
ϕ_ = 0.5 * (1 + α_ * (λ_ - 0.2) + λ_ ^ 2)
```

```
λ_ = 1.582855823809984
```

```
λ_ = sqrt( eig.A * f_yk / N_cr / 1000 ) # rel. slankheid - klasse 1, 2 en 3 doorsneden
```

```
α_ = 0.21
```

```
α_ = getproperty(imperfectie, kolom[:knikkromme]) # imperfectiefactor
```

```
N_cr = 127.46900684143628
```

```
N_cr = π ^ 2 * ((try eig.Iz catch; eig.I end) * E * 10^-3) / L_cr ^2 # kN - Eulerknik
```

```
L_cr = 5.2
```

```
L_cr = 2 * geom[:L] # m - Kniklengte, kolom onderaan vast, bovenaan vrij
```

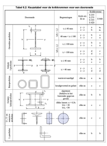
De imperfectiefactor α is in overeenstemming met de knikkromme volgens Tabel 6.2 in *NBN EN 1993-1-1*. **Knikeffecten** kunnen worden **verwaarloost indien**

$$\bar{\lambda} \leq 0.2 \text{ of } \frac{N_{Ed}}{N_{cr}} \leq 0.04$$

```
imperfectie =
```

DataFrameRow (5 columns)

	ao Float64	a Float64	b Float64	c Float64	d Float64
1	0.13	0.21	0.34	0.49	0.76

```
Foldable("Table 6.2 in NBN EN 1993-1-1 §6.3.3", 
```

```
)
```

```
UC_Nstab =
```

$$\text{UC} = \frac{N_{Ed}}{N_{b,Rd}} = \frac{96.6}{108.4} = 0.89 \rightarrow \text{OK}$$

Kipstabiliteit bij op buiging belaste staven

Volgens *NBN EN 1993-1-1 §6.3.2*

Toepasbaarheid

Onderstaande uiteenzetting geldt enkel voor staven van **klasse 1 en 2** waarbij voor het weerstandsmoment het plastische weerstandsmoment $W_{pl,y}$ mag gehanteerd worden.

```
M_bRd = 10.24835
```

```
M_bRd = χ_LT * eig.Wpl * 10^3 * f_yk / γ_M1 / 10^6 # kNm
```

Bepalen *reductiefactor* χ_{LT} voor **kip** waarbij $_{LT}$ staat voor *lateral torsional* buckling

```
χ_LT = 1.0
```

```
χ_LT = min(1 / (ϕ_LT + sqrt( ϕ_LT ^ 2 - λ_LT ^ 2 )), 1.0) # Reductiefactor kip
```

```
ϕ_LT = 0.4989992165165405
```

```
ϕ_LT = 0.5 * (1 + α_LT * (λ_LT - 0.2) + λ_LT ^ 2)
```

Eenvoudige benadering λ_{LT} volgens *NBN EN 1993-1-1 ANB - Bijlage H*.

```
λ_LT = 0.12088367146184138
```

```
λ_LT = sqrt( (eig.Wpl * f_yk) / M_cr / 1000 ) # relatieve slankheid
```

```
α_LT = 0.21
```

```
α_LT = getproperty(imperfectie, :a) # imperfectiefactor
```

Berekening M_{cr} volgens *NBN EN 1993-1-1 ANB - Bijlage E*.

Dubbelsymmetrische dwarsdoorsnede (§3 *Bijlage E*)

$$M_{cr} = C_1 \frac{\pi^2 EI_z}{(k_z L)^2} \left\{ \sqrt{\left[\left(\frac{k_z}{k_\omega} \right)^2 \frac{I_\omega}{I_z} + \frac{(k_z L)^2 G I_T}{\pi^2 E I_z} + (C_2 z_g)^2 \right]} - C_2 z_g \right\}$$

Eenheden eigenschappen

Als brondbestand voor de liggers, werd een overzichtstabel van Areloc Mittal gehanteerd. Als bronbestand voor de kolommen, werd een lijst van **eurocodeapplied** gehanteerd. Controleer of er geen vergroting van de parameters moet toegepast worden.

```
M_cr = 701.3239419970824
```

```
#M_cr = eig.Wpl * f_yk / 1000 # kNm - Kritisch moment; SHS niet kip gevoelig
M_cr = C_1 * (π ^ 2 * ((try eig.Iz catch; eig.I end) * E * 10^-3)) / (k_z * L_steun)^2 * ( sqrt((k_z / k_ω)^2 * (try eig.Iw * 10^3 catch; 0 end) / (try eig.Iz catch; eig.I end) + (k_z * L_steun) ^2 * G * eig.IT / (π^2 * E * (try eig.Iz catch; eig.I end) * 10^3) + (C_2 * z_g)^2) - C_2 * z_g)
```

```
L_steun = 2.6
```

```
L_steun = geom[:L] # Lengte van de ligger tussen punten met zijdelingse steun
```

```
k_z = 0.7
```

```
# Effectieve lengtefactor k_z = betrekking tot einddraaiing in het vlak
#   - 0.5 volledig ingeklemd
#   - 0.7 één einde ingeklemd, één zijde vrije
#   - 1.0 volledig gebrek aan inklemming
k_z = 0.7 # Onderaan ingeklemd
```

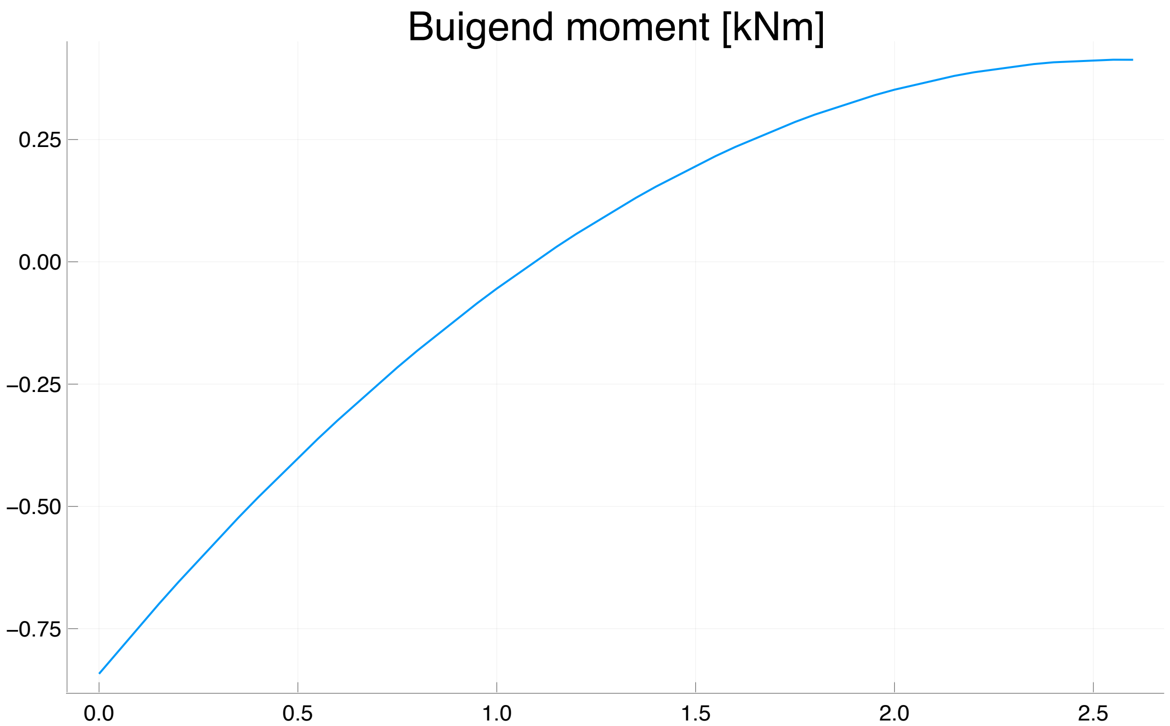
```
k_ω = 1.0
```

