

Beregningsprogram for armerte betongskall

Viktor André Å Fjellheim

Bygg- og miljøteknikk (2 årig)

Innlevert: juni 2013

Hovedveileder: Jan Arve Øverli, KT

Norges teknisk-naturvitenskapelige universitet
Institutt for konstruksjonsteknikk



NORGES TEKNISK NATURVITENSKAPELIGE
UNIVERSITET, NTNU
Institutt for konstruksjonsteknikk

Masteroppgave i konstruksjonsteknikk 2013
for
Viktor Fjellheim
Beregning av betongskall
Non-linear cross-sectional analysis of concrete shells
Innledning

Skallkonstruksjoner kan i en del sammenhenger være statisk nødvendig, økonomisk gunstig og/eller arkitektonisk ønskelig sammenlignet med tradisjonelle bæresystemer.

Oppgaven fokuserer på kapasitetskontroll av tverrsnitt påkjent av skallkrefter. Det skal utvikles et brukervennlig dataprogram som beregner kapasiteten. Kapasitetskontrollen utføres med iterasjonsmetoden som er en metode for ikke-lineær tverrsnittsanalyse av skallsnitt.

Oppgave

Oppgaven kan deles opp som følger:

- Et litteraturstudium og teoridel som beskriver metoder for beregning av betongskall.
- Utvikling av et dataprogram for kapasitetskontroll av betongskall.
- Verifikasjon av dataprogram mot kjente løsninger.
- Lage en brukermanual for beregningsprogrammet.

Oppgaven skal være gjennomført innen den 10. juni 2013.

Trondheim den 31.01.2013

Jan Arve Øverli
Førsteamanuensis / Faglærer

Forord

Denne rapporten er en masteroppgave skrevet i avslutningen av sivilingeniørstudiet ved institutt for konstruksjonsteknikk ved Norges Tekniske- Naturvitenskapelige Universitet (NTNU) våren 2013. Arbeidet utgjør 20 ukers arbeid og tilsvarer 30 studiepoeng.

Hensikten med oppgaven er å utvikle et program som beregner kapasitetsutnyttelse i betongskall, og presenterer resultatene slik at brukeren kan følge beregningsstegene bak programmet. I denne rapporten beskrives teorien bak beregningsmetodene, hvordan denne teorien har blitt overført til et programmeringsspråk og verifisering av programmet mot kjente beregninger og eksempler. Samt en brukerinstruks for grensesnittet i programmet.

Oppgaven er skrevet med tanke på at leseren har bakgrunn i konstruksjonsteknikk, men nødvendigvis ikke har kunnskap om beregningsmetoder for betongskall. Jeg ønsker å rette en stor takk til min veileder Jan Arve Øverli ved institutt for konstruksjonsteknikk.

Trondheim, juni 2013

Viktor Fjellheim

Sammendrag

Denne oppgaven tar for seg overføring av iterasjonsmetoden til et beregningsprogram i MATLAB. Iterasjonsmetoden er en numerisk beregningsmetode som beregner kapasitetutnyttelsen i armerte betongskall. Bakgrunnen for iterasjonsmetoden er hentet fra kompendium i Betongkonstruksjoner 3 [1]. Eurokode 2, NSEN1992-1-1 [3], gir de grunnleggende reglene for dimensjonering og material.

Hovedhensikten med oppgaven er å lage et program som regner riktig i henhold til iterasjonsmetoden. Dette betyr at programmet må verifiseres mot kjente resultater. Det er også lagt vekt på å lage et program som er brukervennlig og presenterer resultatene slik at det er mulig å følge iterasjonsmetoden steg for steg.

Det er gjort en gjennomgang av teorien som ligger til grunn for iterasjonsmetoden for å gi et innblikk i bakgrunnen for den, samt gi en bedre forståelse for verifikasjons beregningene. Det er også tatt en gjennomgang av de ulike materialmodellene som er gitt i Eurokode 2, både for armering og betong, da dette har en stor innvirkning på resultatet i beregningene.

For å verifisere at programmet regner riktig, har først resultatene blitt sammenlignet med enkle tilfeller, med bare en belastning omgangen. Det er få muligheter å beregne skallutnyttelse med alle krefter, derfor er en av de viktigste verifikasjonene sammenligning med et annet program som bruker iterasjonsmetoden.

Resultatene i programmet følger resultatene i eksemplene, det finnes avvik, men disse kan forklares med at den numeriskemetoden er en tilnærming og ikke 100% nøyaktig. Den viktigste verifikasjonen er sammenligningen med et annet program. Dette er et komplekst eksempel med alle krefter, det er oppnådd samme resultat i begge programmene, noe som styrker validiteten til programmet.

Abstract

This thesis deals with the implementation of the Iteration method into a computational program in MATLAB. The Iteration method is a numeric method that computes the capacity of reinforced concrete shells. The background for the iteration method comes from compendium for concrete structures 3 [1]. Eurocode 2, NSEN1992-1-1 [3], outlines the basic rules for design and material data.

The primary purpose of the thesis is creating a program that makes calculations that is correct according to the Iteration method. This means that the program must be verified with known results. It have also been emphasized to make a program that is user friendly and presents the results so it would be possible to follow the iteration method step by step.

It have been done a review of the theory behind the iteration method, to give insight in how the program works and give a better understanding of the verification calculations. There have also been done a review of the different material models presented in Eurocode 2, both for rebar and concrete, since this have great impact of the calculation results.

To verify that the program calculates correct, the results have been compared with simple cases, with only one load at the time. There exist few methods to calculate the capacity of shells with all forces applied. Therefor one of the most important verifications is the comparison with another program that utilizes the iteration method.

The results from the program follows the results from the examples, there exists deviations, but these can be explained with the fact that the numerical method is a approximation and not 100% accurate. The most important verification is with the comparison of the other program. This is a complex example with all forces applied, both program have achieved the same results, wich strengthen the validity of the program.

Innhold

1	Innledning	1
2	Teori	2
2.1	Skall krefter	2
2.2	Dimensjonering av Betongskall	4
2.3	Kapasitetskontroll av Betongskall	4
2.4	Utleddning av Iterasjonsmetoden	5
2.5	Iterasjonsmetoden steg for steg	11
2.6	Skjærkraft	13
2.7	Utnyttelse	14
2.8	Material modell	16
3	Programmering	22
3.1	Oppbygging	22
3.1.1	hoved.m	23
3.1.2	steg.m	23
3.1.3	kc.m	24
3.1.4	ks.m	24
3.1.5	betongmatr.m	24
3.1.6	ss.m	25
3.1.7	skjarkontroll.m	25
3.1.8	ur.m	25
3.2	Overføring av teori til kode	25
4	Brukerinstruks	27
4.1	Innstillinger	27
4.2	Resultat	28
5	Verifisering av programmet	30
5.1	Eksempel 1; Trykk	30
5.2	Eksempel 2; Strekk	32
5.3	Eksempel 3; Tverrkontraksjon	34
5.4	Eksempel 4; Moment	36
5.5	Eksempel 5; Skjærkraft	37

5.6 Sammenligning med annet program	37
6 Konklusjon	40
Referanser	41
Vedlegg	42
A Kontrollberegninger	42
A.1 Kontrollberegning Eksempel 1	42
A.2 Kontrollberegning Eksempel 2	43
A.3 Kontrollberegning Eksempel 3	44
A.4 Kontrollberegning Eksempel 4	45
A.5 Kontrollberegning Eksempel 5	45
A.6 Sammenligning med annet program	45
B Filliste	46
C Utdrag fra programmet	47
C.1 hoved.m	47
C.2 steg.m	49
C.3 kc.m	52
C.4 k_s.m	53
C.5 s_s.m	53
C.6 betongmatr.m	56
C.7 ur.m	60
D Symbolliste	61

Figur og tabelliste

Figurer

1	Spenninger i et skall	3
2	Spenningsresultanter i skall	4
3	Bilinær-betong modell [3]	16
4	Idealisert-betong modell [3]	17
5	Ikkelinær-betong modell [3]	18
6	Spennings-tøyningskurver for typisk varmvalset og kaldarbeidet stål. [3] .	19
7	Stål modell [3]	20
8	Løkke for materialmatrise	23
9	Løkke for betongspenning, idealisert materialmodell	24
10	Skjerm bilde av innstillingsgrensesnittet	27
11	Skjerm bilde av resultatgrensesnittet	28
12	Eksempel 1, Trykk	30
13	Eksempel 2, Strekk	32
14	Eksempel 3, Tverrkontraksjon	34
15	Eksempel 4, moment	36
16	Eksempel 5.8.10 fra kompendium TKT4222 [1]	38

Tabeller

1	Eksempel 1; Trykk	31
2	Eksempel 2; Strekk	33
3	Eksempel 3; Tverrkontraksjon	35
4	Eksempel 4; Moment	36
5	Eksempel 4, Innvirkning av antall lag	37
6	Resultat Eksempel 5.8.10	38
7	Filliste	46
8	Symbolliste	63

1 Innledning

Betongskall er konstruksjonselementer som er belastet med både moment og membrankrefter, samtidig som de har liten tykkelse og kan gjerne være kurvet i to retninger. Når da materialet er satt sammen av både betong og armering, gjerne med en ikke-linær materialmodell, kan det være vanskelig eller upraktisk å regne ut utnyttelse for hånd.

Iterasjonsmetoden er en numerisk metode som regner utnyttelsen av betongskall. I denne oppgaven er det vist hvordan man kan implementere denne metoden i MATLAB. I kapittel 2 er bakgrunnen for iterasjonsmetoden beskrevet, basert på Kompendium i Betongkonstruksjoner 3 [1]. Her er også de ulike materialmodellene brukt beskrevet, hentet fra NSEN1992-1-1[3](fra nå EC2).

I kapittel 3 er det beskrevet hvordan teorien har blitt overført til MATLAB's scriptspråk og i kapittel 4 er det en brukerinstruks for det grafiske grensesnittet. For å sjekke at programmet regner riktig er det blit sammenlignet med kjente eksempler i kapittel 5. Vedlagt ligger beregningene for disse eksemplene samt utdrag fra koden, for å vise hvordan den kobles opp mot teorien.

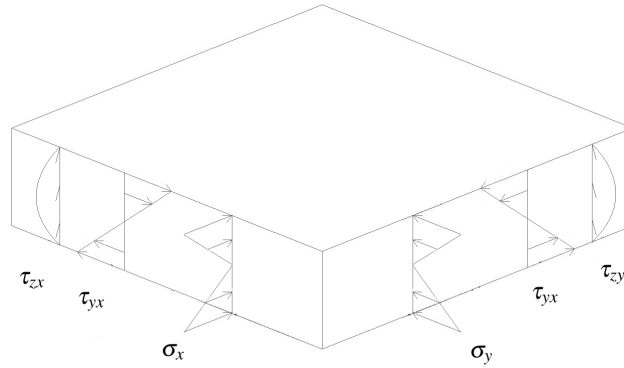
2 Teori

Vi finner skall mange steder, kjente eksempler er eggeskall og vanntanker. Et skall er definert etter geometri og belastning. Skallet har en kurvet inn og utside, separert av en tykkelse t , som er liten i forhold til de totale dimensjonene av skallet. Denne tykkelsen kan være konstant eller variere igjennom skallet. Midtoverflaten er en distanse $t/2$ fra både inn og utsiden. Spenninger i et skall generer membran krefter N , som er krefter per enhet av lengde, målt normalt til skall overflaten. Som i en plate, har skall bøyemoment og torsjonsmoment. Skall kan ta store laster om det er membrankreftene som dominerer. Dette gjør at et skall kan være mye sterkere enn en flat plate med samme volum og materiale, om skallets kurvatur har distribuert lasten slik at det er lite moment i skallet. Disse egenskapene har gjort skallet populært i mange konstruksjoner, som blant annet demninger, siloer, kupler og tanker.

2.1 Skall krefter

Denne oppgaven er begrenset til kapasitetsutnyttelsen av skall, det betyr at kreftene i skallet må være funnet og skallet må være dimensjonert. For å finne kreftene i skall finnes det flere muligheter, den meste brukte metoden for kompliserte konstruksjoner er elementmetoden.

Love-Kirchoff teorien om tynne plater, er en modell som brukes for å finne tøyninger og spenninger i tynne plater belastet med moment og membrankrefter. Teorien har følgende antagelser: skallet er neglisjerbart tynt i forhold til kurvaturen av skallets overflate i midten. Tøyninger og spenninger er små. Rette linjer, som er normalt til midtoverflaten før deformasjon, forblir rette under deformasjon. Spenninger normalt til overflaten i midten av skallet er neglisjerbare.



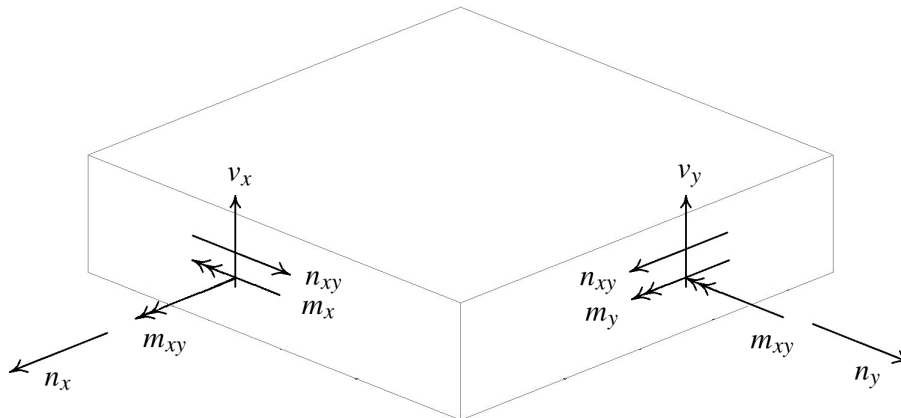
Figur 1: Spenninger i et skall

I figur 1 vises spenningerne i et skall. Fra disse spenningerne får vi med Love-Kirchoff teorien åtte krefter. Det er to bøye momenter (m_x, m_y), et torsjonsmoment (m_{xy}), to skjærkrefter (v_x, v_y) og tre membran krefter (n_x, n_y, n_{xy}). For skall med tykkelse t , i et koordinat system hvor z er normalt til overflaten, x og y er normalt til tverrsnittet, blir spenningsresultantene funnet ved å integrere skallet langs sidene:

$$n_x = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_x dz \quad n_y = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_y dz \quad n_{xy} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{yx} dz$$

$$m_x = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_x z dz \quad m_y = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_y z dz \quad m_{xy} = m_{yx} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_{xy} z dz$$

$$v_x = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{zx} z dz \quad v_y = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{zy} z dz$$



Figur 2: Spenningsresultanter i skall

2.2 Dimensjonering av Betongskall

Selv om oppgaven er begrenset til kapasitetsutnyttelsen, og dimensjoneringen bare brukes som input i metoden vist, kan kapasitetsutnyttelsen brukes til å videre dimensjonere konstruksjonen. Ved å bruke resultatet fra kapasitetsutnyttelsen kan man justere dimensjonene, for å oppnå en optimal utnyttelse av konstruksjonen. Det å dimensjonere armerte skall er en komplisert oppgave, med åtte krefter som skal overføres til langsgående armering i bunn og topp i tillegg til skjærarmering. Samt at de største momentene ikke er på samme plass som de største membran kreftene. Ved å utnytte de metoder for dimensjoneringen som eksisterer, sammen med resultater fra kapasitetsberegninger, er det mulig å lage skall med høy utnyttelse, selv for skall med komplisert last og geometri.

2.3 Kapasitetskontroll av Betongskall

Ved å bruke resultat fra elementmetoden eller andre metoder, og resultat fra dimensjonering kan vi finne kapasitetsutnyttelsen i skallet. Selve utnyttelsen er basert på tøyninger i betong og armering, i forhold til materialets maksimale tøyninger. Metoden som blir brukt i denne oppgaven er iterasjonsmetoden. Dette er en numerisk metode som kan lett tilpasses programmeringsspråk. Metoden er basert på kirchoff's hypotese om linær tøyningfordeling over skallets tykkelse. Det blir da antatt at spenninger ut av planet er null. Det betyr at

skallet kan analyseres som et 2 dimensjonalt problem, men det gjør at iterasjonsmetoden ikke kan brukes for å analysere knutepunkt og andre områder med irregulær geometri.

