

Forbedring og kvalitetssikring av programmet fap2D

Kristian Pedersen

Master i ingeniørvitenskap og IKT

Innlevert: juni 2013

Hovedveileder: Kjell Magne Mathisen, KT

Medveileder: Kolbein Bell, KT

Norges teknisk-naturvitenskapelige universitet
Institutt for konstruksjonsteknikk



MASTEROPPGAVE 2013

FAGOMRÅDE: Konstruksjonsteknikk/ Programvareutvikling	DATO: 10.06.2013	ANTALL SIDER: 126
---	---------------------	----------------------

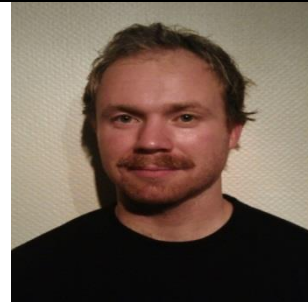
TITTEL:

Forbedring og kvalitetssikring av programmet fap2D

Validation and quality assurance of program **fap2D**

UTFØRT AV:

Kristian Pedersen



SAMMENDRAG:

Denne rapporten presenterer arbeidet lagt ned i videreutvikling av programmet **fap2D**, et program for statisk og dynamisk analyse av plane rammekonstruksjoner.

Denne rapporten er skrevet for å gi dokumentasjon over ny og forbedret implementasjon og vise forskjellen fra den nåværende versjonen og den den versjonen som fantes når arbeidet begynte.

Det egentlige produktet av denne oppgaven er selve programmet **fap2D**. Videre utviklere kan bruke denne rapporten for å se hva som er den nåværende status på programmet, og rapporten gir informasjon om hva som trengs av videre utvikling.

For å gi en oversikt over alt arbeidet som er lagt ned våren 2013 er det lagt ved en logg fra utvikleren som gir en kronologisk oversikt av arbeid utført.

FAGLÆRER: Kjell Magne Mathisen

VEILEDER(E): Kolbein Bell

UTFØRT VED: NTNU

MASTEROPPGAVE 2013

for

Kristian Pedersen

Forbedring og kvalitetssikring av programmet **fap2D**

*Validation and quality assurance of program **fap2D***

Basert på utført prosjektoppgave går oppgaven ut på å videreføre utviklingen av **fap2D**, et program for statiske og dynamiske beregninger av plane rammekonstruksjoner som ble påbegynt allerede i 2006. Frem til 2012 har 8 studenter deltatt i utviklingen av programmet, som prosjekt- og masteroppgaver. I inneværende semester er dette en av tre oppgaver knyttet til utviklingen av **fap2D**.

Fokus i denne oppgaven er programmets kvalitet, ikke nye funksjoner, og målet er et korrekt, stabilt og robust program. Stikkord i denne sammenheng er: uttesting, validering, lagring, «undo/redo», «settings» («customize/options») og hjelp-funksjon via «tooltip».

I den grad tiden tillater skal det også vurderes/implementeres installasjonsprosedyrer som kontrollerer tilgjengelighet/varighet av «lisens», spesielt med tanke på studentbruk.

Opgaven forutsettes koordinert med oppgavene til Erstad og Aasmundrud, slik at det blir valgt mest mulig enhetlige løsninger hva angår både koden og det visuelle.

Besvarelsen er først og fremst selve programmet, men det skal også utarbeides en kortfattet rapport som beskriver problemstillinger, utfordringer og valgte løsninger. En oversikt over problemer som er løst med korrekt resultat bør inngå, og det samme bør eventuelle problemer som ikke er løst tilfredsstillende. Avslutningsvis bør rapporten inneholde en kritisk vurdering med angivelse av uløste/mangelfullt løste problemer. For å kunne bedømme innsatsen er det viktig at rapporten omfatter en kronologisk "logg" som angir hva som er gjort og når; loggen plasseres i et vedlegg. Kandidaten bør også ha i mente at en av rapportens viktigste oppgaver er å sette andre i stand til å gå inn i koden og eventuelt modifisere/videreføre den.

Opgaven lar seg vanskelig innordne den vanlige "malen", men så langt råd bør besvarelsen organiseres i henhold til gjeldende retningslinjer. Selv om rapporten blir relativt kortfattet stilles det de samme formelle krav til den som for masteroppgaver generelt.

Hovedveileder: Kjell Magne Mathisen

Medveileder: Kolbein Bell

Besvarelsen skal leveres til Institutt for konstruksjonsteknikk innen 10. juni 2013.

NTNU, 18. januar, 2013

Kjell Magne Mathisen

Summary

This rapport presents the work done in further development of the program **fap2D**, a program for static and dynamic analysis of 2D frame type structures.