```
# Effectieve lengtefactor k_ω = betrekking tot welving van het uiteinde
#   - 0.5 volledig ingeklemd
#   - 0.7 één einde ingeklemd, één zijde vrije
#   - 1.0 volledig gebrek aan inklemming
k_ω = 1.0 # Kolom kan aan uiteindes niet welven door druk
```

```
z_g = 0
```

```
z_g = 0 # z_a - z_s = verschil belastingspunt tot het zwaartepunt
```

Berekening van de factoren C_1, C_2 en C_3 volgens de tabellen *Tabel E.1 ANB* en *Tabel E.2 ANB*




```
begin
    UGT_M = UGT.M.(0:0.05:geom[:L])
    UGT_Mmax, UGT_M1, UGT_M2 = maximum(UGT_M), UGT_M[1], UGT_M[end]
    # Ψ is de verhouding tussen de eindmomenten, M is het max eindmoment
    Ψ = abs(UGT_M1) > abs(UGT_M2) ? UGT_M2 / UGT_M1 : UGT_M1 / UGT_M2
    plot(0:0.05:geom[:L], UGT_M, title="Buigend moment [kNm]", legend=false)
end
```

```
Ψ_f = 0
```

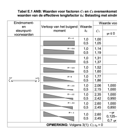
```
Ψ_f = 0 # Voor dubbelsymmetrische doorsneden
```

De verhouding tussen de eindmomenten ψ is gelijk aan -0.49.

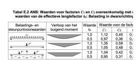
Lees de waardes van C_1, C_2 en C_3 af op onderstaande grafieken indien afkomstig uit zuiver **eindmomenten** of **krachten** (lijn-/puntbelasting) in de dwarsrichting.

```
# Indien k_z = 1.0, volledig gebrek aan inklemming
# Benaderde waarde van C_1 via onderstaande forumule
# C1 = min(1.77 - 1.04 * Ψ + 0.27 * Ψ^2, 2.60)
```

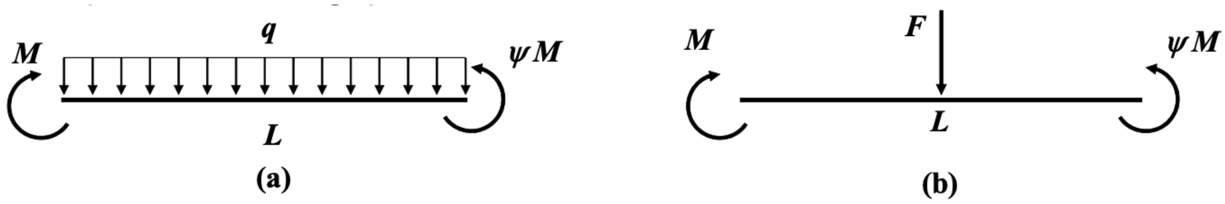
```
Foldable("Tabel E.1 volgens NBN EN 1993-1-1 ANB - Belasting door eindmomenten",
```



```
Foldable("Tabel E.2 volgens NBN EN 1993-1-1 ANB - Belasting in dwarsrichting",
```



Indien het momentverloop is ontstaan uit het **gecombineerd effect** van zowel belastingen door **eindmomenten** (inklemming) als door **lijn- en puntbelastingen**, dan worden de figuren *E.3 ANB* tot *E.10 ANB* gehanteerd.



Figuur E.2 ANB: Eindmomenten gecombineerd met een dwarse belasting

Hierbij worden volgende parameters berekend, waarbij M voor het maximale **eindmoment** staat:

$$\mu = \frac{q\,L^2}{8\,M}(\mathrm{a})\ \mathrm{of}\ \frac{F\,L}{4\,M}(\mathrm{b})$$

```
μ = 0.3529378509780274
```

```
μ = (UGT_M[div(end,2)] - (UGT_M1 + UGT_M2) / 2) / maximum(abs.([UGT_M1, UGT_M2]))
```

In geval van de berekening van de kolommen, nemen we altijd een **positieve** waarde aan van μ, gezien dit leidt tot de kleinste waardes voor de constantes C_1 tot C_3, immers zijn de buigende momenten afkomstig van geometrische imperfecties en kunnen die elkaars effecten versterken.

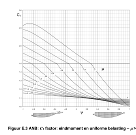
Herevalueer parameters

Bij een wijziging van de belastingen dienen de parameters C_1 en C_2 opnieuw geëvalueerd worden

Bepaal de coëfficiënten voor μ = 0.353 en ψ = -0.49

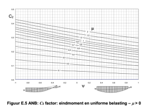
```
C_1 = 1.5
```

```
C_1 = 1.5 # Zie volgende figuur
```

Foldable("Figuur E.3 volgens NBN EN 1993-1-1 ANB - μ > 0", )