På basis av spenningsresultanter finner metoden tøyingsfordelingen som gir likevekt mellom ytre og indre krefter. Det brukes en forskyvnings formuleringen som etablerer forholdet mellom de eksterne og interne kreftene. Ved å etablere en stivhetsmatrise, basert på dimensjoneringen, kan tøyninger finnes i midtplanet av skallet. Ved å bruke disse tøyningene, finnes spenninger i betong og armeringen. Ortotropiske materialemodeller, i retning av hovedspenningsretningen, er brukt for å finne tøyning-spenning forholdet. Basert på disse spenningene lages en ny materialmodell og de interne kreftene blir funnet. Denne prosessen gjentas frem til forskjellen mellom de interne og eksterne kreftene er innenfor en akseptabel verdi.

2.4 Utledning av Iterasjonsmetoden

Målet med iterasjonsmetoden er å finne den tilstanden hvor de interne kreftene er lik de eksterne, det betyr at det trengs en tøyingsfordeling som gir likevekt. Forskyvnings formulering:

$$\mathbf{R} = \mathbf{S}(\boldsymbol{\varepsilon}_{t,r}) \quad (2.4.1)$$

$$\mathbf{R} = \begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix} \quad (2.4.2)$$

Hvor \mathbf{R} er de eksterne kreftene og \mathbf{S} er de interne kreftene basert på $\boldsymbol{\varepsilon}_{t,r}$ som er de generelle tøyningene.

$$\boldsymbol{\varepsilon}_t = \begin{bmatrix} \boldsymbol{\varepsilon}_m \\ \boldsymbol{\kappa} \end{bmatrix} = \begin{bmatrix} \varepsilon_{xm} \\ \varepsilon_{ym} \\ \gamma_{xym} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} \quad (2.4.3)$$

Hvor ε_m og κ er tøyningene og kurvaturen av midt planet av skall elementet.

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \boldsymbol{\varepsilon}_m - z\boldsymbol{\kappa} = \mathbf{A}\boldsymbol{\varepsilon}_t = \begin{bmatrix} 1 & 0 & 0 & -z & 0 & 0 \\ 0 & 1 & 0 & 0 & -z & 0 \\ 0 & 0 & 1 & 0 & 0 & -z \end{bmatrix} \begin{bmatrix} \varepsilon_{xm} \\ \varepsilon_{ym} \\ \gamma_{xym} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} \quad (2.4.4)$$

Sammenhengen mellom tøyning og spenning (2.4.1) er ikke linær, for å ta hensyn til dette har vi følgende formulering:

$$\mathbf{R} = \mathbf{K}(\boldsymbol{\varepsilon}_{t,r}) \boldsymbol{\varepsilon}_{t,r+1} \quad (2.4.5)$$

Hvor $\mathbf{R} = \mathbf{K}(\boldsymbol{\varepsilon}_{t,r})$ er sekant stivhetsmatrisen for betong og armering i iterasjon nr r. Ved å benytte prinsippet om virtuelt arbeid, kan stivhetsmatrisen \mathbf{K} etableres. Den generaliserte forskyvningen og rotasjonen blir uttrykt ved forskyvnings vektoren \mathbf{r} :

$$\mathbf{r} = a \begin{bmatrix} \boldsymbol{\varepsilon}_m \\ \boldsymbol{\kappa} \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_{xy} \\ \theta_x \\ \theta_y \\ \theta_{xy} \end{bmatrix} \quad (2.4.6)$$

Hvor a er dimensjonen av skall elementet. Den virtuelle forskyvnings vektor blir:

$$\delta \mathbf{r} = a \delta \boldsymbol{\varepsilon}_t \quad (2.4.7)$$

Eksternt virtuelt arbeid:

$$A_y = \delta r^T a \mathbf{R} \quad (2.4.8)$$

Internt virtuelt arbeid:

$$A_i = \int_v \delta \boldsymbol{\varepsilon}^T dV \quad (2.4.9)$$

Eksternt arbeid = Internt arbeid:

$$\delta \mathbf{r}^T \mathbf{aR} = \int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV \quad (2.4.10)$$

Hvor V er volumet av skall elementet, og \mathbf{aR} er den totale eksterne belastningen. Materialmodellen er gitt på en generell form, det gir at spenninger i planet er gitt som:

$$\boldsymbol{\sigma} = \mathbf{C}(\boldsymbol{\varepsilon})\boldsymbol{\varepsilon} = \mathbf{C}_\varepsilon \boldsymbol{\varepsilon} \quad (2.4.11)$$

Hvor \mathbf{C} er materialmatrisen, den inkluderer både armering og betong.

$$A_i = \int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_V \delta \boldsymbol{\varepsilon}^T \mathbf{C} \boldsymbol{\varepsilon} dV = \int_V \delta \boldsymbol{\varepsilon}_t^T \mathbf{A}^T \mathbf{C} \mathbf{A} \boldsymbol{\varepsilon}_t dV \quad (2.4.12)$$

$$A_y = A_i \rightarrow \mathbf{a}^2 \delta \boldsymbol{\varepsilon}_t^T \mathbf{R} = \mathbf{a}^2 \delta \boldsymbol{\varepsilon}_t^T \int_{-\frac{h}{2}}^{\frac{h}{2}} \mathbf{A}^T \mathbf{C} \mathbf{A} dz \boldsymbol{\varepsilon}_t \quad (2.4.13)$$

Det gir at likevekts ligningen for skall elementet er:

$$\mathbf{R} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \mathbf{A}^T \mathbf{C} \mathbf{A} dx \boldsymbol{\varepsilon}_t = \mathbf{K} \boldsymbol{\varepsilon}_t \quad (2.4.14)$$

Det gir at stivhetsmatrisen \mathbf{K} kan bli uttrykt med integralet:

$$\mathbf{K} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \mathbf{A}^T \mathbf{C} \mathbf{A} dz \quad (2.4.15)$$

Ved kongurens multiplikasjon av integranden blir stivhetsmatrisen:

$$\mathbf{K} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \begin{bmatrix} \mathbf{C} & -z\mathbf{C} \\ -z\mathbf{C} & z^2\mathbf{C} \end{bmatrix} dz \quad (2.4.16)$$

Med en tilhørende materialmatrise \mathbf{C} , kan tøyninger og kurvatur i skallets midtplan bli funnet fra likevekts ligningen.

$$\boldsymbol{\varepsilon}_t = \mathbf{K}^{-1} \mathbf{R} \quad (2.4.17)$$

\mathbf{K} er uttrykket som en generell funksjon av avstanden z fra midten av skall planet. For å løse integralet blir tverrsnittet delt opp i lag. Med n lag, lagtykkelsen er gitt ved $\Delta = h/n$,

hvor h er tykkelsen av tverrsnittet. Separat fra betongen, blir armeringen delt opp i lag, med avstand z fra midtplanet. Avstanden z er avhenging av overdekningen for tverrsnittet. For å få den totale materialmatrisen \mathbf{C} , summeres alle lagene for betongstivhetsmatrisene \mathbf{K}_c og armeringstivhetsmatrisene \mathbf{K}_s . Denne operasjonen gjøres med antagelse at armering bare er lagt i x og y retning. Dette gir følgende uttrykk for stivhetsmatrisen:

$$\text{Betong: } \mathbf{K}_c = \sum_{i=1}^n \Delta h \cdot \mathbf{A}_i^T \mathbf{C}_i \mathbf{A}_i = \Delta h \sum_{i=1}^n \begin{bmatrix} \mathbf{C}_i & -z_i \mathbf{C}_i \\ -z_i \mathbf{C}_i & z_i^2 \mathbf{C}_i \end{bmatrix} \quad (2.4.18)$$

$$\text{Armering: } \mathbf{K}_s = \sum_{j=1}^m (\mathbf{A}_{sxj} \begin{bmatrix} \mathbf{C}_{sxj} & -z_j \mathbf{C}_{sxj} \\ -z_j \mathbf{C}_{sxj} & z_j^2 \mathbf{C}_{sxj} \end{bmatrix} + \mathbf{A}_{syj} \begin{bmatrix} \mathbf{C}_{syj} & -z_j \mathbf{C}_{syj} \\ -z_j \mathbf{C}_{syj} & z_j^2 \mathbf{C}_{syj} \end{bmatrix}) \quad (2.4.19)$$

$$\mathbf{K} = +\mathbf{K}_c + \mathbf{K}_s \quad (2.4.20)$$

Tilsvarende som de eksterne kreftene, blir de interne spenningsresultantene samlet i en vektor \mathbf{S} :

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_N \\ \mathbf{S}_M \end{bmatrix} \begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix} \quad (2.4.21)$$

Fra spenninger i betong og armering, blir de interne spenningsresultantene funnet ved å integrere:

$$\mathbf{S}_N = \int_{-\frac{h}{2}}^{\frac{h}{2}} \boldsymbol{\sigma} dz \quad (2.4.22)$$

$$\mathbf{S}_M = \int_{-\frac{h}{2}}^{\frac{h}{2}} z \boldsymbol{\sigma} dz \quad (2.4.23)$$

Som for stivhetsmatrisen, er spenningene i armering og betong delt i lag, ved å summere alle lagene blir de interne spenningsresultantene funnet:

$$\mathbf{S}_N = \sum_{i=1}^n h\Delta\sigma_{ci} + \sum_{j=1}^m \begin{bmatrix} A_{sxj}\sigma_{sxj} \\ A_{syj}\sigma_{syj} \\ 0 \end{bmatrix} \quad (2.4.24)$$

$$\mathbf{S}_M = \sum_{i=1}^n h\Delta(z - \sigma_{ci}) + \sum_{j=1}^m \begin{bmatrix} -zA_{sxj}\sigma_{sxj} \\ -zA_{syj}\sigma_{syj} \\ 0 \end{bmatrix} \quad (2.4.25)$$

Når spenningene i betongen og armeringen er funnet, blir det da mulig å generere en ny materialmatrise \mathbf{C} . For et ortotropisk linært elastisk materiale, gir Hookes lov i plan spenning:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}; \quad \mathbf{C} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (2.4.26)$$

Hvor ν er poisson's ratio. For ikke lineære materiale som betong, er ikke Hooks lov gyldig. Oppsprekking i betongen på strekk siden, og ikke linær oppførsel på trykk siden, kan bli tatt hensyn til ved å bruke en ortotropisk material modell i retning av hovedspenningene:

$$\boldsymbol{\sigma}_p = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \mathbf{C}_p \boldsymbol{\varepsilon}_p = \frac{1}{1-\nu^2} \begin{bmatrix} E_{11} & \nu E_{12} & 0 \\ \nu E_{12} & E_{22} & 0 \\ 0 & 0 & \frac{(1-\nu)E_{12}}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} \quad (2.4.27)$$

For å finne hovedspenningsretningene må tøyninger og spenninger transformers. Transformasjon fra de globale aksene, x og y , er definert ved:

$$\boldsymbol{\varepsilon}_p = \mathbf{T}(\theta)\boldsymbol{\varepsilon} \quad (2.4.28)$$

Hvor $\boldsymbol{\varepsilon}_p$ er hovedspennningene, θ er vinkel for hoved tøyningens retnignen og $\mathbf{T}(\theta)$ er transformasjons matrisen.

$$\mathbf{T}(\theta) = \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & \sin \theta \cos \theta \\ \sin^2 \theta & \cos^2 \theta & -\sin \theta \cos \theta \\ -2 \sin \theta \cos \theta & 2 \sin \theta \cos \theta & \cos^2 \theta - \sin^2 \theta \end{bmatrix} \quad (2.4.29)$$

Vinkel for hovedtøyningene er gitt ved:

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{\gamma_{xy}}{\epsilon_x - \epsilon_y} \right) \quad (2.4.30)$$

Det blir antatt at hovedtøyninger og hovedspenninger har samme akse. Det gjør det mulig å transformere hovedspenningene og materialmatrisen til globale xy -koordinater:

$$\sigma_c = \mathbf{T}^T(\theta) \sigma_p = \mathbf{T}^T(\theta) \mathbf{C}_p \epsilon_p = \mathbf{T}^T(\theta) \mathbf{C}_p \mathbf{T}(\theta) \epsilon \quad (2.4.31)$$

$$\mathbf{C} = \mathbf{T}^T \mathbf{C}_p \mathbf{T} \quad (2.4.32)$$

Det blir antatt at armeringen ligger i samme retning som de globale x og y retningene, da er spenning-tøynings sammenhengen for et lag med armering gitt som:

$$\sigma_s = \mathbf{C}_s \epsilon \quad (2.4.33)$$

$$\sigma_s = \begin{bmatrix} \sigma_{sx} \\ \sigma_{sy} \\ \tau_{sxy} \end{bmatrix} = \begin{bmatrix} E_{sx} & 0 & 0 \\ 0 & E_{sy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (2.4.34)$$

Hvor E_{sx} og E_{sy} er sekant modulus for armeringen i den globale retningen, ligning (2.4.34) representerer et to dimensjonalt spenning-tøynings forhold.

Det må defineres et konvergens kriterie for å bestemme når vi er innenfor en akseptabel forskjell mellom interne og eksterne krefter. En måte å gjøre det på er å bruke den relative forskjellen mellom hvert av de seks eksterne og interne spenningsresultantene. Iterasjonen stopper da når forskjellen er innenfor et akseptabelt nivå, β , eller overskredet grensen for maksimum antall iterasjoner.

$$\text{Konvergens om: } \left| \frac{R_k - S_{0,k}}{R_k} \right| < \beta; \quad k = 1, 2, \dots, 6 \quad (2.4.35)$$

Avhengig av oppgaven brukes det forskjellige verdier av β , normal verdi ligger rundt 0,01, men kan i visse tilfeller være mindre for å oppnå ønsket nøyaktighet. For å unngå divergens i løsningen, er det også en grense for antall iterasjoner.

2.5 Iterasjonsmetoden steg for steg

For å få oversikt over metoden, er det her vist iterasjonsmetoden steg for steg. Det er også denne fremgangsmåten som vil bli implementert i MATLAB. Det vil da være mulig å kunne følge disse stegene i programmet.

1. Bestemer den eksterne last vektoren \mathbf{R} , normalt fra FEM analyse, bruker armering og betong gitt fra dimensjonering.
2. Kalkulerer initiell stivhetsmatrise \mathbf{K}_0 antar isotropisk elastisk oppførsel for betong og linær elastisk oppførsel for armeringen:

Betong:

$$\mathbf{K}_{c0} = \sum_{j=1}^m \Delta h \mathbf{A}_i^T \mathbf{C}_{0i} \mathbf{A}_i = \Delta h \sum_{i=1}^n \begin{bmatrix} \mathbf{C}_{0i} & -z_i \mathbf{C}_{0i} \\ -z_i \mathbf{C}_{0i} & z_i^2 \mathbf{C}_{0i} \end{bmatrix} \quad (2.5.1)$$

Armering:

$$\mathbf{K}_{s0} = \sum_{j=1}^m (A_{sxj} \begin{bmatrix} \mathbf{C}_{0sxj} & -z_j \mathbf{C}_{0sxj} \\ -z_j \mathbf{C}_{0sxj} & z_j^2 \mathbf{C}_{0sxj} \end{bmatrix} + A_{syj} \begin{bmatrix} \mathbf{C}_{0syj} & -z_j \mathbf{C}_{0syj} \\ -z_j \mathbf{C}_{0syj} & z_j^2 \mathbf{C}_{0syj} \end{bmatrix}) \quad (2.5.2)$$

$$\mathbf{K}_0 = \mathbf{K}_{c0} + \mathbf{K}_{s0} \quad (2.5.3)$$

3. Finner tøyninger og kurvatur i midten skall planet:

$$\boldsymbol{\varepsilon}_{t0} = \mathbf{K}_0^{-1} \mathbf{R} \quad (2.5.4)$$

4. Finner tøyningene i planet i hvert betong og armerings lag:

$$\boldsymbol{\varepsilon}_{0i} = \mathbf{A}_i \boldsymbol{\varepsilon}_{t0} \quad (2.5.5)$$

5. Finner hovedtøyninger og hovedretninger i betong lagene:

$$\boldsymbol{\varepsilon}_{p0i} = \mathbf{T}_{\varepsilon i}(\boldsymbol{\theta}_i) \boldsymbol{\varepsilon}_{0i} \quad (2.5.6)$$

Hvor $\mathbf{T}_{\varepsilon i}$ er transformasjons matrisen fra globale akser til den lokale hoved retningen.