This rapport is written to give proper documentation of new and improved implementation and show the difference from the current version and the version that existed when this work began.

The real product of this work is the program **fap2D** itself. Future developers can use this rapport to see the current status of the program and the rapport gives information of what is needed of future development.

To give an overview over the work that has been done during the spring 2013 there is a log attached from the developer that gives a chronological overview of the work that has been done.

Forord

I 2006 startet arbeidet med å utvikle et nytt program for beregninger av plane rammekonstruksjoner. Dette programmet skulle kunne regne på statiske og dynamiske problemer, ved bruk av lineær og ikke-lineær teori. Programmet fikk arbeidstittelen **fap2D**. Sverre Holst og Magnus Minaas la planer om et slikt program høsten 2006 sammen med Kolbein Bell, og våren 2007 startet implementasjonen av ideene som ble utviklet høsten 2006.

I 2008/2009 fortsatte Dagfinn Dale Kloven og Gunnar Stenrud Nilsen arbeidet med **fap2D**. Gjennom prosjektoppgave høsten 2008 og våren 2009 startet de fra begynnelsen, hvor de brukte ideer utviklet av Holdt våren 2007.

Høsten 2009 arbeidet Jan Kristian Dolven og Fredrik Lasen med sitt prosjektarbeid som skulle brukes i arbeidet med deres masteroppgave våren 2010. Løsninger utviklet denne høsten ble implementert av Dolven i våren 2010.

Høsten 2010 startet Daniel Aase og Brita Årvik med sin prosjektoppgave, hvor de la planer for videre utvikling av **fap2D**. Våren 2011 implementerte de disse planene, samtidig som Fredrik Larsen arbeidet med å implementere løsninger for dimensjonering av komponenter i stål.

Høsten 2012 begynte Kristian Pedersen, Frans Erstad og Erik Aasmundsrud sine prosjektoppgaver, hvor de planla videre utvikling av programmet sammen med Kolbein Bell. Våren 2013 har bestått i å gjennomføre planene som ble lagt i høsten 2012, og implementere løsningene de kom fram til. Pedersen sitt arbeid har bestått i å rette feil og mangler i programmet og få **fap2D** til å framstå som et mer stabilt rammeprogram. Erstad har implementert ny funksjonalitet med last-tog, og Aasmundsrud har implementert løsninger for beregninger på jordskjelv.

Jeg vil gjerne takke Kolbein Bell for veiledningen gjennom både prosjektoppgaven høsten 2012 og masteroppgaven våren 2013. Bell har vært meget delaktig i prosessen med videre utvikling av **fap2D**, og har lagt ned et viktig arbeid med kommentarer og nye og forbedrede beregningsrutiner.

Jeg vil og takke Erstad og Aasmundsrud for et flott samarbeid gjennom både prosjektoppgave og masteroppgave. De har vært delaktige i utvikling i alle aspekter av programmet, og uten de ville ikke **fap2D** vært like godt utviklet som det er idag.

Jeg vil og takke de tidligere utviklerne for innsatsen de har lagt ned i implementasjon av kode, og dokumentasjonen av tidligere arbeid.

Trondheim, Juni 2013

Kristian Pedersen

Innholdsfortegnelse

Summary	v
Forord	vii
1. Introduksjon	1
2. Definisjoner og terminologi	3
2.1 Rapporten	3
2.1.1 Formatering	3
2.2 Programkoden	3
2.2.1 Formatering	3
2.3 Programmet	4
2.3.1 Konstruksjonsmodell	4
2.3.2 Beregningsmodellen	4
2.3.3 Brukergrensesnittet	4
3. Arbeidsmetoden	7
3.1 Planlegging	7
3.1.1 Prosjektoppgave høsten 2012	7
3.1.2 Hovedoppgaven våren 2013	7
3.2 Utvikling	7
3.3 Møtevirksomhet og samarbeid	7
3.4 Rapportering	8
4. Funksjonalitet i fap2D	9
4.1 Generelt	9
4.1.1 Eksport av bilder	9
4.1.2 Print	10
4.1.3 Loggfil	10
4.1.4 Toolbox	11
4.1.5 Zoom	11
4.1.6 Åpne samme modell	12
4.1.7 Oppstart av programmet	12
4.1.8 Tastaturnarveier	13
4.1.9 Undo/Redo	13
4.1.10 ComputationalModel	14
4.1.11 Error/Information	14