```
C_2 = 0.18
```

```
C_2 = 0.18 # Zie volgende figuur
```

Foldable("Figuur E.5 volgens NBN EN 1993-1-1 ANB - μ > 0", )

De imperfectiefactor $\alpha_L T$ is in overeenstemming met de kipkromme volgens Tabel 6.3 in *NBN EN 1993-1-1*. **Kipeffecten** kunnen worden **verwaarloost indien**

$$\bar{\lambda}_{LT} \leq \bar{\lambda}_{LT,0} \text{ of } \frac{M_{Ed}}{M_{cr}} \leq \bar{\lambda}_{LT,0}^2$$

Waarin $\bar{\lambda}_{LT,0}$ volgens *NBN EN 1993-1-1 §6.3.2.3* gelijk is aan 0.4

```
imperfectie_LT =
```
DataFrameRow (4 columns)

	a	b	c	d
	Float64	Float64	Float64	Float64
1	0.21	0.34	0.49	0.76

```
UC_Mstab =
```

$$\text{UC} = \frac{M_{Ed}}{M_{b,Rd}} = \frac{0.8}{10.2} = 0.08 \rightarrow \text{OK}$$

Prismatische, op buiging en druk belaste staven

Volgens *NBN EN 1993-1-1 §6.3.3*

Toepasbaarheid

Onderstaande uiteenzetting geldt enkel voor staven van **klasse 1 en 2** waarbij voor het weerstandsmoment het plastische weerstandsmoment $W_{pl,y}$ mag gehanteerd worden.

Staven die aan gecombineerde buiging en druk zijn onderworpen behoren te voldoen aan:

$$\frac{N_{Ed}}{\frac{\chi_y N_{Rk}}{\gamma_{M1}}} + k_{yy} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{\frac{\chi_{LT} M_{y,Rk}}{\gamma_{M1}}} \leq 1$$

Merk op dat bovenstaande vergelijking een vereenvoudiging is van formule 6.62 en 6.63 uit *NBN EN 1993-1-1* gezien voor een **SHS** profiel de sterkte in beide richtingen gelijk is. De parameter k_{yy} is een interactiefactor, berekend volgens **bijlage A** (Alternatieve methode 1 normatief). De bijlage is herschreven en toegevoegd als **Bijlage D** in *NBN EN 1993-1-1 ANB*. De parameter $\Delta M_{y,Ed}$ voor een profiel van **klasse 1 tot 3** is gelijk aan 0

Stap 1 - Onderscheid staven gevoelig of niet aan torsie

Volgens *NBN EN 1993-1-1 ANB Bijlage D*

Is de staaf **torsie gevoelig**?

```
torsie_gevoelig = false
```

```
torsie_gevoelig = begin
    if eig.IT > eig.I * 1000
        false # Niet gevoelig voor vervorming door torsie
    else
        if λ_LT0 <= 0.2 * sqrt(C_1) * ((1 - N_Ed / N_crz) * (1 - N_Ed / N_crTF))^(1/4)
            false # Niet gevoelig voor vervorming door torsie
        else
            true # Gevoelig voor vervorming door torsie
        end
    end
end
```

```
λ_LT0 = 0.14587963198573276
```

```
λ_LT0 = min(sqrt(eig.Wpl * f_yk / M_cr0 / 1000), 0.4) # §6.3.2.3 uit NBN EN 1993-1-1
```

```
M_cr0 = 481.5757735046633
```

```
M_cr0 = 1.03 * (π ^ 2 * ((try eig.Iz catch; eig.I end) * E * 10^-3)) / (k_z * L_steun)^2 * sqrt((k_z / k_ω)^2 * (try eig.Iw * 10^3 catch; 0 end) / (try eig.Iz catch; eig.I end) + (k_z * L_steun) ^2 * G * eig.IT / (π^2 * E * (try eig.Iz catch; eig.I end) * 10^3))
```

Berekening **kritieke elastische kracht** met betrekking tot de **torsieknikstabiliteit** $N_{cr,TF}$ en **Euleriaanse elastische knikkracht** $N_{cr,z}$ om de z-as volgens *NBN EN 1993-1-1 ANB Bijlage F*. Merk op, onderstaande berekening is enkel geldig voor een prismatische doorsnede waarbij het zwaartepunt van de doorsnede samenvalt met het torsiecentrum. In onderstaande berekening wordt voor gesloten kokerprofielen de welvingconstante gelijk gesteld aan 0.

```
N_crTF = 85838.77492020908
```

```
N_crTF = 1 / (r_0 ^ 2) * (G * eig.IT * 10^3) / 1000 # kN
```

```
r_0 = 49.47108029077321
```

```
r_0 = try sqrt((eig.Iy + eig.Iz) * 10^6 / eig.A) catch; sqrt((2 * eig.I) * 10^6 / eig.A) end
```

De relevante kniklengte L_{cr} is gelijk aan $2\times$ **de lengte van de kolom** omdat deze onderaan als ingeklemd wordt beschouwd, en bovenaan als vrij. Dit is een conservatieve benadering.

```
5.2
```

```
L_cr
```

```
N_crz = 127.46900684143628
```

```
N_crz = N_cr
```

De factor C_1 hangt af van de **belasting** en de **randvoorwaarden** van de **oplegging** en is bepaald volgens *NBN EN 1993-1-1 ANB Bijlage E*

Stap 2.a - Staaf niet gevoelig voor torsie

Volgens *NBN EN 1993-1-1 ANB Bijlage D*.

We hernemen formules (6.61) en (6.62) uit *NBN EN 1993-1-1* en passen hier alle randvoorwaardes op toe. Prismatische dubbelsymmetrische koker met sterkte in beide assen gelijk.

$$\frac{N_{Ed}}{\frac{\chi_y N_{Rk}}{\gamma_{M1}}} + \mu_y \left[C_{my} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{\left(1 - \frac{N_{Ed}}{N_{cr,y}}\right) C_{yy} \frac{M_{y,Rk}}{\gamma_{M1}}} \right] \leq 1$$

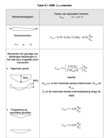
Bovenstaande wordt opgesplitst in de eenheidscontrole voor **enkel druk** en het aangrijpen van **enkel** een **buigend moment**, de overige waardes worden samengevoegd tot een **interactiefactor** k_{yy} uit de **Alternatieve methode 2**, we verwijzen dan ook verder naar k_{yy} als naam voor deze term.

$$\frac{N_{Ed}}{\frac{\chi_y N_{Rk}}{\gamma_{M1}}} + \frac{\mu_y C_{my} \chi_{LT}}{\left(1 - \frac{N_{Ed}}{N_{cr,y}}\right) C_{yy}} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{\frac{\chi_{LT} M_{y,Rk}}{\gamma_{M1}}} \leq 1$$

```
μ_y = 0.326421213912173
```

```
μ_y = (1 - N_Ed / N_cr ) / (1 - χ_ * N_Ed / N_cr )
```

De waarde voor de factor C_{my} wordt afgeleid van *Tabel D.1 ANB*. Voor een staaf niet gevoelig voor torstie is $C_{my} = C_{my,0}$.