Vinkel θ_i er funnet fra:

$$\theta_i = \frac{1}{2} \tan^{-1} \left(\frac{\gamma_{xy}^i}{\epsilon_x^i - \epsilon_y^i} \right) \quad (2.5.7)$$

6. Kalkuler hovedspenningene i hvert betong lag. Hovedspenningen σ_{p0i} er avhengig av spenning-tøynings forholdet. Dette er hvor de ikke-linære effektene i betong er tatt hensyn til.

7. Transformer hovedspenningene i hvert betonglag til det globale xy-koordinat systemet.

$$\sigma_{c0i} = \mathbf{T}_{\epsilon i}^T(\theta_i) \sigma_{p0i} \quad (2.5.8)$$

$$\mathbf{T}_{\epsilon}^T = \mathbf{T}_{\sigma}^{-1} \text{ pga ortogonalitet} \quad (2.5.9)$$

8. Finner armerings spenninger i hvert armerings lag:

$$\sigma_{s0j} = \mathbf{C}_{s0j} \epsilon_{0j} \quad (2.5.10)$$

9. Finner den nye sekant modulusen for alle betong og armerings lag.

10. Kalkuler ny material matrise for betong basert på sekantmodulusen: $\mathbf{C}_{pli}; i = 1, \dots, n$

11. Transformer den lokale material matrisen til de globale xy-aksene:

$$\mathbf{C}_{li} = \mathbf{T}_{\epsilon i}^T \mathbf{C}_{pli} \mathbf{T}_{\epsilon i} \quad (2.5.11)$$

12. Kalkuler den interne kraft vektoren som en summering av betong og armerings bidrag:

$$\mathbf{S}_0 = \Delta h \sum_{i=1}^n \begin{bmatrix} \sigma_{c0i} \\ -z_i \sigma_{c0i} \end{bmatrix} + \sum_{j=1}^m \begin{bmatrix} \mathbf{A}_{sxj} \sigma_{sx0} \\ \mathbf{A}_{syj} \sigma_{sy0} \\ 0 \\ -z_j \mathbf{A}_{sxj} \sigma_{sx0} \\ -z_j \mathbf{A}_{syj} \sigma_{sy0} \\ 0 \end{bmatrix} \quad (2.5.12)$$

13. Finner maksimum relativ forskjell mellom komponentene av eksterne og interne

spenningsresultanter:

$$\text{Maxdiff} = \left| \frac{R_k - S_{0k}}{R_k} \right|; k = 1, 2, \dots, 6 \quad (2.5.13)$$

14. Sjekk konvergens i henhold til den valgte toleransen, for eksempel $\beta = 0,01$

- Om $\text{Maxdiff} \leq \beta$, er likevekt oppnådd og kalkulasjonen kan avsluttes.
- Om $\text{Maxdiff} > \beta$, er det ikke konvergens og kalkulasjonen må fortsette med en ny iterasjon

2.6 Skjærkraft

I den foregående delen ble ikke skjærkrefter tatt hensyn til. Det er heller en separat kontroll for dem, etter at iterasjonsmetoden har kjørt ferdig. Skallet blir da sett på som en bjelkestripe i en generell retning. Bjelkestripen er belastet med krefter V_α , N_α og M_α . Krefter vinkelrett på bjelkestripen er ikke tatt hensyn til. Kraftene må transformeres fra skallretning til bjelkestriperetningen α :

$$V_\alpha = V_x \cos \alpha + V_y \sin \alpha \quad (2.6.1)$$

$$N_\alpha = N_x \cos^2 \alpha + N_y \sin^2 \alpha + 2N_{xy} \sin \alpha \cos \alpha \quad (2.6.2)$$

$$M_\alpha = M_x \cos^2 \alpha + M_y \sin^2 \alpha + 2M_{xy} \sin \alpha \cos \alpha \quad (2.6.3)$$

$$A_{s\alpha} = A_{sx} \cos^2 \alpha + A_{sy} \sin^2 \alpha \quad (2.6.4)$$

Hvor $A_{s\alpha}$ er det effektive tverrsnittet av den langsgående armeringen i α -retningen. For å sjekke at skjærarmeringen er nok for hele elementet må det kontrolleres med varierende α fra 0° til 180° . Steg med 5° er vanligvis nok.

For å finne utnyttelse må den skjærkraften som er funnet, sammenlignes med en kapasitet. Dette er dekket i EC2 6.2 [3], hvor det er en kapasitet for tverrsnitt som trenger skjærarmering og en for tverrsnitt som ikke trenger skjærarmering.

For deler uten skjærarmering:

$$V_{Rd,c} = [C_{Rd,c} k (100 \rho_1 f_{ck})^{1/3} + k_1 \sigma_{cp}] b_w d \quad (2.6.5)$$

Med minste verdi:

$$V_{Rd,c} = (v_{min} + k_1 \sigma_{cp}) b_w d \quad (2.6.6)$$

Hvor:

$$k = 1 + \sqrt{\frac{200}{d}} \leq 2.0 \quad \text{med } d \text{ i mm}$$

$$\rho_1 = \frac{A_{sl}}{b_w d} \leq 0.02$$

A_{sl} er tverrsnittsarealet av lengdearmeringen.

b_w er bredden av tverrsnittet.

$$\sigma_{cp} = N_{ed}/A_c < 0.2 f_{cd}$$

A_c er arealet av betongtverrsnittet.

$$v_{min} = 0.035 k^{3/2} f_{ck}^{1/2}$$

For deler med skjærarmering:

$$V_{Rd,s} = \frac{A_{sw}}{s} z f_{ywd} \cot \theta \quad (2.6.7)$$

Med maks verdi:

$$V_{Rd,max} = \alpha_{cw} b_w z v_1 f_{cd} / (\cot \theta + \tan \theta) \quad (2.6.8)$$

Hvor:

A_{sw} er skjærarmerings tverrsnittsareal

s er senteravstanden mellom bøyler

f_{ywd} er flytespenningen for skjærarmeringen.

v_1 er en fasthetsreduksjonsfaktor for betong opprisset på grunn av skjærkraft

α_{cw} er en koeffisient som tar hensyn til spenningstilstanden i trykkgurten

θ er vinkel for trykkstaverl

2.7 Utnyttelse

For å uttrykke kapasiteten av skallet er det valgt å benytte utnyttelse ratioer. Med disse ratioene er det mulig å sjekke at de dimensjonerende lastene ikke overgår dimensjonerende motstand. Utnyttelsesratioene er basert på tøyningene i tverrsnittet mot maks tøyninger tillatt for det valgte materialet. For betong er denne grensen gitt i EC2 ved ϵ_{cu} som er grensetøyningen. For armering er grensen i EC2 rundt 5% tøyning, men det kan heller

være gunstig å bare vise selve armeringstøyingene, siden grensen er relativt høy. For betong er utnyttelse ratioen definert som:

$$UR_c = \frac{\epsilon_c}{\epsilon_{cu}} 100\% \quad (2.7.1)$$

Hvor ϵ_c er maksimum trykk hoved tøyning og ϵ_{cu} er maksimal trykk tøyning. Utnyttelse ratioen for armering er definert som:

$$UR_s = \frac{\epsilon_s}{\epsilon_{ud}} 100\% \quad (2.7.2)$$

Hvor ϵ_s er tøyningen i armeringen, og ϵ_{ud} er tøyningens grensen for armeringen. Utnyttelse ratioen for tverrskjær er basert på forholdet mellom dimensjonerende skjærkraft og skjærmotstand:

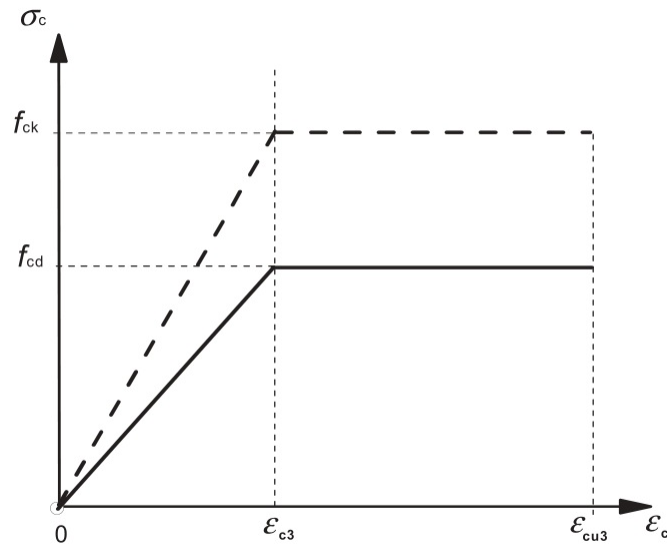
$$UR_c = \frac{V_\alpha}{V_{Rd,c}} 100\% \quad (2.7.3)$$

For komponenter som trenger skjærarmering:

$$UR_s = \frac{V_\alpha}{V_{Rd,s}} 100\% \quad (2.7.4)$$

2.8 Material modell

For å kunne finne sammenhengen mellom σ og ϵ trenger vi en modell som gir materialets spenning-tøyningforløp. Den virkelige oppførselen til betong følger en ikke-linær kurve, men EC2 tillater forenklete spenning-tøyningsforløp. I standarden er det presentert tre forskjellige materialmodeller for betong, henholdsvis, bilinær(Figur 3), idealisert(Figur 4) og ikke-linær(Figur 5). Alle material verdier er gitt i tabell 3.1 i EC2[3].



Figur 3: Bilinear-betong modell [3]

EC2, 3.1.7(2) [3] andre forenklete spennings-tøyningssammenhenger kan benyttes hvis de tilsvarer, eller er mer konservative enn det som er definert i 3.1.7(1)[3].

$$\sigma_c = f_{cd} E_{cm} \quad \text{for} \quad 0 \leq \epsilon_c \leq \epsilon_{cu3} \quad (2.8.1)$$

$$\sigma_c = f_{cd} \quad \text{for} \quad \epsilon_{c3} \leq \epsilon_c \leq \epsilon_{cu3} \quad (2.8.2)$$

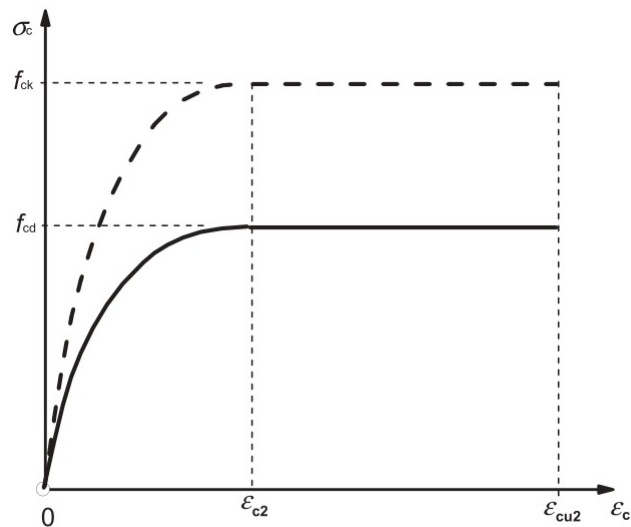
hvor:

f_{cd} er dimensjonerende trykkfasthet

E_{cm} er elastisitsmodulen for betong

ϵ_{c3} er tøyningene idet maksimalspenningen nås

ϵ_{cu3} er tøyningsgrensen i bruddgrensetilstanden



Figur 4: Idealisert-betong modell [3]

EC2, 3.1.7(1)[3] angir spenning-tøyningssammenhengen i trykk som kan benyttes for kapasitetsberegninger i bruddgrensetilstand.

$$\sigma_c = f_{cd} \left[1 - \left(1 - \frac{\epsilon_c}{\epsilon_{c2}} \right)^n \right] \quad \text{for } 0 \leq \epsilon_c \leq \epsilon_{c2} \quad (2.8.3)$$

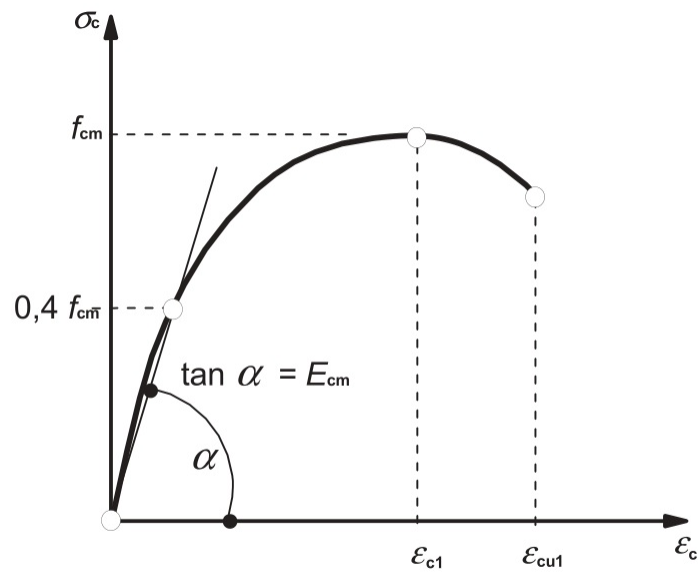
$$\sigma_c = f_{cd} \quad \text{for } \epsilon_{c2} \leq \epsilon_c \leq \epsilon_{cu2} \quad (2.8.4)$$

hvor:

f_{cd} er dimensjonerende trykkfasthet

ϵ_{c2} er tøyningene idet maksimalspenningen nås

ϵ_{cu2} er tøyningsgrensen i bruddgrensetilstanden



Figur 5: Ikkelinær-betong modell [3]

EC2, 3.1.5(1)[3] gir spenning-tøyningssammenhengen for ikke-linær konstruksjonsanalyse.

$$\frac{\sigma_c}{f_{cm}} = \frac{k\eta - \eta^2}{1 + (k-2)\eta} \quad \text{for } 0 \leq \epsilon_c \leq \epsilon_{cu1} \quad (2.8.5)$$

$$\eta = \epsilon_c / \epsilon_{c1} \quad (2.8.6)$$

$$k = 1.05 \cdot E_{cm} \cdot |\epsilon_{c1}| / f_{cm} \quad (2.8.7)$$

hvor:

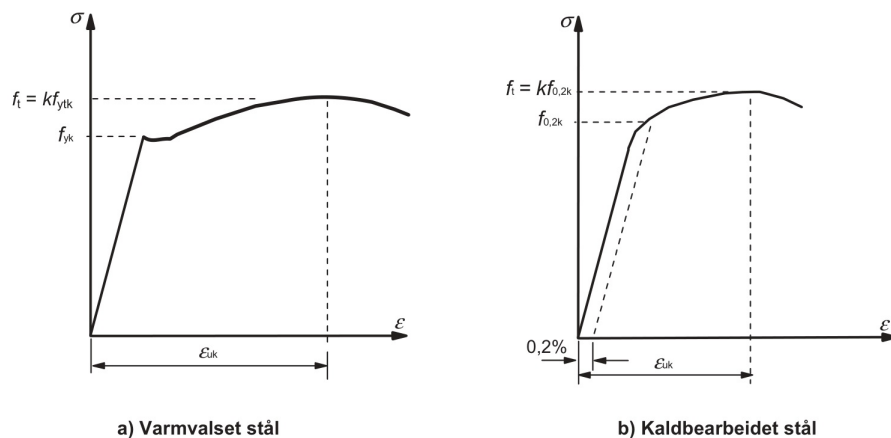
- f_{cm} er middelverdien av trykkfastheten
- E_{cm} er elastisitsmodulen til betongen
- ε_{c1} er tøyningene idet maksimalspenningen nås
- ε_{cu1} er tøyningsgrensen i bruddgrensetilstand