4.1.12 Tooltips	14
4.1.13 Lagring	15
4.1.14 Dialogbokser	15
4.1.15 Antall repaints	15
4.2 Modeling.....	16
4.2.1 Materialer	16
4.2.2 Spring.....	16
4.2.3 Point mass	17
4.2.4 Manage meshing	17
4.3 Loading	19
4.3.1 Punktlast.....	19
4.3.2 Linjelast.....	19
4.3.3 Foreskrevne forskyvninger	19
4.3.4 Initiell tøyning/temperatur	19
4.3.5 Lastkombinasjoner	21
4.3.6 View loading	23
4.4 Analysis.....	24
4.4.1 Linear static	24
4.4.2 Influence lines.....	24
4.4.3 Real time analysis	24
4.4.4 Buckling	24
4.4.5 Non linear static	25
4.4.6 Free vibration	27
4.4.7 Dynamic time domain	28
4.4.8 Dynamic frequency domain	30
4.4.9 Earthquake time integration	31
4.4.10 Earthquake response spectrum	31
4.5 Results	32
4.5.1 Axial and shear stresses.....	32
4.5.2 Section forces	33
4.6 Popup menu	36
4.6.1 Joint	37
4.6.2 Lineload	42

4.6.3 Point load.....	43
4.6.4 Member	43
5. OpenGL.....	49
5.1 Joints.....	49
5.2 Members	50
5.3 Linjelaster	51
5.2 Finne-metoder.....	52
5.3 Results	55
6. Videre arbeid.....	57
6.1 Lagring av innstillinger.....	57
6.2 Lagring av resultater.....	57
6.3 Dialogbokser.....	57
6.4 Lisensiering.....	57
6.5 Undo/Redo	57
6.6 Tegning	57
6.7 Member data panel.....	58
6.8 Beta-testing	58
6.9 Kopiering	58
6.10 Stabilitet og sikkerhet.....	58
6.11 Slå user controls sammen	58
6.12 Nummerering av members	58
6.13 Events for mus.....	59
7. Referanseliste.....	61
A. Logg.....	63
B. Programvare brukt	87
B.1 Operativsystem	87
B.2. Uviklingsmiljø	87
B3. Tilleggsprogramvare til Visual Studio 2012.....	87
B.4 Tekstbehandling	87
B5. Annet	87
C. SVN	89
D. Kodekonvensjoner	95
E. Forklaring av koden	103

E.1 Prosjekter	103
E.2 Datasett	104
E.3 Code metrics.....	107
F. Innhold på cden	109
F.1 Dokumentasjon	109
F.2 Hovedoppgaven.....	109
F.3 Installasjon.....	109
F.4 Kildekode	109
F.5 Krav	109
F.6 Prosjektoppgave	109
G. Krav til hardare og software	111
G.1 Krav til hardware.....	111
G.2 Krav til software	111
H. Units	113
H.1 GUI	113
H.2 Input til Frame2D	113
H.2 Output fra Frame2D.....	114

1. Introduksjon

Under prosjektoppgaven høsten 2012 ble det avdekket feil og mangler ved **fap2D** som trengtes utbedring ved intensiv bruk av programmet. Det ble utarbeidet en oversikt over arbeid som behøvdes gjennomført.

Denne rapporten oppsummerer statusen til programmet etter endt utvikling våren 2013. Den skal være med å gi nye utviklere en innsikt i status av programmet per dags dato, tanker rundt videre arbeid som trengs mer utvikling og dokumentere arbeidet som er lagd ned i programmet. Oppgaven starter med en kort innføring i definisjoner og terminologi (kapittel 2). Kapittel 3 beskriver hvordan arbeidet har blitt gjennomført.

Kapittel 4 tar for seg endringer i brukergrensesnittet til **fap2D**. Det er delt inn i seks mindre kapitler som systematisk går gjennom programmets funksjonalitet, hvor det blir vist fram små og store endringer i brukergrensesnittet.

Kapittel 5 går gjennom endrede tegnemetoder. Det er delt inn fem mindre kapitler som systematisk går gjennom de forskjellige metodene som er blitt endret.

Kapittel 6 gir forslag til videreutvikling og forbedringer til programmet. Dette kapitlet går gjennom ting som er utbedret, men som fortsatt ikke er perfekt implementert. Det er og noen nye ting som har dukket opp underveis, noe som det ikke har vært tid til denne våren og saker som henger igjen etter hovedoppgaven til Dolven[1] og Aase og Årvik[2].

Loggen er vedlagt i appendiks A. Denne loggen er et dokument som beskriver kronologisk utført arbeid. Loggen viser mer i detalj hva som er blitt gjort, hvorfor valg er blitt gjort og feil som har dukket opp underveis. Loggen kan være til nyttig bruk til senere utviklere som kan gi forklaring til hvorfor valg har blitt gjort.

Det er lagt ved en liten gjennomgang av koden til senere utviklere i appendiks F. Etter denne masteroppgaven er det mye kode som kan ta tid for nye utviklere å sette seg inn i. Ved å forklare hvordan koden er bygd opp kan det gi senere utviklere et fortrinn, og kan gjøre det enklere for de å sette seg raskere inn i koden.