```
Foldable("Tabel D.5 volgens NBN EN 1993-1-1 ANB",
```

```
)
```

```
C_my0 = 0.24245977596641677
```

```
C_my0 = 1 + ( π ^ 2 * E * (try eig.Iy catch; eig.I end) * 10^(-3) * maximum(UGT.v.
(0:0.05:geom[:L])) / (geom[:L]^2 * UGT_Mmax) - 1 ) * N_Ed / N_cr
```

Interactiefactor C_{yy} die een maat is voor de hoeveelheid plasticiteit in de staafdoorsnede bij bezwijken.

$$C_{yy} = 1 + (w_y - 1)\left[\left(2 - \frac{1.6}{w_y} C_{my}^2 \bar{\lambda}_{max} - \frac{1.6}{w_y} C_{my}^2 \bar{\lambda}_{max}^2\right) n_{pl}\right] \geq \frac{W_{el,y}}{W_{pl,y}}$$

```
C_yy0 = 1.0912402711188565
```

```
C_yy0 = max(1 + (w_y - 1) * ((2 - (1.6 / w_y * C_my0 ^ 2) * (λ_max + λ_max ^ 2)) *
n_pl), eig.Wel / eig.Wpl)
```

```
w_y = 1.1802435723951283
```

```
w_y = min(eig.Wpl / eig.Wel, 1.5)
```

λ_max = 1.582855823809984

```
λ_max = try max(λ_y, λ_z) catch; λ_ end
```

n_pl = 0.30235905625225057

```
n_pl = N_Ed / N_cRd / γ_M1
```

k_yy_a =

$$\frac{\mu_y\, C_{my}\, \chi_{LT}}{\left(1 - \frac{N_{Ed}}{N_{cr,y}}\right) C_{yy}} = 0.299$$

Stap 2.b - Staaf gevoelig voor torsie

Volgens *NBN EN 1993-1-1 ANB Bijlage D*

We hernemen formules (6.61) en (6.62) uit *NBN EN 1993-1-1* en passen hier alle randvoorwaardes op toe. Prismatische dubbelsymmetrische koker met sterkte in beide assen gelijk.

$$\frac{N_{Ed}}{\frac{\chi_y\, N_{Rk}}{\gamma_{M1}}} + \mu_y \left[\frac{C_{mLT}\, C_{my}}{\chi_{LT}} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{\left(1 - \frac{N_{Ed}}{N_{cr,y}}\right) C_{yy} \frac{M_{y,Rk}}{\gamma_{M1}}} \right] \leq 1$$

Bovenstaande wordt opgesplitst in de eenheidscontrole voor **enkel druk** en het aangrijpen van **enkel** een **buigend moment**, de overige waardes worden samengevoegd tot een **interactiefactor** k_{yy} uit de **Alternatieve methode 2**, we verwijzen dan ook verder naar k_{yy} als naam voor deze term.

$$\frac{N_{Ed}}{\frac{\chi_y\, N_{Rk}}{\gamma_{M1}}} + \frac{\mu_y\, C_{mLT}\, C_{my}}{\left(1 - \frac{N_{Ed}}{N_{cr,y}}\right) C_{yy}} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{\frac{\chi_{LT} M_{y,Rk}}{\gamma_{M1}}} \leq 1$$

C_my = -0.11315802416834947

```
C_my = C_my0 + (1 - C_my0) * (sqrt(ε_y) * a_LT) / (1 + sqrt(ε_y) * a_LT)
```

ε_y = 0.3207983761840325

```
ε_y = M_Ed / N_Ed * eig.A / eig.Wel # Voor klasse 1, 2 of 3
```

a_LT = -0.5640408899579075

```
a_LT = 1 - eig.IT  / ((try eig.Iy catch; eig.I end) * 1000)
```

Interactiefactor C_{yy} die een maat is voor de hoeveelheid plasticiteit in de staafdoorsnede bij bezwijken.

$$C_{yy} = 1 + (w_y - 1)\left[\left(2 - \frac{1.6}{w_y} C_{my}^2 \bar{\lambda}_{max} - \frac{1.6}{w_y} C_{my}^2 \bar{\lambda}_{max}^2\right) n_{pl} - b_{LT}\right] \geq \frac{W_{el,y}}{W_{pl,y}}$$

`C_yy` = 1.1051289369584734

```
C_yy = max(1 + (w_y - 1) * ((2 - (1.6 / w_y * C_my ^ 2) * (λ_max + λ_max ^ 2)) *
n_pl - b_LT), eig.Wel / eig.Wpl)
```

`b_LT` = 0

```
b_LT = 0 # Maar buiging in 1 richting
```

De factor C_{mLT} dekt de invloed van de normaalkracht en van de doorsnedevorm op de kipweerstand.

`C_mLT` = -0.014675937869061295

```
C_mLT = C_my ^ 2 * a_LT / sqrt((1 - N_Ed / N_cr) * (1 - N_Ed / N_crTF))
```

`k_yy_b` =

$$\frac{\mu_y \, C_{mLT} \, C_{my}}{\left(1 - \dfrac{N_{Ed}}{N_{cr,y}}\right) C_{yy}} = 0.002$$

Unity check

Onderstaande vorm aangehouden omdat dit in overeenstemming is met *Alternatieve methode 2* opgenomen in *NBN EN 1993-1-1 Bijlage B*. De controle zelf is wel uitgevoerd volgens *Alternatieve methode 1* omdat deze normatief is in België.

`UC_MNstab` =

$$\mathrm{UC} = \frac{N_{Ed}}{N_{b,Rd}} + \frac{\mu_y \, C_{my} \, \chi_{LT}}{\left(1 - \dfrac{N_{Ed}}{N_{cr,y}}\right) C_{yy}} \cdot \frac{M_{Ed}}{M_{b,Rd}} = \frac{1016.9}{1111.3} = 0.92 \rightarrow$$ OK