Bøyestrekfastheten til betong er gitt i EC2, 3.1.8(1) [3] og er som følger:

$$f_{ctm,fl} = \max[(1.6 - h/1000)f_{ctm}; f_{ctm}] \quad (2.8.8)$$

hvor:

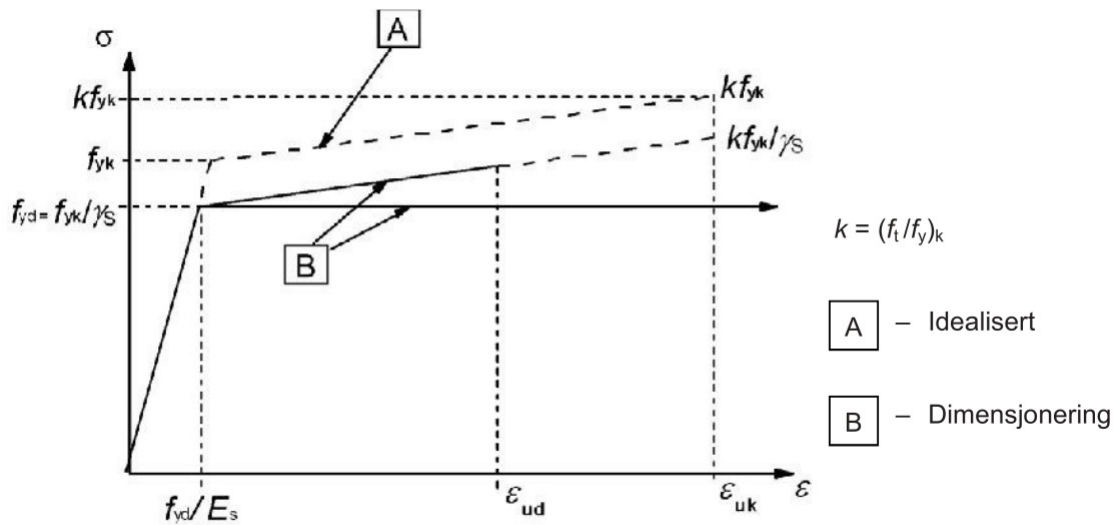
- h er konstruksjonsdelens totale tykkelse
- f_{ctm} er midlere aksialstrekfasthet i henhold til tabell 3.1[3]



Figur 6: Spenning-tøyningsskurver for typisk varmvalset og kaldarbeidet stål. [3]

For armeringsstål trenger vi også en passende materialmodell som gir en tilfredsstillende spenning-tøyningssammenheng, som betong har armeringsstål i virkeligheten en ikke-linær

kurve (Figur 6). EC2 gir to forenklete modeller for dimensjonering (Figur 7), en modell med stigende øvre del og en med en horisontal øvre del.



Figur 7: Stål modell [3]

EC2, 3.2.7[3] gir forutsetninger for dimensjonering ved bruk av den flate modellen.

$$\sigma_s = \varepsilon_s E_s \quad \text{for} \quad 0 \leq \varepsilon_s \leq \varepsilon_{flyt} \quad (2.8.9)$$

$$\sigma_s = f_{yd} \quad \text{for} \quad \varepsilon_{flyt} < \varepsilon_{uk} \quad (2.8.10)$$

hvor:

$\varepsilon_{flyt} = f_{yd} / E_s$ tøyning ved flyting

E_s elastisetsmodul for armeringsstål

f_{yd} flytespenning for armeringsstål

ε_{uk} er tøyningsgrensen i bruddgrensetilstand

EC2, 3.2.7[3] gir forutsetninger for dimensjonering ved bruk av den stigende modellen.

$$\sigma_s = \varepsilon_s E_s \quad \text{for} \quad 0 \leq \varepsilon_s \leq \varepsilon_{flyt} \quad (2.8.11)$$

$$\sigma_s = f_{yd} k \left(\frac{\varepsilon_s}{\varepsilon_{ud}} \right) \quad \text{for} \quad \varepsilon_{flyt} < \varepsilon_s < \varepsilon_{ud} \quad (2.8.12)$$

hvor:

k faktor for største spenning gitt i EC2, Tabell NA 3.5[3]

ε_{ud} er tøyningsgrensen i bruddgrensetilstand

3 Programmering

Programmet er blitt skrevet i MATLAB R2012b, grunnen til dette er at MATLAB er tilrettelagt for numeriske simuleringer og dermed passer bra for å integrere iterasjonsmetoden i kode. MATLAB har også en rekke fleksible plotte funksjoner som gjør det enkelt å presentere matematiske resultater, samt en egen applikasjon for å generere det grafiske grensesnittet for brukeren. Dette gjør at MATLAB fremstår som et bedre valg for denne oppgaven enn et mer generelt programmeringsspråk som PHYTON eller JAVA.

Når programmet ble laget, var det ikke tatt hensyn til å lage en optimal kode, men en robust kode som vil fungere uansett. Med dagens datakraft, vil ikke en mer optimal kode være merkbart raskere (sammenlign Eksempel 1, i kapittel 5). Det er også tatt hensyn til at koden kanskje vil bli videreført av andre, derfor er det en del kommentarer, og det er blitt lagt vekt på å skrive koden så tydelig som mulig og merke alle variabler logisk. Den delen av koden som omhandler presentasjonen av resultatene er laget slik at resultatene skal lett kunne knyttes opp mot teorien i kapittel 2.

3.1 Oppbygging

Koden er bygget opp rundt to hovedfiler som kjører det grafiske grensesnittet, henholdsvis en del for innstillingene og en for resultatene. Dette er laget i ”GUIDE”, som er en applikasjon i MATLAB, hvor den nødvendige koden for det grafiske grensesnittet genereres. De forskjellige komponentene er knyttet til ulike funksjoner som kjører iterasjons metoden med de variablene som er valgt av brukeren i grensesnittet. Selve funksjonen for iterasjons metoden er delt inn i flere underfunksjoner. Hoved funksjonen definerer alle variabler og konstanter som trengs, deretter er det en underfunksjon som kjører alle iterasjonene til det er konvergens eller frem til maks antall iterasjoner er kjørt. I iterasjonsfunksjonen er det også flere under funksjoner for stivhetsmatrisene, spenning-tøyningssammenhengene og utnyttelsesratioer. De to grensesnitt filene er beskrevet i detalj, i kapittel 4.


```
for i = 0:1:n-1;
    C(1,3*i+1)=(1/(1-v(i+1)^2))*E11;
    C(1,3*i+2)=(1/(1-v(i+1)^2))*v(i+1)*E12;
    C(2,3*i+1)=(1/(1-v(i+1)^2))*v(i+1)*E12;
    C(2,3*i+2)=(1/(1-v(i+1)^2))*E22;
    C(3,3*i+3)=(1/(1-v(i+1)^2))*((1-v(i+1))*E12)/2;
end
```

Figur 8: Løkke for materialmatrise

3.1.1 hoved.m

Filen som styrer iterasjonsmetoden er hoved.m, her blir alle konstanter deklartert basert på input fra grensesnittet. Her blir også de initielle materialmatrisene laget (Figur 8) og matriser for plotting. Matrisene for plotting lagrer all informasjon for en gitt komponent over alle iterasjonene. Denne filen kjører steg.m, ur.m og skjarkontroll.m når iterasjonene er ferdig.

Det er mulig å sette på timer av iterasjonsfunksjonen, ved å fjerne % tengne forran *tic* [9] og *toc* i koden. Da vil tid brukt av utregningen komme opp i MATLAB's kommando vindu, hver gang funksjonen kjøres.

3.1.2 steg.m

Dette er funksjonen som kjører iterasjonsmetoden steg for steg frem til enten konvergens kriteriet er oppfylt eller maks antall iterasjoner er kjørt. Den kjører først kc.m og k_s.m for deretter å finne ϵ_0 . Tøyningene for hvert betonglag, hovedtøynings vinkel og tilhørende hovedtøyninger blir funnet igjennom en løkke som går igjennom alle betonglagene. Deretter blir betongmatr.m kjørt. Med resultater fra betongmatr.m kjøres en løkke for å finne poisson's tall for hvert betong lag og deretter blir ny materialmatrise for betongen funnet.

3.1.3 kc.m

Denne funksjonen bygger opp betongstivhetsmatrisen(2.5.1) basert på materialmatrisen funnet i steg.m.

3.1.4 ks.m

Denne funksjonen bygger opp armeringstivhetsmatrisen(2.5.2) basert på materialmatrisen funnet i steg.m.

3.1.5 betongmatr.m

```

if epsilon_p0i(1,i) < 0
    sigma_p0i(1,i) = -f_cd*(1-(1-epsilon_p0i(1,i)/epsilon_c3));
if epsilon_p0i(1,i) <= epsilon_c3;
    sigma_p0i(1,i) = -f_cd;
    if epsilon_p0i(1,i) <= epsilon_cu3
        sigma_p0i(1,i) = 0;
    end
end
else
    sigma_p0i(1,i) = 0;
end

```

Figur 9: Løkke for betongspenning, idealisert materialmodell

Denne funksjonen bestemmer betongspenningen avhengig av hvilken materialmodell (2.8) som er valgt. Figur 9 viser hvordan løkken går igjennom hvert betonglag og bestemmer spenningen basert på hovedtøyningene. Etter at spenningene er funnet blir det bygget opp en ny elastisetsmodul for hvert betonglag.

$$E_{kki} = \frac{\sigma_{kp0i}}{\epsilon_{kp0i}} \quad \text{for } k = 1, 2$$

$$E_{12} = \frac{E_{11} + E_{22}}{2}$$

3.1.6 ss.m

Denne funksjonen bestemmer armeringsspenningene basert på tøyningene og deretter finner ny materialmatrise for armeringen. Basert på spenningene finner den bidraget til de interne kreftene fra armeringen.

3.1.7 skjarkontroll.m

Etter at iterasjonsmetoden har kjørt, finnes her skjærkapasiteten og utnyttelsen.

3.1.8 ur.m

Finner utnyttelse for betongen basert på hvilken materialmodell som er benyttet.

3.2 Overføring av teori til kode

For det meste av iterasjonsmetoden er det ganske rett frem å overføre teorien til kode, siden iterasjonsmetoden er tilrettelagt nettopp for dette, så i det neste avsnittet vil det bare bli tatt opp et par tilfeller som havner utenfor det normale. For et mer detaljert innblikk i koden ligger det utdrag i vedlegget, evt for å få fullt innsyn må programmet åpnes i MATLAB.

Et av de første problemene i koden er inverteringen av stivhetsmatrisen (2.4.17), på grunn av at vi har en to delt stivhetsmatrise med både armering og betong kan vi ende opp med en matrise som ikke kan inverteres på vanlig måte. Dette oppstår når deler eller hele betongen er opprisset, en løsning vil være å redusere størrelsen på matrisen, men dette er tungvindt og vil kreve veldig mye kode. En mye lettere løsning er å benytte matlabs innebygde funksjon for pseudoinvertering [6]. Før inverteringen skjer sjekker matlab hvor sensitiv matrisen er for feil vha den innebygde funksjonen `cond`[7], avhengig om den er sensitiv eller ikke vil enten vanlig invertering eller pseudoinvertering kjøre.

I hoved.m blir E_{cm} definert avhengig av hvilken materialmodell som er valgt. For den bilinære og idealiserte modellen er dette rett frem hvor den er funnet via ϵ_{c3} og ϵ_{c2} . Men for den ikke-linære modellen oppgir EC2 i Fig 3.2 [3] at den er tilnærmet funnet ved å bruke $0.4f_{cm}$. Siden kurven er ikke-linær er det vanskelig å ta ut denne verdien direkte, et alternativ kunne vært å bruke tabullerte verdier, men dette gir ikke mulighet for at brukeren kan legge inn en egen fasthetsklasse. Så for å finne E_{cm} brukes Newton-Rhapson

iterasjoner [5]. Se vedlegg C for detaljer.

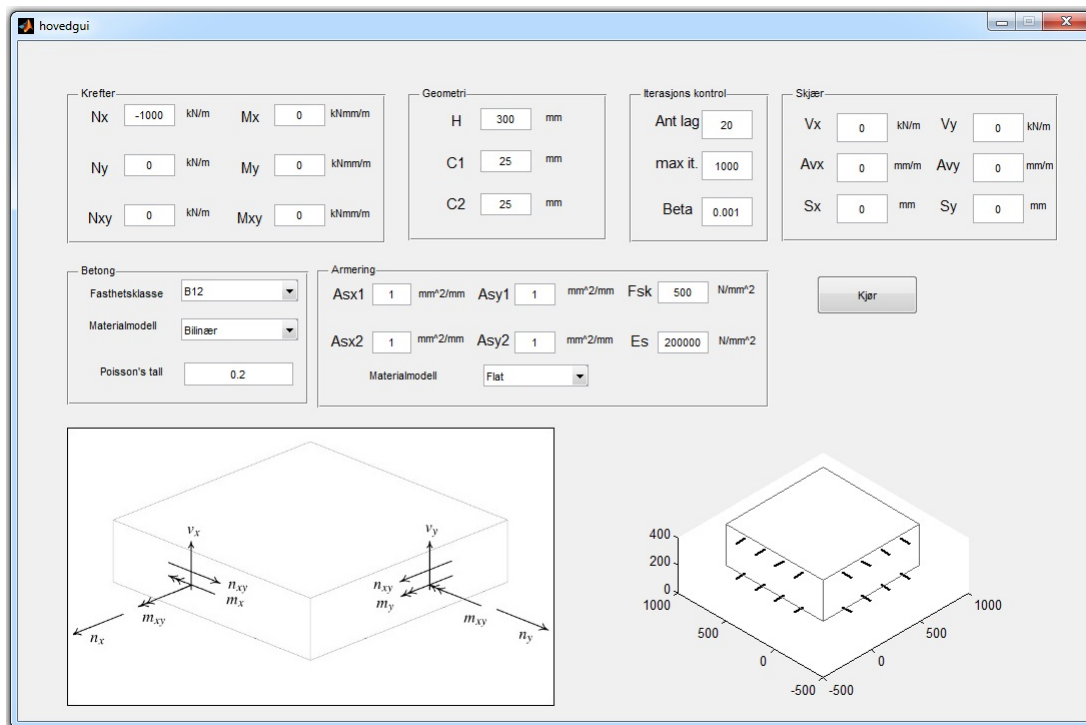
Et annet problem er hvordan koden skal ta hensyn til strekkfasthet i betongen. Det ble oppdaget at om strekkfastheten ble tatt med ville ikke koden konvergere i visse tilfeller hvor skallet er belastet med moment, men deler av betongens strekkfasthet ville svinge frem og tilbake rundt 0 strekkfasthet. Det ble ikke funnet noen bra løsning på dette og lar heller programmet regne kapasitet uten noen strekkfasthet. I stedet for er det en egen innstilling i steg.m hvor brukeren selv kan ta med strekkfasthet i betongen, ved å sette tverrmatr=1. Siden det kan skape problemer med konvergens, ble det valgt å ha la dette være en skjult innstilling.

Om noe burde vært gjort annerledes av programmeringen, så skulle det ha vært brukt celler[10] istedenfor matriser for alle de mer sammensatte variablene, som feks stivhetsmatrisen. Det ville innebære å skrive om veldig mye av koden, men ville gjøre koden veldig lettere å lese. Med vanlige matriser blir indeksene for å legge til komponenter veldig lange og uoversiktlige. Her er en komponent for betong stivhetsmatrisen vist: $K_c011 = C(1 : 3, (3 * i + 1) : (3 * i + 3)) + K_c011$, om man hadde brukt celler ville det blitt: $K_c011 = C\{i\} + K_c011$.