Denne oppgaven bygger i stor grad på arbeid fra tidligere arbeid. Det er lagt stor vekt på at **fap2D** skal bli stabilt og sikkert nok til at det kan brukes daglig av studenter og ansatte ved NTNU til beregning av plane rammekonstruksjoner.

2. Definisjoner og terminologi

Dette kapitlet gir en oversikt over terminologien som er brukt i rapporten(2.1), i programkoden(2.2) og i programmet(2.3).

2.1 Rapporten

2.1.1 Formatering

Tabell 1 viser typen tekst som er brukt gjennom hele rapporten. Kolonnen til venstre viser hva slags font som blir brukt, og kolonnen til høyre gir en beskrivelse for hver av de fem typene.

Type font	Beskrivelse
Normal tekst	Denne typen skrift er brukt for det meste i rapporten
<i>Kursiv tekst</i>	Denne typen skrift er brukt til engelske ord og uttrykk som ikke har noen god oversettelse
Maskinskrift	Denne typen skrift er brukt på programkode
Uthevet tekst	Denne typen skrift er brukt på ting som er hentet fra grensesnittet til fap2D
[tall i klammer]	Denne typen brukes til å referere til et vedlegg i referanselisten

Tabell 1 - Formateringer brukt i rapporten

2.2 Programkoden

2.2.1 Formatering

Vedlegg D viser et dokument som er blitt benyttet av utviklerne av **fap2D** våren 2013. Det beskriver kodekonvensjoner og hvordan all programkode skal bygges opp. Selve navnekonvensjonene er blitt bevart fra de tidligere utviklerne, men er introdusert en streng skriveform som brukes på all kode. Slik blir det enklere for senere utviklere å sette seg inn i koden, og å holde det konsistent gjennom hele programmet gjør det mer oversiktlig og enklere å finne frem til den koden man vil navigere til. Tabell 2 viser navnekonvensjoner som er brukt i programkoden.

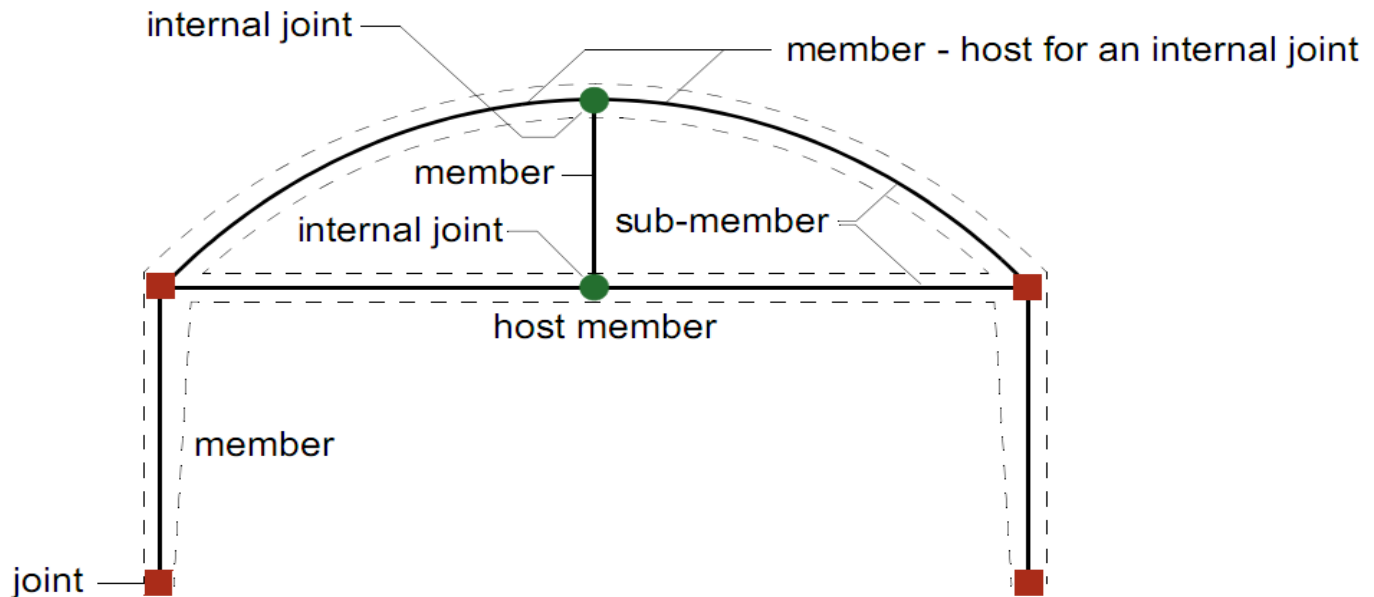
Navnekonvensjon	Bruksområde
camelCasing	Brukes for å navngi variabler som ikke er private
<code>_camelCasing</code>	Brukes for å navngi variabler som er private
KONSTANT	Konstante verdier navngis med kun store bokstaver
PascalCasing	Brukes for alle klasser, typer, metoder og navnerom. Samtidig for navn som består av flere ord
PrePascalCasing	Valgfri prefiks brukes for å indikere at klasser arver fra en klasse i standardbiblioteket. Prefikset angir hvilken klasse de arver fra