```
UC_MNstab = torsie_gevoelig ? UC_Nstab + k_yy_b * UC_Mstab : UC_Nstab + k_yy_a *
UC_Mstab
```

Bruikbaarheidsgrenstoestanden :GGT

Selecteer de abscis t = 0.0

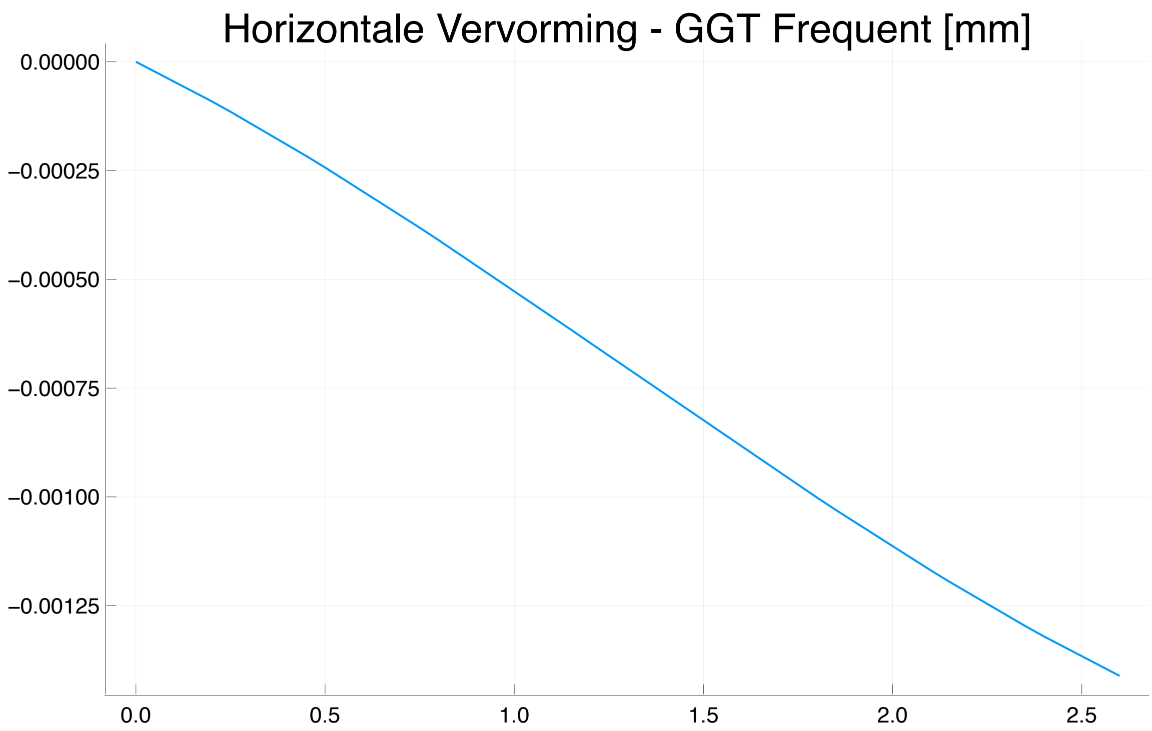
Horizontale verplaatsing

Volgens *NBN EN 1993-1-1 §7.2.2* met verwijzing naar *NBN EN 1990 - bijlage A1.4*. In de natianale bijlage (*ANB*) wordt verwezen naar de norm *NBN B 03-003* die toegepast dient te worden in België. Grenswaarden van de vervorming wordt vastgelegd in tabel 3 onder §7.

GGT_F =
DataFrameRow (7 columns)

	check	N	V	M	α	v	σ
	Symbol	#101#10...	#101#10...	#101#10...	#101#10...	#101#10...	#101#10...
1	GGT	#101	#101	#101	#101	#101	#101

```
GGT_F = opl[opl.check .== :GGT,:] |> first
```



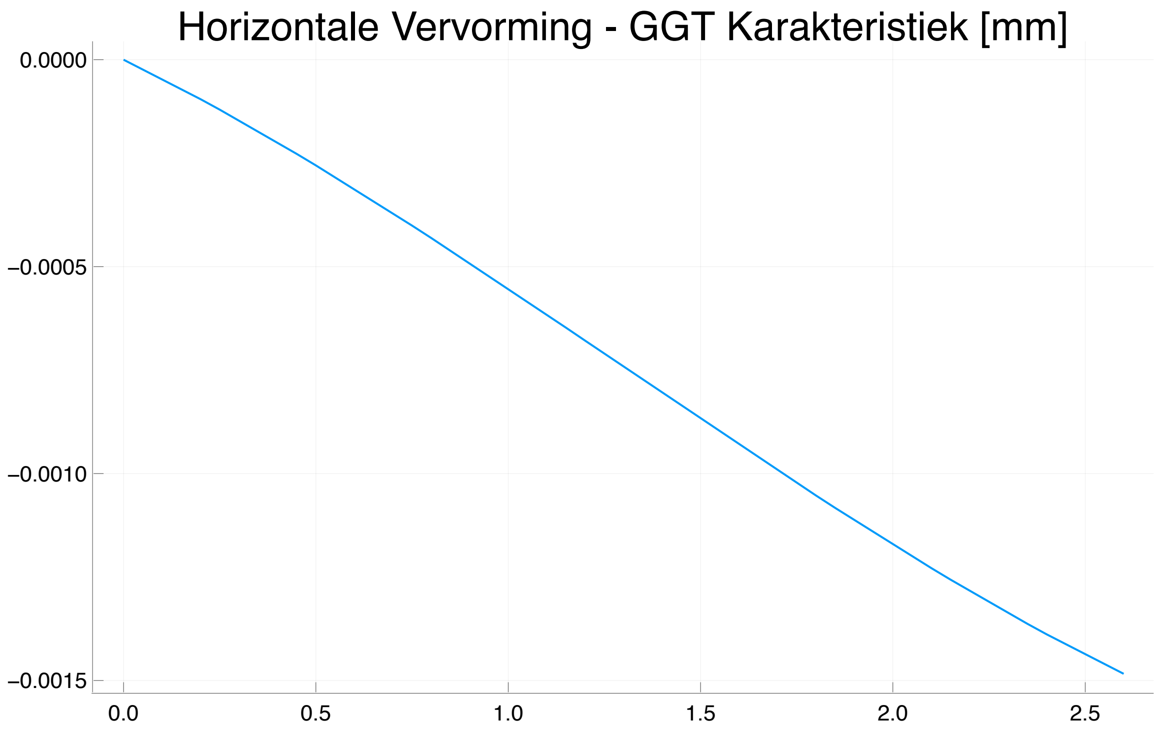
GGT_K_vmax = 0.0014838436799053155