4 Brukerinstruks

Grensesnittet er designet for å være så brukervennlig som mulig. For å kjøre programmet startes MATLAB og så kjører man hovedgui.m for å starte grensesnittet. De nødvendige filene for å kjøre er gitt i Vedlegg B. Dette kapittel er lagt med programmet som brukermanual.

4.1 Innstillinger

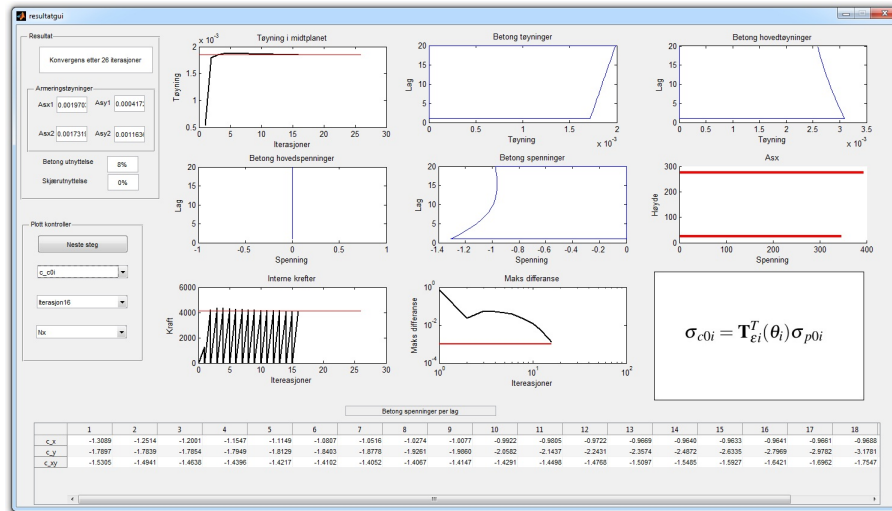


Figur 10: Skjermbilde av innstillingsgrensesnittet

Første panel viser innstillinger for kreftene, fortegnene følger figuren vist i grensesnittet. I geometri panelet er C1 avstand fra topp til armering og C2 fra bunn. Iterasjons kontroll panelet styrer maks antall iterasjoner ved max.it. I betong panelet velges fasthetsklasse og materialmodell fra nedtrekkslister. I panelet for skjær velges skjærkrefter V_x og V_y ,

skjærarmering A_{vx} og A_{vy} , avstand mellom skjærarmeringen S_x og S_y . Bilde nede til høyre reflekterer den valgte geometrien og armeringen, bilde til venstre er samme som Figur 2.

4.2 Resultat



Figur 11: Skjerm bilde av resultatgrensesnittet

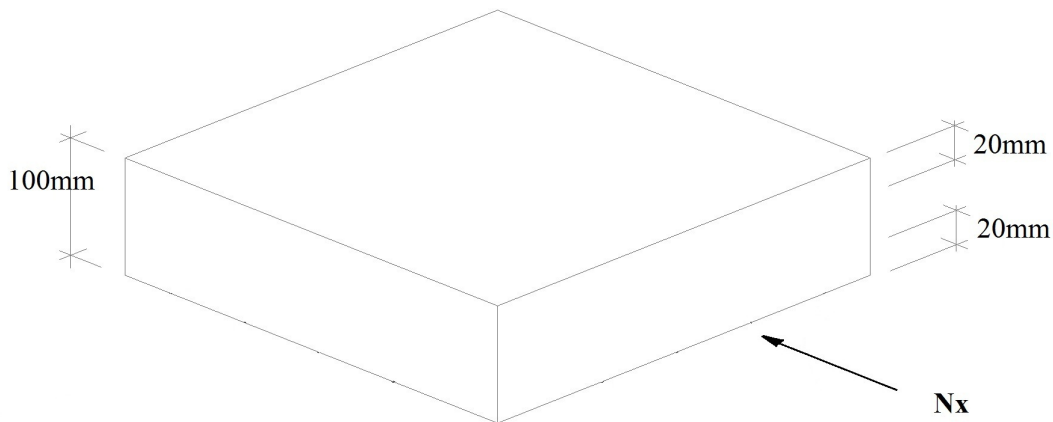
Dette vinduet åpnes automatisk når man trykker kjør i innstillingsgrensesnittet, når det først blir åpnet er ingen av plottene oppdatert. Resultat panelet viser antall iterasjoner som er blitt brukt, evt om det ikke er konvergens, og utnyttelse av armering, betong og skjærkapasitet. I plott kontroller er det tre nedtrekks menyer, den første velger hvilken komponent av plottet som er aktivt, nummer to velger hvilken iterasjon og nummer tre velger hvilken kraft som er aktiv. Plott oppdateringen fungerer slik at om man velger noe fra liste nummer to og tre vil alle plottene oppdatere seg, mens liste nummer to bare oppdaterer matrisen nederst i grensesnittet. Knappen for neste steg vil gå igjennom den første nedtrekkslisten og oppdatere plott for plott, og når den listen er gått igjennom vil den skifte til neste iterasjon og gjenntar prosessen. På denne måten gå igjennom iterasjonsprosessen på samme måte som programmet gjør.

Det første plottet viser tøyningene i midtplanet (2.5.4) mot iterasjoner, den røde horisontale linjen viser den endelige tøyningen. Plott nummer to viser tøyninger i hvert betong lag (2.5.5) mot lag nummer. Plott tre viser betonghovedtøyningene(2.5.6) mot lag nummer. Plott nummer fire viser betonghovedspenningene mot lag nummer. Plott nummer fem viser betong spenningene(2.5.8) mot lag nummer. Plott nummer seks viser armeringspenning(2.5.10) mot høyden, hvor linjene reflekterer plasseringen av armeringen i tverrsnittet. Plott syv viser de interne kreften(2.5.12) mot iterasjoner, hvor den røde horisontale linjen er belastning for den kraften. Plott åtte viser maksdifferanse (2.5.13) mot antall iterasjoner, dette er et loglog[8] plott for å kunne vise når den gitte kraften passerer grensen for konvergens. Det siste plottet viser hvilken formel som ble brukt for å komme frem til den aktive komponenten fra nedtrekksliste nummer en. Nederst i grensesnittet er det også en matrise som viser verdier fra den aktive komponenten og iterasjonen.

5 Verifisering av programmet

For å forsikre at programmet gir riktig resultat ved bruk av ulike innstillinger må det sammenlignes med kjente metoder. Følgende vil programmet sammenlignes med flere eksempler hvor det brukes ulike oppsett. I de fire første eksemplene er det testet i både x og y retning, og oppnådd like resultat. Det er også sammenlignet med et eksempel hvor det er brukt et tilsvarende program for å regne ut resultatet[1].

I eksempel 1 og 4 er det forsøkt å bruke forskjellige iterasjons innstillinger for å se hva for innvirkninger dette har på resultatet og beregningstiden i programmet. I de eksemplene som det er mulig er det blitt kontroll regnet for hånd med kjente metoder. Kontroll utregningene ligger i tillegg A.



Figur 12: Eksempel 1, Trykk

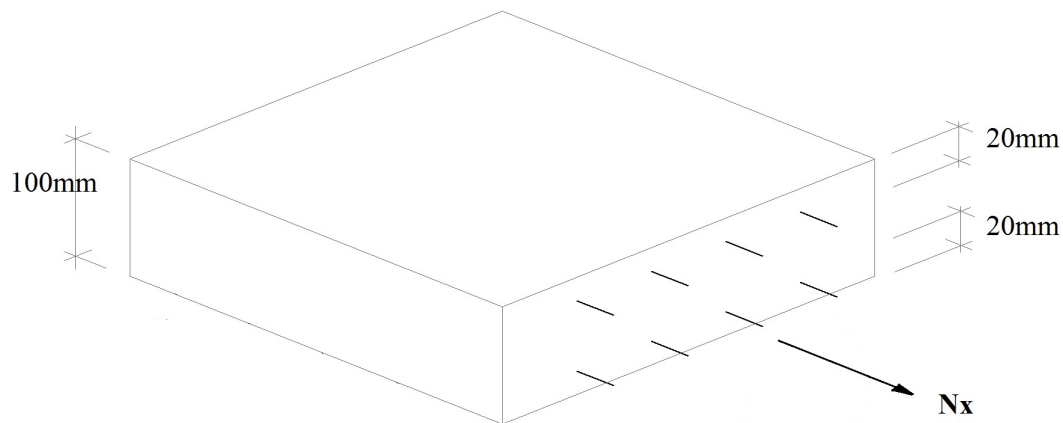
5.1 Eksempel 1; Trykk

Dette er det enkleste tilfellet med kun en trykk kraft på betongen uten armering. Kraften er i x retningen, det brukes B30 i betong kvalitet og høyden på tversnittet er 100mm.

Bilinær materialmodell			
Kraft	1000 kN	1700 kN	2150 kN
Spenning program	10 N/mm ²	17 N/mm ²	-
Spenning kontroll	10 N/mm ²	17 N/mm ²	-
Tøyning program	0.001	0.0017	-
Tøyning kontroll	0.001	0.0017	-
Antall iterasjoner	2	2	1000
Idealisert materialmodell, $\beta = 0.001$			
Kraft	1000 kN	1700 kN	2150 kN
Spenning program	9.99 N/mm ²	16.983 N/mm ²	-
Spenning kontroll	10 N/mm ²	17 N/mm ²	-
Tøyning program	0.000715	0.0019	-
Tøyning kontroll	0.00716	0.0019	-
Antall iterasjoner	5	5	1000
Idealisert materialmodell, $\beta = 0.0001$			
Kraft	1000 kN	1700 kN	2150 kN
Spenning program	9.99 N/mm ²	16.998 N/mm ²	-
Spenning kontroll	10 N/mm ²	17 N/mm ²	-
Tøyning program	0.000716	0.002	-
Tøyning kontroll	0.00716	0.002	-
Antall iterasjoner	7	100	1000
Ikke-linær materialmodell, $\beta = 0.001$			
Kraft	1000 kN	1700 kN	2150 kN
Spenning program	9.99 N/mm ²	16.991 N/mm ²	21.479 N/mm ²
Spenning kontroll	10 N/mm ²	17 N/mm ²	21.5 N/mm ²
Tøyning program	0.000749	0.0014	0.0021
Tøyning kontroll	0.000749	0.0014	0.0021
Antall iterasjoner	2	4	17
Ikke-linær material modell, $\beta = 0.0001$			
Kraft	1000 kN	1700 kN	2150 kN
Spenning program	9.99 N/mm ²	16.998 N/mm ²	21.4979 N/mm ²
Spenning kontroll	10 N/mm ²	17 N/mm ²	21.5 N/mm ²
Tøyning program	0.000749	0.0014	0.0021
Tøyning kontroll	0.000749	0.0014	0.0021
Antall iterasjoner	3	6	35

Tabell 1: Eksempel 1; Trykk

Den bilinære modellen gir et eksakt resultat, men den idealiserte og ikke-linære modellen vil ikke gi et nøyaktig resultat, dette kan bedres ved å minke β , dette øker da antallet nødvendige iterasjoner. Antallet økte iterasjoner er neglisjerbart og har veldig lite å si for total tid brukt på utregningen. Illustrert her ved å ta tiden på iterasjonsprosessen. Den ikke-linære modellen, belastet med 2150kN, brukte ved $\beta = 0.001$, 0.049 sekunder, mot 0.099 sekunder for $\beta = 0.0001$.



Figur 13: Eksempel 2, Strekk

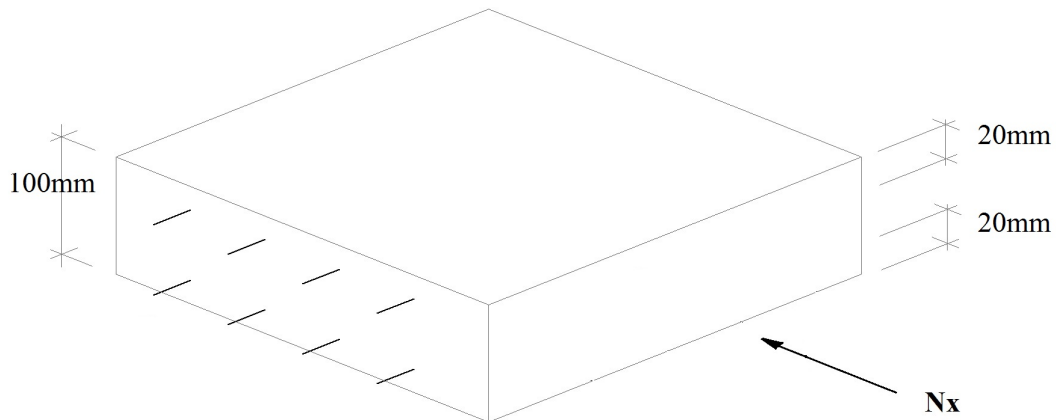
5.2 Eksempel 2; Strekk

Det andre eksemplet vil teste armeringen. Her utsetes tverrsnittet for rent strek i x retning. Høyden på tverrsnittet er 100mm og det brukes 1mm/mm armering, samtidig er tverrkontraksjonstallet satt til 0.

Flat material modell		
Kraft	400kN	450kN
Spenning program	400 N/mm^2	-
Spenning kontroll	400 N/mm^2	-
Tøyning program	0.02	-
Tøyning kontroll	0.02	-
Antall iterasjoner	2	1000
Stigende material modell		
Kraft	400kN	450kN
Spenning program	400 N/mm^2	450 N/mm^2
Spenning kontroll	400 N/mm^2	450 N/mm^2
Tøyning program	0.02	0.01905
Tøyning kontroll	0.02	0.01968
Antall iterasjoner	2	163

Tabell 2: Eksempel 2; Strekk

Som vi ser klarer programmet å beregne korrekt tøyning med få iterasjoner før flyting, men for den stigende modellen etter flyting bruker programmet mange iterasjoner og får heller ikke nøyaktig samme tøyning. Dette kan forbedres ved å bruke en mindre β , men med en bekostning av mange iterasjoner.



Figur 14: Eksempel 3, Tverrkontraksjon

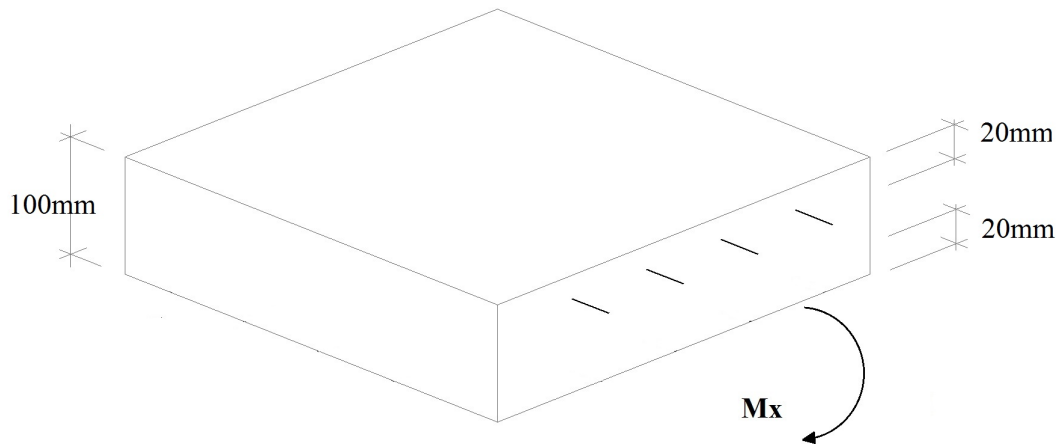
5.3 Eksempel 3; Tverrkontraksjon

Det tredje eksemplet undersøker hvordan programmet tar hensyn til tverrkontraksjon. Høyden på tverrsnittet er 100mm, c_1 og c_2 er 20mm, betongfastheten er B30 med $\nu = 0.2$ det brukes $1\text{mm}^2/\text{mm}$ armering med en flat materialmodell i y retning. I første del brukes det en bilinær materialmodell med strekkfasthet, i del 2 brukes en modell uten strekkfasthet.