Tabell 2 - Navnekonvensjoner brukt i koden

2.3 Programmet

2.3.1 Konstruksjonsmodell

En konstruksjonsmodell i **fap2D** består av *members* og *joints*. *Members* kan deles inn i *sub-members*, og *joints* som plasseres på en *member* kalles *internal joints*. Figur 1 viser en inndeling av disse. Figuren er hentet fra prosjektoppgaven til Dolven og Larsen[3]. For mer informasjon om konstruksjonsmodellen henvises til brukermanualen til **fap2D**[4].



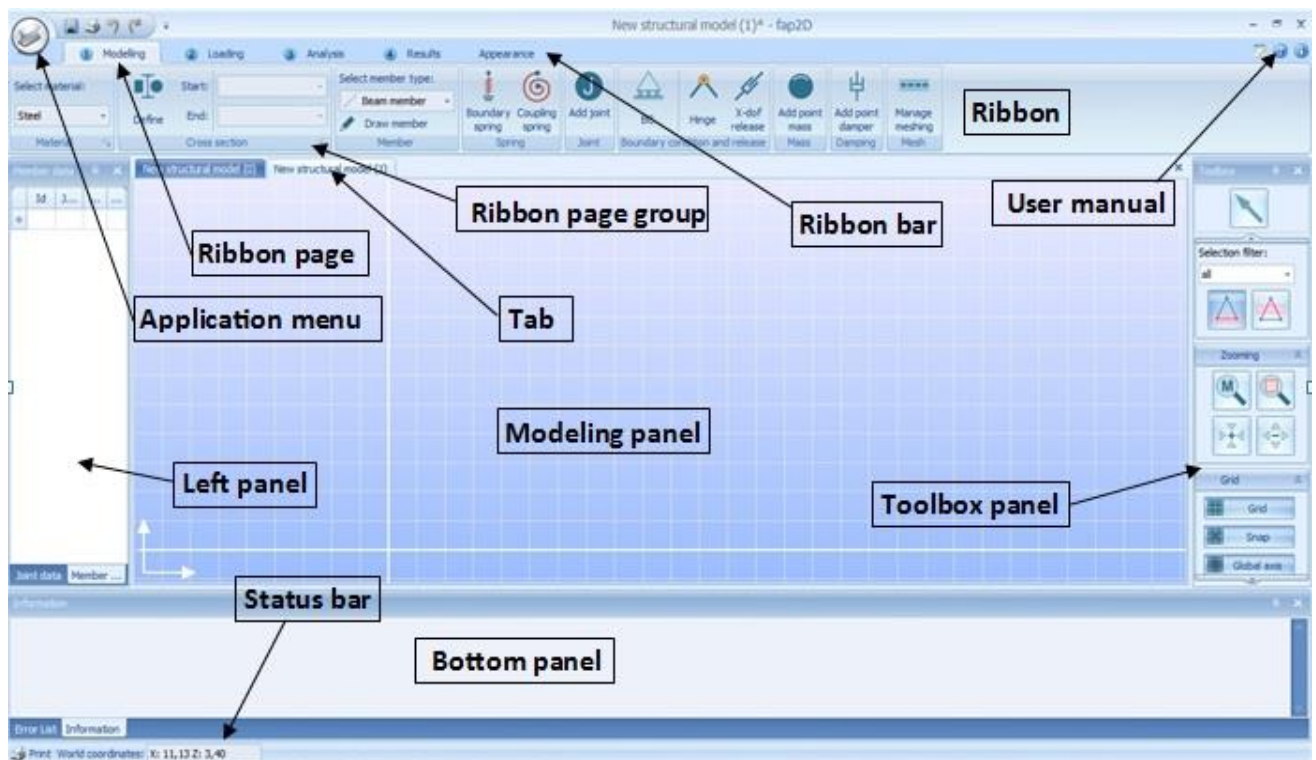
Figur 1 - *Members, joints, sub-members og internal joints*

2.3.2 Beregningsmodellen

Brukeren forholder seg til en konstruksjonsmodell, og denne blir konvertert til en beregningsmodell som kan kjøre flere analyser, og produsere ønsket resultat. For mer informasjon om beregningsmodellen henvises til brukermanualen til **fap2D**[4].