```
GGT_K_vmax = maximum(extrema(GGT_K.v.(0:0.05:geom[:L])) .|> abs)
```

GGT_K =
DataFrameRow (7 columns)

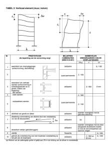
	check	N	V	M	α	v	σ
	Symbol	#101#10...	#101#10...	#101#10...	#101#10...	#101#10...	#101#10...
1	GGT_K	#101	#101	#101	#101	#101	#101

```
GGT_K = opl[opl.check .== :GGT_K,:] |> first
```



GGT_F_vmax = 0.0014114134954041‍74

```
GGT_F_vmax = maximum(extrema(GGT_F.v.(0:0.05:geom[:L])) .|> abs)
```

Foldable("Afdruk NBN B 03-003 Tabel 3", )

UC_v =

$$\text{UC} = \frac{v_{max}}{h_1 \ / \ 250} = \frac{1.4}{10.4} = 0.14 \rightarrow \text{OK}$$

Achterliggende berekeningen

Hieronder worden algemene controles uitgewerkt

Dependencies en hulpfuncties

```
using PlutoUI    , ImageView    , Images    , Plots    , SymPy    , Luxor    , SQLite    ,
DataFrames    , Underscores    , Interpolations    , PlutoTest
```

Laad de database in

```
db = SQLite.DB("assets/db/db.sqlite")
```

```julia
db = SQLite.DB("assets/db/db.sqlite")
```

Start de `plotly` backend

```
PlotlyBackend()
```

```julia
plotly()
```

Eigen `Check` type met ook een eigen uitdraai

```julia
using UUIDs
```

```julia
@enum Check OK=true NOK=false
```

```julia
function Base.show(io::IO, mime::MIME"text/html", c::Check)
    id = UUIDs.uuid4().value
    color = (c == Check(false)) ? "crimson" : "yellowgreen"
    Base.write(io, """
        <style>
        .marked-$id {
            color: white;
            background-color: $(color);
            padding: 0 2px;
        }
        </style>
        <mark class="marked-$id">$(string(c))</mark>
        """)
end
```

Definieer een nieuwe *type* getiteld *Unity Check* of `UC`

```julia
struct UC
    beschrijving::Markdown.MD
    waarde::Float64 # teller
    limiet::Float64 # noemer
end
```

```julia
struct Constant
    beschrijving::Markdown.MD
    waarde::Real
end
```

```julia
function Base.show(io::IO, mime::MIME"text/html", uc::UC)
    afronden = t -> (d -> round(d, digits=t))
    subs = Dict(
        "beschrijving" => uc.beschrijving.content[1].formula,
        "waarde" => uc.waarde |> afronden(1),
        "limiet" => uc.limiet |> afronden(1),
        "uc" => (uc.waarde / uc.limiet) |> afronden(2)
    )
    format = raw"$\text{UC} = beschrijving = \dfrac{waarde}{limiet} =
uc\rightarrow$"

    Base.write(io, """
        <div style="display: flex; align-items: center; justify-content: center;">
            <div>
    """)
    show(io, mime, Markdown.parse(
            replace(format, r"beschrijving|waarde|limiet|uc" => s -> subs[s]))
    )
    Base.write(io, """
            </div>
            <div style="flex: 1; padding-left: 2px;">
    """)
    show(io, mime, Check(subs["uc"] <= 1))
    Base.write(io, """
            </div>
        </div>
    """)
end
```

```julia
function Base.show(io::IO, mime::MIME"text/html", c::Constant)
    afronden = t -> (d -> round(d, digits=t))

    show(io, mime, Markdown.MD(
            Markdown.LaTeX(
                string(
                    first(c.beschrijving.content).formula,
                    "=",
                    c.waarde |> afronden(3)
                )
            )
        )
    )
end
```

```julia
Base.copy(uc::UC) = UC(uc.beschrijving, uc.waarde, uc.limiet)
```

Algemene functies om somaties van `Markdown.MD` types mogelijk te maken, alsook de sommatie van de *custom* `UC` *struct*

```julia
function Base.:+(uc1::UC, uc2::UC)::UC
    c1, c2 = getproperty.([uc1, uc2], :beschrijving)
    w1, w2 = getproperty.([uc1, uc2], :waarde)
    l1, l2 = getproperty.([uc1, uc2], :limiet)
    return UC(c1 + c2, w1 * l2 + w2 * l1, l1 * l2)
end
```

```julia
function Base.:+(md1::Markdown.MD, md2::Markdown.MD)::Markdown.MD
    pgs1, pgs2 = md1.content, md2.content
    if length(pgs1) == 1 && length(pgs2) == 1
        pg1, pg2 = (pgs1, pgs2) .|> first
        tp1, tp2 = (pg1, pg2) .|> typeof
        if all((tp1, tp2) .== Markdown.LaTeX)
            c1, c2 = getproperty.([pg1, pg2], :formula)
            return Markdown.MD(Markdown.LaTeX(string(c1,"+",c2)))
        elseif all((tp1, tp2) .== Markdown.Paragraph)
            c1, c2 = getproperty.([pg1, pg2], :content) .|> first
            return Markdown.MD(Markdown.Paragraph(string(c1," ",c2)))
        else
            return Markdown.MD(Markdown.Paragraph("..."))
        end
    end
end
```

```julia
function Base.:*(md1::Markdown.MD, md2::Markdown.MD)::Markdown.MD
    pgs1, pgs2 = md1.content, md2.content
    if length(pgs1) == 1 && length(pgs2) == 1
        pg1, pg2 = (pgs1, pgs2) .|> first
        tp1, tp2 = (pg1, pg2) .|> typeof
        if all((tp1, tp2) .== Markdown.LaTeX)
            c1, c2 = getproperty.([pg1, pg2], :formula)
            return Markdown.MD(Markdown.LaTeX(string(c1,raw"\cdot ",c2)))
        elseif all((tp1, tp2) .== Markdown.Paragraph)
            c1, c2 = getproperty.([pg1, pg2], :content) .|> first
            return Markdown.MD(Markdown.Paragraph(string(c1," ",c2)))
        else
            return Markdown.MD(Markdown.Paragraph("..."))
        end
    end
end
```

```julia
function Base.:*(scalar::Real, uc::UC)::UC
    beschrijving = Markdown.MD(Markdown.LaTeX("$scalar"))  * uc.beschrijving
    waarde = scalar * uc.waarde
    return UC(beschrijving, waarde, uc.limiet)
end
```