Bilinær materialmodell, $\beta = 0.001$, $f_{ctm} = 1.6433N/mm^2$	
Kraft	100kN
Betong spenning program	$1N/mm^2$
Betong spenning kontroll	$1 N/mm^2$
Betong tøyning program	-0.00010224
Betong tøyning kontroll	-0.00010224
Stål tøyning program	0.00001707
Stål tøyning kontroll	0.00001707
Antall iterasjoner	1
Bilinær materialmodell, $\beta = 0.001$, $f_{ctm} = 0N/mm^2$	
Kraft	100kN
Betong spenning program	$-1N/mm^2$
Betong spenning kontroll	$-1N/mm^2$
Betong tøyning program	-0.00010294
Betong tøyning kontroll	-0.00010294
Stål tøyning program	0
Stål tøyning kontroll	0
Antall iterasjoner	2

Tabell 3: Eksempel 3; Tverrkontraksjon

Som vi ser i tabell 3, begge gir samme svar som kontrollberegningene. I del 1 tar vi hensyn til strekkfastheten i betongen og får overført kreftene til armeringen i motsetning til del 2. Del 2 sin første iterasjon er identisk til del 1, men blir oppdatert for å ta hensyn til at poisson's tall blir null når betongen er opprisset. Tilsvarende vil skje for betongen med strekkfasthet, om vi går over tøyningsgrensen i strekk. Vi får helt eksakt svar her, med en linær material modell blir programmets utregning identisk til kontrollberegningene.



Figur 15: Eksempel 4, moment

5.4 Eksempel 4; Moment

Det fjerde eksemplet tester programmet for tverrsnitt belastet med moment i en retning. Tverrsnittet er 100mm, c_1 er 20mm, fasthetsklasse B30, $1\text{mm}^2/\text{mm}$ i x retning og poisson's tall er 0. Tverrsnittet testes for maksimum moment utnyttelse, som er regnet ut i A.4, 29.2227kNm.

Idealisert material modell $\beta = 0.001$			
Moment	29.2227kNmm/m	29.2227kNmm/m	29.3kNmm/m
Antall lag	20	30	30
Antall iterasjoner	1000	3	1000

Tabell 4: Eksempel 4; Moment

Som vi ser i Tabell 4 klarer vi oppnå samme kapasitet som i kontrollberegningene, men vi trenger å øke antall lag til 30 for å få denne kapasiteten. Grunnen til dette er at det er akkurat i grensen for hva tverrsnittets kapasitet, som vi ser vil vi ikke få konvergens for 29.3kNm. I dette tilfellet vil økt antall lag gi en mer nøyaktig nøytralakse for tverrsnittet

som her gir bedre kapasitet. I andre tilfeller vil det være nødvendig å ha et gitt antall lag for å få den riktige tøyningen.

Idealisert materialmodell $\beta = 0.001$			
Moment	20kNmm/m	20kNmm/m	20kNmm/m
Antall lag	2	15	30
Maks betongspenning	$7.3N/mm^2$	$13.66N/mm^2$	$14.03N/mm^2$
Antall iterasjoner	5	5	5

Tabell 5: Eksempel 4, Innvirkning av antall lag

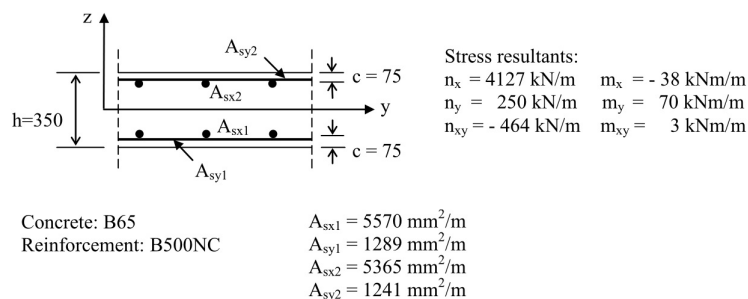
Som vi ser i Tabell 5 vil et lavt antall lag gi forskjellig spenning fra et høyt antall. Dette er som forventet da et tverrsnitt i bøyning med bare 2 lag henter tøyninger en fjerdedel fra toppen mens med et høyt antall lag vil tøyningene tilnærmet bli hentet fra ytterkantene. Det å bruke flere antall lag har heller liten innvirkning på total tid brukt, selv om det er samme antall iterasjoner vil programmet bruke lenger tid på utregningene, 0.0037s for 2 lag mot 0.01s for 30 lag, men dette er allikevel ikke merkbart. Det vil da være hensiktsmessig å bruke et stort antall lag ved utregninger for å få så eksakte tøyninger som mulig.

5.5 Eksempel 5; Skjærkraft

Kontroll av skjærkraft er en separat operasjon som skjer etter at iterasjonsmetoden har kjørt. I dette eksempelet brukes fasthetsklass B30, høyde 100mm, 20mm overdekkning på begge sider, $1mm^2/mm$ lengde armering i x retning, $350mm^2/m$ skjærarmering i x retning med 100mm senter avstand og er belastet med en skjærkraft i x retning på $100kN/m$. Programmet gir en utnyttelse på 97% som tilsvarer kontroll beregningene.

5.6 Sammenligning med annet program

I kompendium for TKT4222 [1], er det i 5.8.10 regnet et eksempel med et tilsvarende program som benytter iterasjonsmetoden. Ved å sammenligne resultat fra det eksemplet med programmet gir det indikasjon på om programmet regner korrekt.



Figur 16: Eksempel 5.8.10 fra kompendium TKT4222 [1]

Som det vises i Figur 16, er det benyttet betongfasthet B65, dette er ikke en fasthetsklasse som er med i programmet, denne ble da lagt rett inn i koden når eksemplet skulle testes og fjernet deretter og om det skal testes på nytt må brukeren selv gjøre dette, det er vist i kontrollberegningene hvordan dette kan gjøres. Videre er det brukt en ikke-linær material modell og $\beta = 0.001$, det er ikke oppgitt noe mer informasjon enn dette. Det er to deler av eksemplet hvor det er samme oppsett bare at det i del 2 blir det redusert mengde armering i A_{sy2} , $500 \text{ mm}^2/\text{m}$. Programmet bruker ikke samme plassering av armeringen som i eksemplet, A_{sx1} tilsvarer A_{sx2} i eksemplet og A_{sy1} tilsvarer A_{sy2} . I tabell 6 er det utgangspunkt i programmets plassering av armeringen. Resultatene er oppgitt som tøyninger i armering og betong.

Del 1 Eksempel 5.8.10					
Tøyning	Betong	A_{sx1}	A_{sx2}	A_{sy1}	A_{sy2}
Eksempel	0.0004	0.0041	0.002	0.0013	0.0031
Program	0.0004	0.0041	0.002	0.0013	0.0031
Del 2 Eksempel 5.8.10					
Eksempel	0.0007	0.0092	0.0021	0.0038	0.0022
Program	0.0007	0.0092	0.0021	0.0038	0.0022

Tabell 6: Resultat Eksempel 5.8.10

Som vi ser fra tabell 6 er resultatene helt like. Men om vi ser på antall iterasjoner brukt i

eksemplet; 365 i Del 1 og 349 i Del 2, programmet brukte 389 i Del 1 og 350 i Del 2. Siden eksemplet kun oppgir noen få tøyninger, uten flere desimaler, er det vanskelig å se hva som er forskjellen. Videre måtte det brukes $n = 50$ for å oppnå disse resultatene, det er ikke nevnt hva som ble brukt i eksemplet. Samt er det usikkerhet rundt hvilken elastitetsmodul som ble brukt i den ikke-linære modellen, EC2 [3] oppgir at E_{cm} er tilnærmet vha $0.4f_{cm}$. Det ble brukt $E_{cm} = 24000$ i programmet for å oppnå disse resultatene, men små forandringer på denne gir store forandringer i de endelige tøyningene.

Dette er et komplisert eksempel med alle krefter, og uten mer data fra det originale eksemplet er det vanskelig å trekke noen konklusjoner om hvorfor disse forskjellene eksisterer. Siden det nettopp er et så komplisert eksempel og det er oppnådd samme resultater indikerer dette at programmet regner riktig.

6 Konklusjon

Det kan konkluderes med at programmet regner riktig etter de eksemplene som er beregnet. Det eksisterer små forskjeller mellom resultatene fra programmet og eksemplene. For de simple eksemplene kan forskjellene i resultatene forklares med at den numeriske metoden er en tilnærming. I Eksempel 1, ved å minske β vil svaret gå mot det eksakte svaret, tilsvarende for de andre eksemplene er det mulig å forbedre resultatet ved å justere de numeriske parameterne.

For sammenligningen med det andre programmet finnes det forskjeller, dette er på grunn av manglende informasjon rundt hvilke innstillinger som ble brukt for det andre programmet. Mye av konklusjonen hviler på at resultatet fra sammenligningen med det andre programmet er korrekt. Siden eksempelet er såpass komplekst, med alle krefter, ikke linær materialmodell og armering i alle retninger, så er det sannsynlig at begge programmene regner riktig.

For å sjekke kapasitet av skjærkraft er det en egen separat kontroll fra iterasjonsmetoden. Denne kontrollen i programmet har ingen innvirkning på de andre delene av programmet. Fra det ene enkle eksempelet som er kjørt er det ikke mulig å trekke noen konklusjoner om programmet regner riktig, annet enn ved de simpleste tilfeller. Det må gjennomføres en grundig sjekk, hvor skjærkreftene er kombinert med de andre kreftene, for å få en endelig konklusjon.

Det er flere muligheter for videreføring av programmet. Først og fremst å kontrollere delen for skjærkapasitet med kjente metoder, eventuelt da forandre på koden om det er noe som ikke er tilfredsstillende. For å få en uavhengig verifikasjonen av programmet, ikke bare en sammenligning med annet program, burde det gjennomføres en sjekk med elementmetoden eller andre kjente metoder.

Videre er det mulig å forbedre hvordan resultatet presenteres, målet var å lage et brukervennlig program som viser hvordan iterasjonsmetoden beregner kapasitet. Hvorvidt presentasjonen i dette programmet er bra, er det ikke mulig å dra noen konklusjoner om, da det er opp til den individuelle bruker å vurdere dette.

Referanser

- [1] Jan Arve Øverli og Svein I. Sørensen. *TKT4222 Concrete Structures 3 Autumn 2012 Compendium*. (2012).
- [2] Svein I. Sørensen. *Betongkonstruksjoner. Beregning og dimensjonering etter Eurocode 2*. tapir akademisk forlag. (2010).
- [3] Norsk Standard. *NS-EN 1992-1-1:2004 + NA:2008. Eurokode 2: Prosjektering av betongkonstruksjoner Del 1-1:Allmenne regler og regler for bygninger*. (2004).
- [4] O.S. Hopperstad og T.Børvik. *Lecture Notes TKT4135 Mechanics of materials*. (2012).
- [5] Robert D. Cook, David S. Malkus, Michael E.Plesha og Robert J. Witt. *Concepts and applications of finite element analysis. Fourth edition*. John Wiley and Sons, INC. (2002).
- [6] The MathWorks, Inc. MATLAB Documentation Center. Moore-Penrose inverse (pseudoinverse) of symbolic matrix. Hentet 13.05.2013 fra <http://www.mathworks.se/help/symbolic/pinv.html>
- [7] The MathWorks, Inc. MATLAB Documentation Center. Condition number with respect to inversion. Hentet 13.05.2013 fra <http://www.mathworks.se/help/matlab/ref/cond.html>
- [8] The MathWorks, Inc. MATLAB Documentation Center. Log-log scale plot. Hentet 13.05.2013 fra <http://www.mathworks.se/help/matlab/ref/loglog.html>
- [9] The MathWorks, Inc. MATLAB Documentation Center. Start clock to measure performance. Hentet 13.05.2013 fra <http://www.mathworks.se/help/matlab/ref/tic.html>
- [10] The MathWorks, Inc. MATLAB Documentation Center. Create cell array. Hentet 13.05.2013 fra <http://www.mathworks.se/help/matlab/ref/cell.html>

Vedlegg

A Kontrollberegninger

Alle enkelt utregningene er ikke med, men alle metodene er vist.

A.1 Kontrollberegning Eksempel 1

Bilinær material modell:

$$\text{Spenning: } \sigma_c = \frac{N}{A} = \frac{1000kN}{100mm^2} = 10N/mm^2$$

$$\text{Tøyning: } \varepsilon_c = \frac{\sigma_c}{E} = \frac{10N/mm^2}{9714N/mm^2} = 0.001$$

Idealisert material modell:

$$\begin{aligned} \text{Tøyning: } \varepsilon_c &= \left(1 - \left(1 - \frac{\sigma_c}{f_{cd}}\right)^{(1/n)}\right) \varepsilon_{c2} = \\ &= \left(1 - \left(1 - \frac{10N/mm^2}{17N/mm^2}\right)^{(1/2)}\right) 0.002 = 0.000716 \end{aligned}$$

Ikke-linær material modell:

Faktorer: $\eta = \frac{\epsilon_c}{\epsilon_{c1}} \quad k = 1.05 \cdot E_{cm} \cdot |\epsilon_{c1}| / f_{cm}$

$$k = 1.05 \cdot 13486 \text{ N/mm}^2 \cdot |0.0022| / 21.53 \text{ N/mm}^2 = 1.315$$

Annengradsligning: $\alpha = 1 \quad \beta = k \left(\left(\frac{\sigma_c}{f_{cm}} \right) - 2 \right) - \left(\frac{\sigma_c}{f_{cm}} \right) =$

$$1.315 \left(\left(\frac{10 \text{ N/mm}^2}{21.53 \text{ N/mm}^2} \right) - 2 \right) - \left(\frac{10 \text{ N/mm}^2}{21.53 \text{ N/mm}^2} \right) = -1.704 \quad \gamma = \frac{\sigma_c}{f_{cm}} = \frac{10 \text{ N/mm}^2}{21.53 \text{ N/mm}^2} = 0.464$$

$$\eta = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} = \frac{1.704 \pm \sqrt{-1.704^2 - 4 \cdot 1 \cdot 0.464}}{2 \cdot 1} \quad \eta_1 = 0.3407 \quad \eta_2 = 1.3629$$

Tøyning: $\epsilon_c = \eta_1 \epsilon_{c1} = 0.3407 \cdot 0.0022 = 0.000749$

A.2 Kontrollberegning Eksempel 2

Flat materialmodell

Spenning: $\sigma_s = \frac{N}{A_s} = \frac{400 \text{ kN/m}}{1 \text{ mm}^2/\text{mm}} = 400 \text{ N/mm}^2$

Tøyning: $\epsilon_s = \frac{\sigma}{E_s} = \frac{400 \text{ N/mm}^2}{200000 \text{ N/mm}^2} = 0.002$

Stigende materialmodell

Spenning: $\sigma_s = \frac{N}{A_s} = \frac{450 \text{ kN/m}}{1 \text{ mm}^2/\text{mm}} = 450 \text{ N/mm}^2$

Tøyning: $\sigma_s = f_{yd} + f_{yd}(k-1) \frac{\epsilon_s}{\epsilon_{ud}}$

$$\epsilon_s = \epsilon_{ud} \left(\frac{\sigma_s - f_{yd}}{f_{yd}(k-1)} \right) = 0.05 \cdot 0.9 \cdot \left(\frac{450 \text{ N/mm}^2 - 434 \text{ N/mm}^2}{434 \text{ N/mm}^2 (1.08 - 1)} \right) = 0.01968$$

A.3 Kontrollberegning Eksempel 3

Følger oppsett for ortotropisk elastisitet [4].

$$\text{Last: } N = \begin{bmatrix} -100kN \\ 0 \\ 0 \end{bmatrix}$$

$$\text{Materialmatrise betong: } C_c = \frac{E_{cm}}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{9714}{1-0.2^2} \begin{bmatrix} 1 & 0.2 & 0 \\ 0.2 & 1 & 0 \\ 0 & 0 & \frac{1-0.2}{2} \end{bmatrix} N/mm^2$$

$$\text{Materialmatrise armering: } C_s = \begin{bmatrix} 0 & 0 & 0 \\ 0 & E_s & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 200000 & 0 \\ 0 & 0 & 0 \end{bmatrix} N/mm^2$$

$$\text{Stivhetsmatrise: } D = h \cdot b \cdot C_c + A_s \cdot C_s = \begin{bmatrix} 1011900 & 202400 & 0 \\ 202400 & 1211900 & 0 \\ 0 & 0 & 404800 \end{bmatrix}$$

$$\text{Tøyninger: } \varepsilon = \text{inv}(D) \cdot N = \begin{bmatrix} -0.00010224 \\ 0.00001707 \\ 0 \end{bmatrix}$$

$$\text{Betongspenninger: } \sigma_c = C_c \cdot \varepsilon = \begin{bmatrix} -1 \\ -0.0341 \\ 0 \end{bmatrix} N/mm^2$$

$$\text{Armeringspenninger: } \sigma_s = C_s \cdot \varepsilon = \begin{bmatrix} 0 \\ 3.4146 \\ 0 \end{bmatrix} N/mm^2$$

Tilsvarende for del 2, bare med $\nu = 0$.