2.3.3 Brukergrensesnittet

Brukergrensesnittet er basert på Microsoft sin *ribbon*-funksjonalitet som brukes i Microsoft Office[5]. Se figur 2 for bilde av programmets åpningskjermet etter en ny modell har blitt laget.



Figur 2 - Skjermbilde av fap2D januar 2013

Når programmet åpnes befinner brukeren seg i **Modeling**-fanen, så beveger brukeren seg til høyre til **Loading**, **Analysis** og **Results**. Denne metoden er kjent for de fleste som har brukt Microsoft-produkter de seneste årene. Dette gjør at det blir intuitivt og oversiktlig. En *ribbon page* blir i denne rapporten omtalt som en fane. For mer informasjon og oversikt over brukergrensesnittet henvises det til brukermanualen til **fap2D** og prosjektoppgaven til Pedersen[6].

3. Arbeidsmetoden

3.1 Planlegging

3.1.1 Prosjektoppgave høsten 2012

Høsten bestod stort sett i å gjøre seg kjent med **fap2D**, både kode og brukergrensesnitt.

Feilmeldinger ble samlet inn av alle som var knyttet til utvikling av programmet; Kristian Pedersen, Erik Aasmundsrud, Frans Erstad og Kolbein Bell. Disse registrert i prosjektoppgaven, og det ble gjort en vurdering av hva slags prioritet de forskjellige skulle ha. Hovedoppgavene til Dolven[1] og Aase & Årvik[2] ble og studert for å se hva de mente som var viktig arbeid videre.

3.1.2 Hovedoppgaven våren 2013

Denne hovedoppgaven er en oversikt og oppsummering av hva som er blitt gjort og hva som er blitt implementert i **fap2D** våren 2013 undertegnede. Og den vil forhåpentligvis bli til nyttig hjelp for nye studenter som ønsker å arbeide videre med **fap2D**.

3.2 Utvikling

Våren 2013 har det vært fire personer som har vært med på utviklingen av **fap2D**: Kristian Pedersen, Erik Aasmundsrud, Frans Erstad og Kolbein Bell. Undertegnede har utbedret feil som er avdekket og har arbeidet med å rydde opp i og skrive om eksisterende kode. Det har vært lite implementering av ny funksjonalitet, da høyeste prioritet har vært og få til robust og stabilt program. Erstad har implementert lasttog og store deler av funksjonaliteten forbundet med dette. Aasmundsrud har implementert jordskjelvanalyser, i tidsplanet og responsspektrum. Bell har hatt ansvaret for å utvikle metoder for å regne på disse nye analysene, forbedre de andre analysene og har vært til stor hjelp med å utvikle **fap2D** i riktig retning.

Det er mye ny kode og mye kode som er blitt endret under denne prosessen. Og der det har vært problemer med å finne gode og enkle løsninger har forumet til Visual Studio[7] vært til god hjelp. Her går det an å stille spørsmål til et forum fullt av dyktige utviklere, og komme frem til gode løsninger sammen.

3.3 Møtevirksomhet og samarbeid

Vi har hatt ukentlige møter for å diskutere progresjonen i arbeidet med **fap2D**. Her har vi lagt fram hva som er blitt siden sist, diskutert implementasjonene, hva som må gjøres fremover og kommentert hverandres arbeid. Disse møtene har vært til stor hjelp for alle sammen. Å få ukentlig tilbakemeldinger på arbeidet som er blitt lagt ned har vist seg å være essensielt for den videre utviklingen av **fap2D**.

Pedersen, Aasmundsrud og Erstad har og hatt kontinuerlige samtaler via Facebook[8] og Skype[9]. Her har det blitt diskutert alt som er implementert og kommet med kommentarer over andres arbeid. Det har og vært til stor nytte når en av utviklerne har sittet fast, og de andre kan da hjelpe til og snakke sammen i «real-time» samtidig som debugging foregår. Det ble og opprettet et dokument på Dropbox[10], Rapporterte Feil, hvor det er blitt skrevet inn alle feil som er oppdaget høsten 2012 og gjennom utviklingen våren 2013. Her står det hva slags feil det er, hvem som har funnet den, beskrivelse av feilen, når den er funnet, hvem som eventuelt har fikset feilen og hvilken dato den er blitt fikset på.

Det har og vært samtaler over e-post mellom alle fire. Dette har vært nyttig når noe som skal deles mellom alle fire ikke kan vente til neste møte. Ved behov har en også hatt egne møter med veileder.

Møtevirksomhet og samarbeid over sosiale media og e-post har gjort arbeidsprosessen langt enklere og har bidratt til et godt samhold innad i gruppa.