```julia
function Base.:/(scalar::Real, uc::UC)::UC
    beschrijving = Markdown.MD(Markdown.LaTeX(raw"\dfrac{1}{$scalar}"))  *
uc.beschrijving
    limiet = scalar * uc.limiet
    return UC(beschrijving, uc.waarde, limiet)
end
```

```julia
function Base.:*(c::Constant, uc::UC)::UC
    beschrijving = c.beschrijving  * uc.beschrijving
    waarde = c.waarde * uc.waarde
    return UC(beschrijving, waarde, uc.limiet)
end
```

```julia
function Base.:/(c::Constant, uc::UC)::UC
    beschrijving = c.beschrijving  * uc.beschrijving
    limiet = c.waarde * uc.limiet
    return UC(beschrijving, uc.waarde, limiet)
end
```

Opsplitsing in kolommen

```julia
struct TwoColumn{L, R}
    left::L
    right::R
end
```

```julia
function Base.show(io::IO, mime::MIME"text/html", tc::TwoColumn)
    Base.write(io, """
        <div style="display: flex; align-items: center; justify-content: center;">
            <div style="flex: 50%; overflow-x: scroll;">
    """)
    show(io, mime, tc.left)
    Base.write(io, """
            </div>
            <div style="flex: 50%;">
    """)
    show(io, mime, tc.right)
    Base.write(io, """
            </div>
        </div>
    """)
end
```

Maak een html `collapse` element

```julia
struct Foldable{C}
    title::String
    content::C
end
```

```julia
function Base.show(io::IO, mime::MIME"text/html", fld::Foldable)
    Base.write(io, """
        <details>
            <summary>
                $(fld.title)
            </summary>
            <p>
    """)
    show(io, mime, fld.content)
    Base.write(io, """
            </p>
        </summary>
    """)
end
```

Zet een NamedTuple om naar een Dict waarbij de *keys* geevalueerd zijn als *variables*

```
ToDict (generic function with 1 method)
```

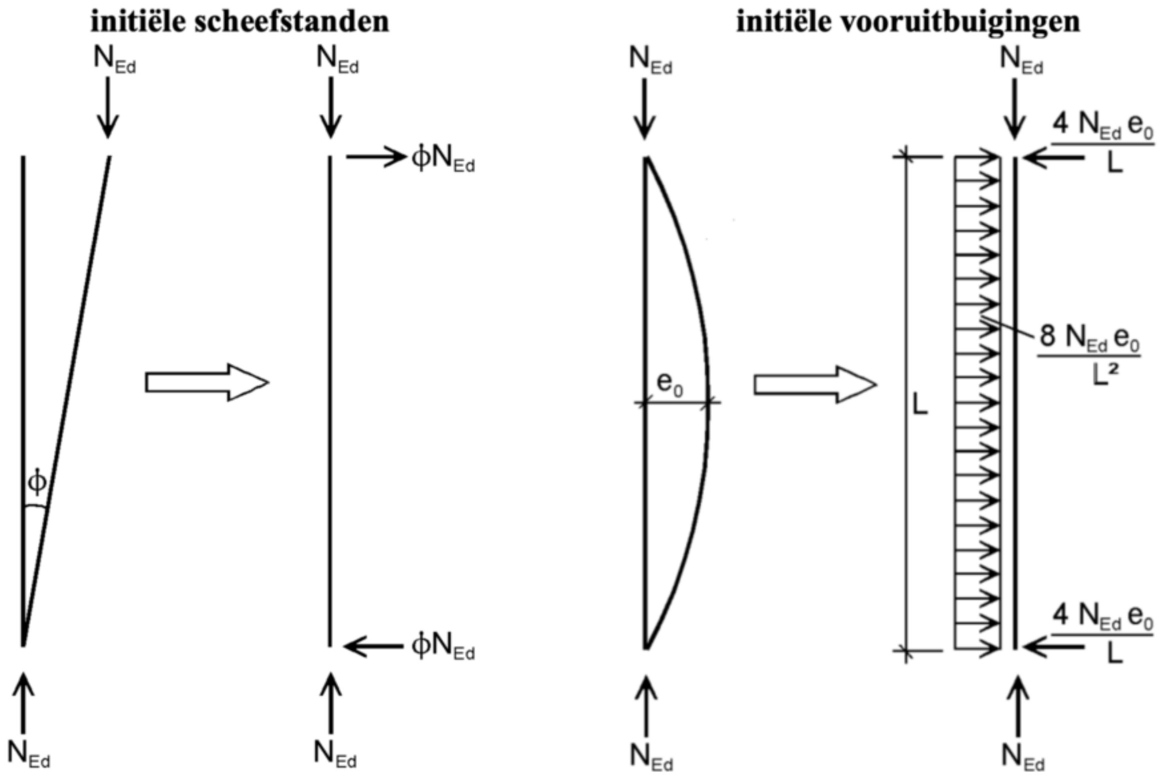
```julia
ToDict(r::NamedTuple) = Dict(keys(r) .|> eval .=> values(r))
```

```
rnd (generic function with 1 method)
```

```julia
rnd(n) = round(n; digits=2)
```

Belastingsschema's

De belastingsschema's worden in het algemeen uitgewerkt en het **superpositie beginstel** wordt gehanteerd om de verschillende belastingseffecten samen te stellen



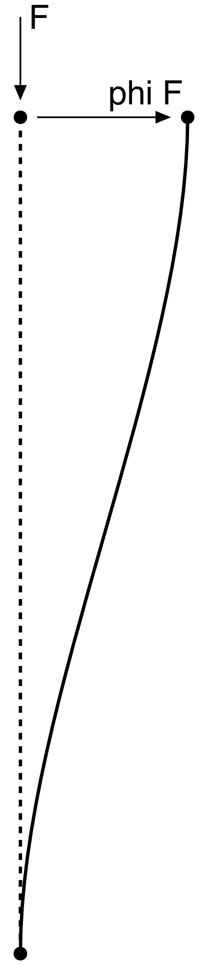

```
figuur5_4
```

Schema 1. Ingeklemde kolom die bovenaan *vast* gehouden wordt - *Initiële vervorming*

Door de *initiële vervorming* en *intitiële vooruitbuiging* worden er krachten opgewekt in de kolom. **Schema 1** begroot de interne krachtswerking onder een **initiële vervorming**. De hoekverdraaiing ϕ wordt vervangen door een **equivalente horizontale puntkracht** ϕF die aangrijpt bovenaan de ligger.

$$(F,\ L,\ M_L,\ \phi,\ A,\ EI,\ I,\ z,\ t)$$