A.4 Kontrollberegning Eksempel 4

Følger oppsett [2] for å finne maksimum momentkapasitet.

Balansert trykksonehøyde:
$$\alpha_b = \frac{\epsilon_{cu}}{\epsilon_{cu} + \epsilon_{yd}} = \frac{0.0035}{0.0035 + 0.00217} = 0.619$$

Balansert armeringsmengde:
$$A_{s,b} = 0.8 \frac{f_{cd}}{f_{sd}} b d \alpha_b = 0.8 \frac{17}{434} 1000 \cdot 80 \cdot 0.619 = 1543 \text{ mm}^2$$

$$A_s < A_{s,b}$$

Tverrsnittet er underarmert

Aktuell α finnes:
$$\alpha = \frac{f_{sd} A_s}{0.8 f_{cd} b d} = \frac{434 \cdot 1000}{0.8 \cdot 17 \cdot 1000 \cdot 80} = 0.3996$$

Momentkapasitet:
$$M_{rd} = 0.8 \alpha (1 - 0.4 \alpha) f_{cd} b d^2$$

$$= 0.8 \cdot 0.3996 \cdot (1 - 0.4 \cdot 0.3996) \cdot 17 \cdot 1000 \cdot 80^2 = 29.227 \text{ kNm}$$

A.5 Kontrollberegning Eksempel 5

Finner kapasiteten av tilsvarende tverrsnitt.

Kapasitet skjærarmering:

$$V_{Rds} = \frac{A_{sw}}{s} \cdot z \cdot f_{ywd} \cdot \cot \theta$$

$$V_{Rds} = \frac{350 \text{ mm}}{200} \cdot 54 \text{ mm} \cdot 434 \text{ N/mm}^2 \cdot 2.5 = 102.5 \text{ kN}$$

$$UR_{vc} = \frac{100 \text{ kN}}{102.5 \text{ kN}} \cdot 100\% = 97\%$$

A.6 Sammenligning med annet program

For å sette inn egen fasthetsklasse for betongen, gå til betongmatr.m, og sett inn parameterne øverst i koden. Følgende innstillinger ble brukt for eksempelet:

```
f_cd = 36.833;
E = 24000;
epsilon_c1 = -0.00265;
epsilon_cu1 = -0.0029;
```

B Filliste

For å kunne kjøre programmet må følgende filer ligge i samme mappe:

Filnavn	Filtype
hovedgui.m	MATLAB kode
hovedgui.fig	MATLAB figur
resultatgui.m	MATLAB kode
resultatgui.fig	MATLAB figur
hoved.m	MATLAB kode
steg.m	MATLAB kode
kc.m	MATLAB kode
k_s.m	MATLAB kode
betongmatr.m	MATLAB kode
s_s.m	MATLAB kode
skjarkontroll.m	MATLAB kode
ur.m	MATLAB kode
1(r).jpeg	Bilde
2(ktot).jpeg	Bilde
3(etot).jpeg	Bilde
4(ei).jpeg	Bilde
7(oci).jpeg	Bilde
8(osj).jpeg	Bilde
9(stot).jpeg	Bilde
10(maxdiff).jpeg	Bilde
krefter.jpeg	Bilde

Tabell 7: Filliste

C Utdrag fra programmet

Hele programmet er ikke tatt med, for å full innsikt må filene åpnes i MATLAB. Bare utdrag er tatt med for å vise hvordan ulike deler av teorien har blitt overført til funksjoner i MATLAB. Ikke alle kommentarer og deklarasjoner av variabler er tatt med.

C.1 hoved.m

```
%Geometri

%Armerings arm
z_s=[h/2-c1,-h/2+c1,h/2-c1,-h/2+c2];

%Oppdeling i lag
delta_h = h/n;
z=1:n-1;
for i = 0:1:n-1;
    z(i+1) = ((-h/2)+(delta_h/2)) +(delta_h*i));
end
z=transpose(z);

%Betong material
%Bestemmer Ecm
if betongmat == 1
    E = abs(f_cd/epsilon_c3);
elseif betongmat == 2
    E = abs(f_cd/epsilon_c2);
elseif betongmat == 3
    maxe = 1;
    f_cm = f_cd + (8*0.85/1.5);
    fcm4 = 0.4*f_cm;
    epsilon_e = fcm4/E;
    ek=1;

%Iterer for a finne ikkelinaer Ecm

while ek > 0.001 && maxe < 200
    maxe = maxe+1;
    k = (1.05*E*epsilon_c1)/f_cm;
```

```

        eta = epsilon_e/epsilon_c1;
        oc = f_cm*((k*eta-eta^2)/(1+(k-2)*eta));
        E = oc/epsilon_e;
        epsilon_e = fcm4/E;
        ek = abs(oc-fcm4);
    end
end
E = abs(E);

%Deler opp v for alle betonglagene
v=zeros(1,20);
for i =1:n;
    v(i) = v1;
end

%Initiell materialmatrise betong

for i = 0:1:n-1;
    C(1,3*i+1)=(1/(1-v(i+1)^2))*E11;
    C(1,3*i+2)=(1/(1-v(i+1)^2))*v(i+1)*E12;
    C(2,3*i+1)=(1/(1-v(i+1)^2))*v(i+1)*E12;
    C(2,3*i+2)=(1/(1-v(i+1)^2))*E22;
    C(3,3*i+3)=(1/(1-v(i+1)^2))*((1-v(i+1))*E12)/2;
end

%Material matrise armering
C_s = zeros(3,12);
C_s(1,1) = E_sint;
C_s(1,4) = E_sint;
C_s(2,8) = E_sint;
C_s(2,11) = E_sint;

%Kjorer steg.m hvor iterasjons prosessen foregar

[ks,imax,...,epsilon_t0_tot] = steg(C,z,delta_h,n,...,h);

%Skjaerkraft beregning
[UR_vc] = skjarkontroll(f_cd,As,h,c1,c2,Sx,Sy,R,As_vx,As_vy,f_sd,V_x,V_y);

%Kjorer ur.m for a bestemme utnyttelse ratioer
[UR_C,UR_S] = ur(epsilon_cul,...,betongmat);

```

C.2 steg.m

```
while (ks>beta && imax<n_iterasjoner)

%Teller antall iterasjoner gjort
imax=imax+1;

%Kjører kc.m for a danne stivhetsmatrisen for betong
[K_c0] = kc(C,z,delta_h,n);

%Kjører k_s. for a danne stivhetsmatrisen for armeringen
[K_s] = k_s(As,C_s,z_s);
%Total stivhet
K_0 = K_s + K_c0;

%Finne toyninger i midten av skall planet
%Ma sjekke hvilken inverterings metode som brukes

if cond(K_0)>1e10;
    epsilon_t0 = pinv(K_0)*R;
else
    epsilon_t0 = K_0\R;
end

%Finner toyningene i planet i hvert betong lag

for i=1:1:n;
    epsilon_0i(1:3,i)=[1,0,0,-z(i),0,0;0,1,0,0,-z(i),0;0,0,1,0,0,-z(i)]*epsilon_t0;
end

%Finne hovedtoyninger og hoved retninger i betonglagene

for i=1:1:n;
    theta_i(i)=0.5*atan((epsilon_0i(3,i))/(epsilon_0i(1,i)-epsilon_0i(2,i)));
end

%Transformasjons matrise

for i=0:1:n-1;
    T(1,(3*i)+1) = cos(theta_i(i+1))^2;
```

```

T(1, (3*i)+2) = sin(theta_i(i+1))^2;
T(1, (3*i)+3) = sin(theta_i(i+1))*cos(theta_i(i+1));
T(2, (3*i)+1) = sin(theta_i(i+1))^2;
T(2, (3*i)+2) = cos(theta_i(i+1))^2;
T(2, (3*i)+3) = -sin(theta_i(i+1))*cos(theta_i(i+1));
T(3, (3*i)+1) = -2*sin(theta_i(i+1))*cos(theta_i(i+1));
T(3, (3*i)+2) = 2*sin(theta_i(i+1))*cos(theta_i(i+1));
T(3, (3*i)+3) = (cos(theta_i(i+1))^2)-(sin(theta_i(i+1))^2);
end

%Transformerer toyingene
epsilon_p0i = zeros(3,n);

for i=0:1:n-1;
    epsilon_p0i(1:3,i+1) = T(1:3, ((3*i)+1):(3*i)+3)*epsilon_0i(1:3,i+1);
end

%Kjorer betongmatr.m for a finne betongspenninger og ny elastisetsmodul

[sigma_p0i,...,E12_ri] = betongmatr(betongmat,...,E12_ri);

%Sjekker om betongen er i toying i hvert lag for a sette v verdiene
%Her er det mulig a velge om betongen skal ha noen toyningskapasitet sett
%tverrmatr == 1 for at betongen skal fa kapasitet
tverrmatr = 0;

if tverrmatr == 1
    for i= 1:n;
        if epsilon_p0i(1,i) >=epsilon_cs;
            v(i) = 0;
        elseif epsilon_p0i(2,i) >=epsilon_cs;
            v(i) = 0;
        end
    end
else
    for i= 1:n;
        if epsilon_p0i(1,i) >=0;
            v(i) = 0;
        elseif epsilon_p0i(2,i) >=0;
            v(i) = 0;
        else
            v(i) = v1;
        end
    end
end

```

```

        end
    end
end

%Lager ny material matrise for betongen

for i=0:1:n-1;
    C_p(1,3*i+1)=(1/(1-(v(i+1)^2)))*E11_ri(i+1);
    C_p(1,3*i+2)=(1/(1-(v(i+1)^2)))*v(i+1)*E12_ri(i+1);
    C_p(2,3*i+1)=(1/(1-(v(i+1)^2)))*v(i+1)*E12_ri(i+1);
    C_p(2,3*i+2)=(1/(1-(v(i+1)^2)))*E22_ri(i+1);
    C_p(3,3*i+3)=(1/(1-(v(i+1)^2)))*((1-v(i+1))*E12_ri(i+1))/2;
end

C_test = C_p;

%Transformerer material matrisen til x og y retninger

for i=0:1:n-1;
    C(1:3,((3*i)+1):(3*i)+3) = transpose(T(1:3,((3*i)+1):(3*i)+3))*C_p(1:3,((3*i)+1):(3
end

%Betong spenninger basert pa materialmatrisen

for i=0:1:n-1;
    sigma_cpi(1:3,i+1) = C_test(1:3,((3*i)+1):(3*i)+3)*epsilon_p0i(1:3,i+1);
end

%Betong spenninger i x og y retning

for i=0:1:n-1;
    sigma_c(1:3,i+1) = transpose(T(1:3,((3*i)+1):(3*i)+3))*sigma_cpi(1:3,i+1);
end

%Kjorer s_s for a finne armerings spenninger i hvert armerings lag og
%interne krefter fra armering

[S_s,..., C_s] = s_s(z_s,...,imax);

%Interne krefter fra betongen

for i =1:1:n;

```

```

S_0c(1,1) = S_0c(1,1) + sigma_c(1,i)*delta_h;
S_0c(2,1) = S_0c(2,1) + sigma_c(2,i)*delta_h;
S_0c(3,1) = S_0c(3,1) + sigma_c(3,i)*delta_h;
S_0c(4,1) = S_0c(4,1) - z(i)*sigma_c(1,i)*delta_h;
S_0c(5,1) = S_0c(5,1) - z(i)*sigma_c(2,i)*delta_h;
S_0c(6,1) = S_0c(6,1) - z(i)*sigma_c(3,i)*delta_h;
end

S_0 = S_0c + S_s;

%Finner maksimum relativ differanse mellom komponentene av eksterne og interne
%spenningsresultanter

for k=1:1:6;
    if R(k) == 0;
        Maxdiff(k,1) = 0;
    else
        Maxdiff(k,1) = abs((S_0(k)-R(k))/R(k));
    end
end
Maxdiff_tot(1:6,imax) = Maxdiff;
ks=max(Maxdiff);
end

```

C.3 kc.m

```

function [K_c0] = kc(C,z,delta_h,n)

K_c011 = zeros(3,3);
K_c012 = zeros(3,3);
K_c022 = zeros(3,3);

for i = 0:1:n-1;
    K_c011 = C(1:3,(3*i+1):(3*i+3)) +K_c011;
    K_c012 = -(z(i+1))*(C(1:3,(3*i+1):(3*i+3))) + K_c012;
    K_c022 = (z(i+1)^2)*(C(1:3,(3*i+1):(3*i+3))) + K_c022;
end

K_c0 = delta_h*[K_c011, K_c012;K_c012,K_c022];

```

C.4 k_s.m

```
function [K_s] = k_s(As,C_s,z_s)

K_sx1 = zeros(6);
K_sx2 = zeros(6);
K_sy1 = zeros(6);
K_sy2 = zeros(6);

K_sx1(1:3,1:3) = As(1)*C_s(1:3,1:3);
K_sx1(4:6,1:3) = -z_s(1)*As(1)*C_s(1:3,1:3);
K_sx1(1:3,4:6) = -z_s(1)*As(1)*C_s(1:3,1:3);
K_sx1(4:6,4:6) = (z_s(1)^2)*As(1)*C_s(1:3,1:3);

K_sx2(1:3,1:3) = As(2)*C_s(1:3,4:6);
K_sx2(4:6,1:3) = -z_s(2)*As(2)*C_s(1:3,4:6);
K_sx2(1:3,4:6) = -z_s(2)*As(2)*C_s(1:3,4:6);
K_sx2(4:6,4:6) = (z_s(2)^2)*As(2)*C_s(1:3,4:6);

K_sy1(1:3,1:3) = As(3)*C_s(1:3,7:9);
K_sy1(4:6,1:3) = -z_s(3)*As(3)*C_s(1:3,7:9);
K_sy1(1:3,4:6) = -z_s(3)*As(3)*C_s(1:3,7:9);
K_sy1(4:6,4:6) = (z_s(3)^2)*As(3)*C_s(1:3,7:9);

K_sy2(1:3,1:3) = As(4)*C_s(1:3,10:12);
K_sy2(4:6,1:3) = -z_s(4)*As(4)*C_s(1:3,10:12);
K_sy2(1:3,4:6) = -z_s(4)*As(4)*C_s(1:3,10:12);
K_sy2(4:6,4:6) = (z_s(4)^2)*As(4)*C_s(1:3,10:12);

K_s = K_sx1 + K_sx2 + K_sy1 +K_sy2;
```