3.4 Rapportering

Gjennom hele våren har alle de tre studentene ført en egen logg(Vedlegg A). Her har dagens arbeid blitt registrert, tanker rundt dette og videre arbeid som må gjøres. Loggens intensjoner er å gjøre rede for hva som er blitt gjort, hvorfor valg har blitt gjort og den har også fungert som et lite forum hvor en kan lufte frustrasjoner og diskutere med seg selv. Loggen har gjort arbeidet mer oversiktlig og det har vært en fin måte å følge opp eget arbeid på.

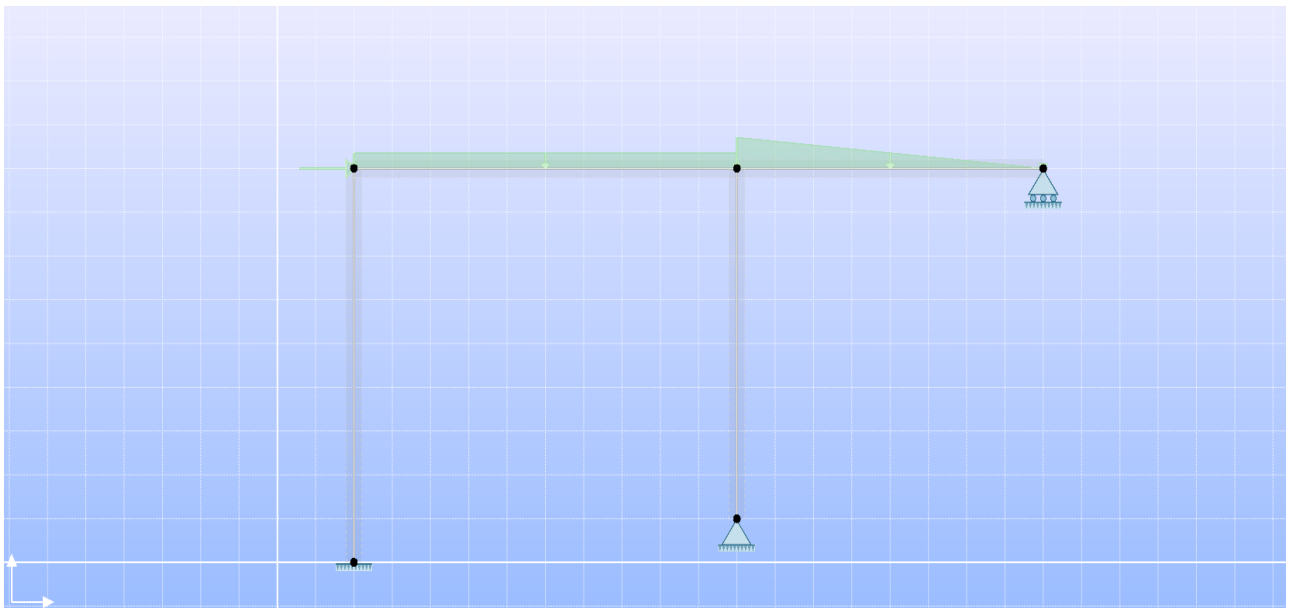
4. Funksjonalitet i fap2D

Mye av arbeidet denne våren har bestått av opprydding i gammel kode som ikke fungerte eller som fungerte dårlig. Målet var et bedre og mer robust program som kan benyttes som et pålitelig verktøy til beregninger av rammer og fagverk.