```
F, L, M_L, phi, A, EI, I, z, t = symbols(raw"F L M_L \phi A EI I z t", real=true)
```



`R1H =`

$$F\phi$$

`R1V =`

$$F$$

`R1M =`

$$FL\phi - M_L$$

Bepalen $N(t), V(t)$ en $M(t)$

Bepalen van de interne krachtswerking

`N1 =`

$$F$$

```
N1 = F # Onafhankelijk van t
```

`V1 =`

$$-F\phi$$

```
V1 = - R1H # Onafhankelijk van t
```

`M1_ =`

$$-FL\phi + F\phi t + M_L$$

```
M1_ = - R1M + R1H * t # Onbekende M_L = moment die kolom recht houdt centraal
```

`M1 =`

$$-\frac{FL\phi}{2} + F\phi t$$

Spanning in de doorsnede

`σ1 =`

$$\frac{z\left(-\frac{FL\phi}{2} + F\phi t\right)}{I} + \frac{F}{A}$$

```
σ1 = M1 * z / I + N1 / A
```

Bepalen $\alpha(t)$ en $v(t)$

Bepalen van de hoekverdraaiing α en de vervorming v

$$(C,\ D)$$

`α1_ =`

$$C + \frac{F\phi t^2}{2} + t(-FL\phi + M_L)$$

```
α1_ = integrate(M1_, t) + C1
```

`α1 =`

$$\frac{-\frac{FL\phi t}{2} + \frac{F\phi t^2}{2}}{EI}$$

v1_ =

$$Ct + D + \frac{F\phi t^3}{6} + t^2\left(-\frac{FL\phi}{2} + \frac{M_L}{2}\right)$$

```
v1_ = integrate(α1_, t) + D1
```

v1 =

$$\frac{-\frac{FL\phi t^2}{4} + \frac{F\phi t^3}{6}}{EI}$$

Toepassen kinematische randvoorwaarde

Opleggen van de hoekverdraaiing op $t = L$

rvw1 = $\left[D,\ C,\ C + \frac{FL^2\phi}{2} + L(-FL\phi + M_L)\right]$

```
rvw1 = [
    v1_(t=>0),
    α1_(t=>0),
    α1_(t=>L)
]
```

opl1 = Dict($C \Rightarrow 0$, $D \Rightarrow 0$, $M_L \Rightarrow \frac{FL\phi}{2}$)

Schema 2. Ingeklemde kolom die bovenaan *vast* gehouden wordt - *Initiële vooruitbuiging*

Door de *initiële vervorming* en *intitiële vooruitbuiging* worden er krachten opgewekt in de kolom. **Schema 2** begroot de interne krachtswerking onder een **initiële vooruitbuiging**. De hoekverdraaiing ϕ wordt vervangen door een **equivalente horizontale lijnbelasting** p met een grootte $8Fe_0/L^2$ die aangrijpt over de gehele kolom. Onderaan en bovenaan worden compenserende krachten voorzien.

e_0 =

$$e_0$$

```
e_0 = symbols("e_0", real=true)
```

R2H =

$$\frac{4Fe_0}{L}$$

```
R2H = 8 * F * e_0 / L^2 * L - 4 * F * e_0 / L # Onafhankelijk van t
```

R2V =

$$F$$

```
R2V = F # Onafhankelijk van t
```

R2M =

$$\frac{4Fe_0}{L} - M_L$$

Bepalen $N(t), V(t)$ en $M(t)$

Bepalen van de interne krachtswerking

N2 =

$$F$$

```
N2 = F # Onafhankelijk van t
```

V2 =

$$-\frac{4Fe_0}{L} + \frac{8Fe_0t}{L^2}$$

```
V2 = - R2H + 8 * F * e_0 / L^2 * t # Afhankelijk van t
```

M2_ =

$$\frac{4Fe_0t}{L} - \frac{4Fe_0}{L} - \frac{4Fe_0t^2}{L^2} + M_L$$

```
M2_ = - R2M + R2H * t - 8 * F * e_0 / L^2 * t * t/2 # Afhankelijk van t
```

M2 =

$$\frac{4Fe_0t}{L} - \frac{4Fe_0}{L} - \frac{4Fe_0t^2}{L^2} + \frac{-2FLe_0 + 12Fe_0}{3L}$$

```
M2 = M2_(opl2...)
```

Spanning in de doorsnede

σ2 =

$$\frac{z\left(\frac{4Fe_0t}{L} - \frac{4Fe_0}{L} - \frac{4Fe_0t^2}{L^2} + \frac{-2FLe_0+12Fe_0}{3L}\right)}{I} + \frac{F}{A}$$

```
σ2 = M2 * z / I + N2 / A
```

Bepalen $\alpha(t)$ en $v(t)$

Bepalen van de hoekverdraaiing α en de vervorming v

$(C,\ D)$

α2_ =

$$C + \frac{2Fe_0t^2}{L} - \frac{4Fe_0t^3}{3L^2} + \frac{t(-4Fe_0 + LM_L)}{L}$$

```
α2_ = integrate(M2_, t) + C2
```

α2 =

$$\frac{-\frac{2Fe_0t}{3} + \frac{2Fe_0t^2}{L} - \frac{4Fe_0t^3}{3L^2}}{EI}$$

```
α2 = α2_(opl2...) / EI
```

v2_ =

$$Ct + D + \frac{2Fe_0t^3}{3L} - \frac{Fe_0t^4}{3L^2} + \frac{t^2(-4Fe_0 + LM_L)}{2L}$$

```
v2_ = integrate(α2_, t) + D2
```

v2 =

$$\frac{-\frac{2Fe_0t}{3} + \frac{2Fe_0t^2}{L} - \frac{4Fe_0t^3}{3L^2}}{EI}$$

```
v2 = α2_(opl2...) / EI
```

Toepassen kinematische randvoorwaarde

Opleggen van de hoekverdraaiing op $t = L$

rvw2 = $\left[D,\ C,\ C + \frac{2FLe_0}{3} - 4Fe_0 + LM_L\right]$

```
rvw2 = [
    v2_(t => 0),
    α2_(t => 0),     # Geef hoekverdraaiing op t = 0
    α2_(t => L)      # Geef hoekverdraaiing op t = 0
]
```

opl2 = Dict($C \Rightarrow 0$, $D \Rightarrow 0$, $M_L \Rightarrow \frac{-2FLe_0 + 12Fe_0}{3L}$)

```
opl2 = solve(rvw2, [C2, D2, M_L])
```