C.5 s_s.m

```
function [S_s,...,C_s] = s_s(z_s,...,imax)

%Finner toyninger i armeringslagene
for i = 1:4
    if As(i) == 0
        epsilon_0j(1:3,i) = [0;0;0];
```

```

else
    epsilon_0j(1:3,i) = [1 0 0 -z_s(i) 0 0;0 1 0 0 -z_s(i) 0;
    0 0 1 0 0 -z_s(i)]*epsilon_t0;
end
end

%Finner armerings spenninger
%Velger material modell
if stalmat == 1
    %Flat modell
    for i = 1:4;
        if abs(epsilon_0j(1,i)) < epsilon_s02;
            sigma_s0j(1,i) = E_sint*epsilon_0j(1,i);
            if abs(epsilon_0j(1,i)) > epsilon_uk;
                sigma_s0j(1,i) = 0;
            end
        else
            sigma_s0j(1,i) = f_sd*sign(epsilon_0j(1,i));
        end
    end
end

for i = 1:4;
    if abs(epsilon_0j(2,i)) < epsilon_s02;
        sigma_s0j(2,i) = E_sint*epsilon_0j(2,i);
        if abs(epsilon_0j(2,i)) > epsilon_uk;
            sigma_s0j(2,i) =0;
        end
    else
        sigma_s0j(2,i) = f_sd*sign(epsilon_0j(2,i));
    end
end
end
elseif stalmat ==2
    %Stigende modell
    for i = 1:4;
        if abs(epsilon_0j(1,i)) >= epsilon_s02
            sigma_s0j(1,i) = sign(epsilon_0j(1,i))*f_sd +
            (f_sd*(k-1))*(epsilon_0j(1,i)/epsilon_ud);
            if abs(epsilon_0j(1,i)) > epsilon_ud
                sigma_s0j(1,i) = 0;
            end
        else
            sigma_s0j(1,i) = E_sint*epsilon_0j(1,i);

```



```
end

if abs(epsilon_0j(2,i)) >= epsilon_s02
    sigma_s0j(2,i) = sign(epsilon_0j(2,i))*f_sd +
    (f_sd*(k-1))*(epsilon_0j(2,i)/epsilon_ud);
    if abs(epsilon_0j(2,i)) > epsilon_ud
        sigma_s0j(2,i) = 0;
    end
else
    sigma_s0j(2,i) = E_sint*epsilon_0j(2,i);
end
end
end

%Kalkulerer ny materialmatrise for armering
%Velger material modell
if stalmat == 1
    %Flat modell
    for i =1:2;
        if epsilon_0j(1,i) == 0
            Esx(i) = 0;
        else
            Esx(i) = sigma_s0j(1,i)/epsilon_0j(1,i);
        end
    end
    for i =3:4;
        if epsilon_0j(2,i) == 0
            Esy(i-2) = 0;
        else
            Esy(i-2) = sigma_s0j(2,i)/epsilon_0j(2,i);
        end
    end
elseif stalmat == 2;
    for i = 1:2;
        if epsilon_0j(1,i) == 0
            Esx(i) = 0;
        else
            Esx(i) = (sigma_s0j(1,i)/epsilon_0j(1,i));
        end
    end
    for i = 3:4;
        if epsilon_0j(2,i) == 0
```

```

        Esy(i-2) = 0;
    else
        Esy(i-2) = (sigma_s0j(2,i)/epsilon_0j(2,i));
    end
end
end

%Lager ny material matrise
C_s = zeros(3,12);
C_s(1,1) = Esx(1);
C_s(1,4) = Esx(2);
C_s(2,8) = Esy(1);
C_s(2,11) = Esy(2);

%Finner armeringsspenningene basert pa ny materialmatrise
sigma_s0j(1,1) = C_s(1,1)*epsilon_0j(1,1);
sigma_s0j(1,2) = C_s(1,4)*epsilon_0j(1,2);
sigma_s0j(2,3) = C_s(2,8)*epsilon_0j(2,3);
sigma_s0j(2,4) = C_s(2,11)*epsilon_0j(2,4);

%Finner internekrefter fra armeringen
for i = 1:2;
    S_s(1) = S_s(1) + sigma_s0j(1,i)*As(i);
    S_s(2) = S_s(2) + sigma_s0j(2,i+2)*As(i+2);
    S_s(4) = S_s(4) - z_s(i)*sigma_s0j(1,i)*As(i);
    S_s(5) = S_s(5) - z_s(i+2)*sigma_s0j(2,i+2)*As(i+2);
end

```

C.6 betongmatr.m

```

function [sigma_p0i,...,E12_ri] = betongmatr(betongmat,...,E12_ri)

%Regner ut spenninger
%Bestemmer hvilken material modell som skal brukes
if betongmat == 1
    %Bilinaer materialmodell
    %Finner spenningene i hovedretningene
    sigma_p0i = zeros(3,n);
    for i = 1:1:n;
        if epsilon_p0i(1,i) < 0
            sigma_p0i(1,i) = -f_cd*(1-(1-epsilon_p0i(1,i)/epsilon_c3));
        end
    end
end

```

```
        if epsilon_p0i(1,i) <= epsilon_c3;
            sigma_p0i(1,i) = -f_cd;
            if epsilon_p0i(1,i) <= epsilon_cu3
                sigma_p0i(1,i) = 0;
            end
        end
    else
        sigma_p0i(1,i) = 0;
    end

    if epsilon_p0i(2,i) < 0
        sigma_p0i(2,i) = -f_cd*(1-(1-epsilon_p0i(2,i)/epsilon_c3));
        if epsilon_p0i(2,i) <= epsilon_c3;
            sigma_p0i(2,i) = -f_cd;
            if epsilon_p0i(2,i) <= epsilon_cu3
                sigma_p0i(2,i) = 0;
            end
        end
    else
        sigma_p0i(2,i) = 0;
    end
end

%Lager ny sekantmodul for betongen
for i=1:1:n;
    if epsilon_p0i(1,i)>=0
        E11_ri(i,1) = 0;
    else
        E11_ri(i,1) = (sigma_p0i(1,i))/epsilon_p0i(1,i);
    end

    if epsilon_p0i(2,i) >= 0
        E22_ri(i,1) = 0;
    else
        E22_ri(i,1) = (sigma_p0i(2,i))/epsilon_p0i(2,i);
    end

    E12_ri(i,1) = (E11_ri(i,1)+E22_ri(i,1))/2;
end
```

```
elseif betongmat ==2
    %Idealisert modell
    %Bestemmer spenningene i hovedretningen
    for i = 1:1:n;
        if epsilon_p0i(1,i)<0;
            sigma_p0i(1,i) = -f_cd*(1-(1-epsilon_p0i(1,i)/epsilon_c2)^n_eksp);
            if epsilon_p0i(1,i)<epsilon_c2;
                sigma_p0i(1,i) = -f_cd;
                if epsilon_p0i(1,i) < epsilon_cu2
                    sigma_p0i(1,i) = 0;
                end
            end
        else
            sigma_p0i(1,i) = 0;
        end

        if epsilon_p0i(2,i)<0;
            sigma_p0i(2,i) = -f_cd*(1-(1-epsilon_p0i(2,i)/epsilon_c2)^n_eksp);
            if epsilon_p0i(2,i)<epsilon_c2;
                sigma_p0i(2,i) = -f_cd;
                if epsilon_p0i(2,i) < epsilon_cu2
                    sigma_p0i(2,i) = 0;
                end
            end
        else
            sigma_p0i(2,i) = 0;
        end
    end

    %Oppretter ny sekant modulus for betong
    for i=1:1:n;
        if epsilon_p0i(1,i)>=0
            E11_ri(i,1) = 0;
        else
            E11_ri(i,1) = (sigma_p0i(1,i))/epsilon_p0i(1,i);
        end

        if epsilon_p0i(2,i) >= 0
            E22_ri(i,1) = 0;
        end
    end
end
```

```
else
    E22_ri(i,1) = (sigma_p0i(2,i))/epsilon_p0i(2,i);
end

E12_ri(i,1) = (E11_ri(i,1)+E22_ri(i,1))/2;

end
elseif betongmat == 3

%Ikkelinaer material modell
for i = 1:n;
    if epsilon_p0i(1,i) < 0;
        eta = epsilon_p0i(1,i) / epsilon_c1;
        k = 1.05*E*abs(epsilon_c1)/f_cm;
        sigma_p0i(1,i) = -f_cm*((k*eta-eta^2)/(1+(k-2)*eta));
        if epsilon_p0i(1,i) < epsilon_cul;
            sigma_p0i(1,i) = 0;
        end
    else
        sigma_p0i(1,i) = 0;
    end
    if epsilon_p0i(2,i) < 0;
        eta = epsilon_p0i(2,i) / epsilon_c1;
        k = 1.05*E*abs(epsilon_c1)/f_cm;
        sigma_p0i(2,i) = -f_cm*((k*eta-eta^2)/(1+(k-2)*eta));
        if epsilon_p0i(2,i) < epsilon_cul;
            sigma_p0i(2,i) = 0;
        end
    else
        sigma_p0i(2,i) = 0;
    end
end

%Oppretter ny sekant modulus
for i=1:1:n;
    if epsilon_p0i(1,i)>=0
        E11_ri(i,1) = 0;
    else
        E11_ri(i,1) = (sigma_p0i(1,i))/epsilon_p0i(1,i);
    end
end
```

```
    if epsilon_p0i(2,i) >= 0
        E22_ri(i,1) = 0;
    else
        E22_ri(i,1) = (sigma_p0i(2,i))/epsilon_p0i(2,i);
    end

    E12_ri(i,1) = (E22_ri(i,1)+E11_ri(i,1))/2;

end

end
```

C.7 ur.m

```
function [UR_C,UR_S] = ur(epsilon_cul,...,betongmat);

%Finner betong utnyttelse
%Velger materialmodell
    if betongmat == 1
        UR_C = 100*(epsilon_c/epsilon_cu3);
        if epsilon_c >= 0
            UR_C = 0;
        end
    elseif betongmat == 2
        UR_C = 100*(epsilon_c/epsilon_cu2);
        if epsilon_c >= 0
            UR_C = 0;
        end
    elseif betongmat == 3
        UR_C = 100*(epsilon_c/epsilon_cul);
        if epsilon_c >=0
            UR_C = 0;
        end
    end
end
```

D Symbolliste

Symbol forklaring for programmet med tilsvarende tegn fra teorien. Rekkefølge etter hvor de er funnet i koden.

MATLAB betegnelse	Teori	Beskrivelse
mx	M_x	Moment x retning
my	M_y	Moment y retning
mxy	M_{xy}	Torsjonsmoment
nx	N_x	Aksialkraft x retning
Ny	N_y	Aksialkraft y retning
Nxy	N_{xy}	Skjærkraft i xy retning
R	R	Last vektor
z_s	z_j	Armerings arm
h	h	Tverrsnittshøyde
C1	C1	Avstand fra overside til armering
C2	C2	Avstand fra underside til armering
delta_h	Δh	Lagtykkelse
n	n	Antall lag
z	z_i	Betongarm
v	ν	Poisson's tall
alpha_cc	α_{cc}	Betongfasthetsfaktor
gamma_c	γ_c	Betongfastetsfaktor
f_cd	f_{cd}	Dimensjonerende betongtrykkfasthet
f_ck	f_{ck}	Karakteristisk betongtrykkfasthet
epsilon_c1	ϵ_{c1}	trykktøyning i betongen ved største spenning
epsilon_c2	ϵ_{c2}	trykktøyning i betongen ved største spenning
epsilon_c3	ϵ_{c3}	trykktøyning i betongen ved største spenning
epsilon_cu1	ϵ_{cu1}	tøyningsgrense for betong i trykk
epsilon_cu2	ϵ_{cu2}	tøyningsgrense for betong i trykk
epsilon_cu3	ϵ_{cu3}	tøyningsgrense for betong i trykk
betongmat		Materialmodell for betongen
E	E_{cm}	Elastetsmodul for betong

C	C_i	Materialmatrise for betong
gamma_s	γ_s	Materialfaktor for armeringsstål
f_sd	f_{sd}	Dimensjonerende fasthet armeringsstål
f_sk	f_{sk}	Karakteristisk fasthet armeringsstål
epsilon_s02	ϵ_u	tøyning i armering ved største last
E_sint	E_s	Elastetsmodul for armering
C_s	C_{sxj}, C_{syj}	Materialmatrise for armering
As	A_{sxj}, A_{syj}	Armering
ks		Maks verdi fra Maksdiff (2.5.13)
imax		Gjeldende iterasjon
fctm	f_{ctm}	Strekkfasthet for betong
epsilon_cs		Makstøyning i strekk for betong
n_eksp	n	EkspONENT for idealisert betong materialmodell
beta	β	Konvergens kriterie
n_iterasjoner		Grense for antall iterasjoner
K_c0	K_{c0}	Betongstivhetsmatrise
K_s	K_s	Armeringstivhetsmatrise
K_0	K	Stivhetsmatrise
epsilon_t0	ϵ_{t0}	Tøyninger i skallets midtplan
epsilon_0i	ϵ_{0i}	Tøyning i hvert betonglag
theta_i	θ_i	vinkel til hovedtøyningsretningen
T	T	Transformasjonsmatrise
epsilon_p0i	ϵ_{p0i}	Tøyning i hovedretning for hvert betonglag
sigma_p0i	σ_{p0i}	Betongspenning i hovedretning
f_cm	f_{cm}	Middelverdi av betongens sylindertykkfasthet
eta	η	Faktor for ikke-linære materialmodell betong
k	k	Faktor for ikke-linære materialmodell betong
tverrmatr		Innstilling for strekkfasthet betong
C_p	C_p	Materialmatrise for betong i hovedtøyningsretning
sigma_c	σ_{c0i}	Betongspenninger i xy retning
epsilon_0j	ϵ_{0j}	Tøyning i armering
epsilon_uk	ϵ_{uk}	Tøyningsgrense for armering
epsilon_ud	ϵ_{ud}	Redusert tøyningsgrense for armering

σ_{s0j}	σ_{s0j}	Spenning i armeringen
stalmat		Materialmodell for armeringen
S_s	S_0	Armeringsbidrag til den interne kraftvektoren
S_{0c}	S_0	Betongbidrag til den interne kraftvektoren
Maxdiff	Maxdiff	Maksimum relativ forskjell mellom eksterne og interne spenningsresultanter
UR_c	UR_c	Utnyttelse ratio betong
V_x	V_x	Skjærkraft i x retning
V_y	V_y	Skjærkraft i y retning
d	d	Tverrsnittets effektive høyde
z	z	Indre momentarm
b_w	b_w	Minste tverrsnittsbredde i strekksonen
As_{vx}	As_{vx}	Skjærarmering i x retning
As_{vy}	As_{vy}	Skjærarmering i y retning
S_x	S_x	Senteravstand mellom skjærarmering i x retning
S_y	S_y	Senteravstand mellom skjærarmering i y retning
V_a	V_a	Skjærkraft i ekvivalent bjelkeretning
N_a	N_a	Membrankraft i ekvivalent bjelkeretning
M_a	M_a	Moment i ekvivalent bjelkeretning
A_{sa}	A_{sa}	Armeringsareal i ekvivalent bjelkeretning
k	k	Faktor for skjærkraftkapasitet
ρ_l	ρ_l	Armeringsforhold for strekkarmering
σ_{cp}	σ_{cp}	Midlere trykkspenning
k_1	k_1	Faktor for skjærkapasitet
C_{Rdc}	C_{Rdc}	Faktor for skjærkapasitet
v_{min}	v_{min}	Faktor for minste verdi knyttet til hovedstrekkbrudd
V_{Rdc}	V_{Rdc}	Skjærkapasitet
UR_{vc}	UR_{vc}	Skjærutnyttelse

Tabell 8: Symbolliste

Alle variabler med endelse $_{tot}$ er de samme, bare at de lagres over alle iterasjoner. Det er flere variabler i programmet, disse er ikke direkte knyttet til teorien, men er midlertidige variabler for å kjøre koden.