4.1 Generelt

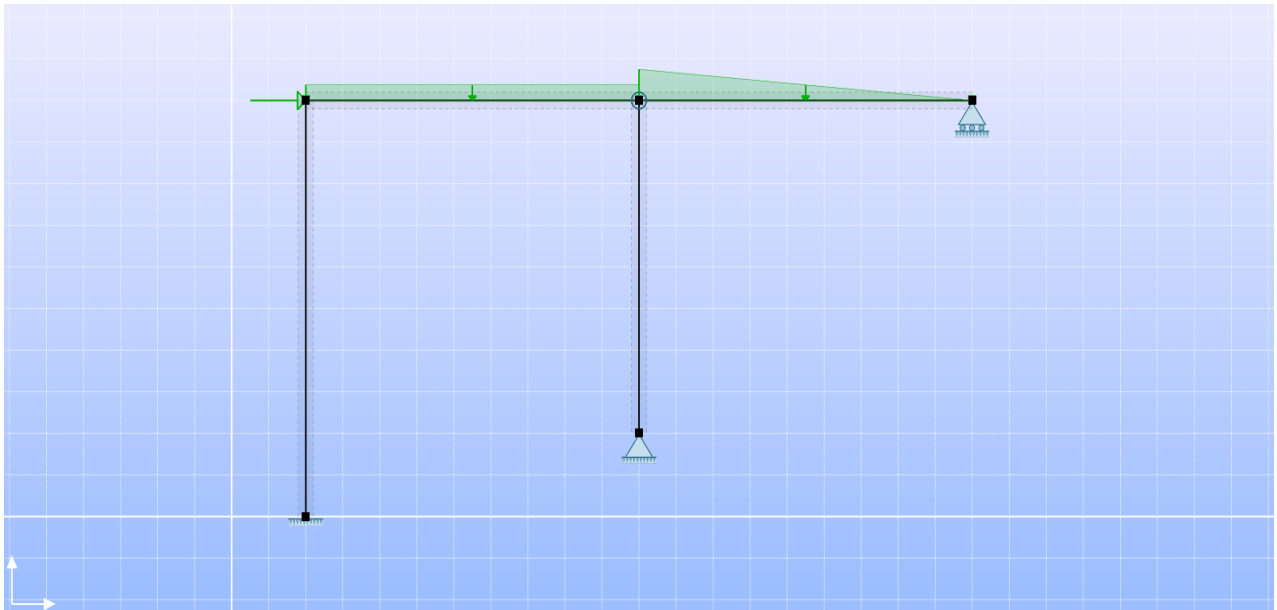
4.1.1 Eksport av bilder

En funksjon som fungerte dårlig var **Export model to image**, som ligger under applikasjonsmenyen. Denne henter ut modellen som vises i vinduet og lagrer den på harddisken som en bildefil. Fra før kunne det kun lagres i PNG-format. Nå er det mulig å lagre modellen i bildeformatene PNG, JPG, BMP, GIF og TIFF.



Figur 3 - Ramme eksportert i versjon 2.0

Figur 3 viser en ramme som er eksportert etter den gamle metoden. Laster og *members* blir litt gjennomsliktige. Figur 4 viser samme modellen eksportert ved den nye metoden. Bildet er mye klarere og alt som var gjennomsliktig før er ikke det nå lengre. OpenGL bruker RGBA-verdier når modellen blir tatt bilde av. Og den gamle metoden fjernet ikke alpha-verdien fra bildet før det ble lagret, denne alpha-verdien er den verdien som står for gjennomsliktighet (*transparency*) i et bilde. Når denne verdien ble tatt vare på og lagret i PNG-format så ble bildene uklare, siden PNG-format lagres sammen med *transparency*, mens f.eks. JPG ikke gjør det.



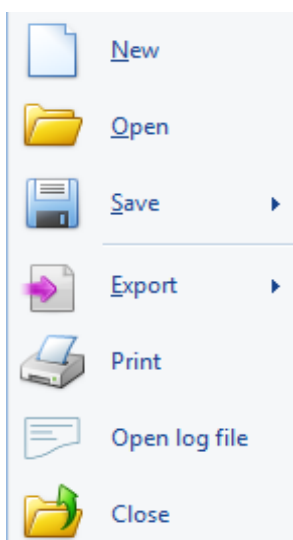
Figur 4 - Ramme eksportert i versjon 3.0

4.1.2 Print

Under applikasjonsmenyen finnes en knapp som heter **Print**. Denne hadde ingen funksjonalitet i versjon 2.0, det har den i versjon 3.0. Den printer ut modellen vist i modellvinduet på et A4-ark i liggende format. Størrelsen blir tilpasset slik at det blir printet helt ut til sidene av arket, mens størrelsesorden bredde/høyde blir tatt vare på. Da blir ikke bildet som printes ut «dratt» noe og det blir fint og klart.

4.1.3 Loggfil

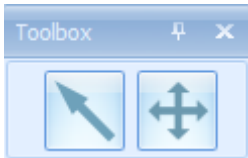
I versjon 2.0 av **fap2D** lå det en knapp under applikasjonsmenyen som heter **Import**. Denne knappen hadde ingen funksjon og ideen bak den er noe uklar. Denne er fjernet og isteden ligger det nå en knapp der som heter **Open log file**. Den åpner opp loggfila i programmet Notepad[11]. Figur 5 viser den nye applikasjonsmenyen med **Import** fjernet og **Open log file** lagt til.



Figur 5 - Den nye applikasjonsmenyen

4.1.4 Toolbox

Under **Toolbox** er det lagt til en knapp som gir brukeren muligheten til å bevege modellvinduet rundt. Figur 6 viser denne nye knappen som ligger ved siden av **Reset mouse pointer**. Brukeren klikker inn denne og kan så bevege modellvinduet ved å holde inne venstre musknapp. Det er og mulig å bruke denne funksjonen uten å klikke på **Move model**-knappen ved å holde inne midtre musknapp og bevege muspekeren rundt i modellvinduet.



Figur 6 - Ny knapp under toolbox

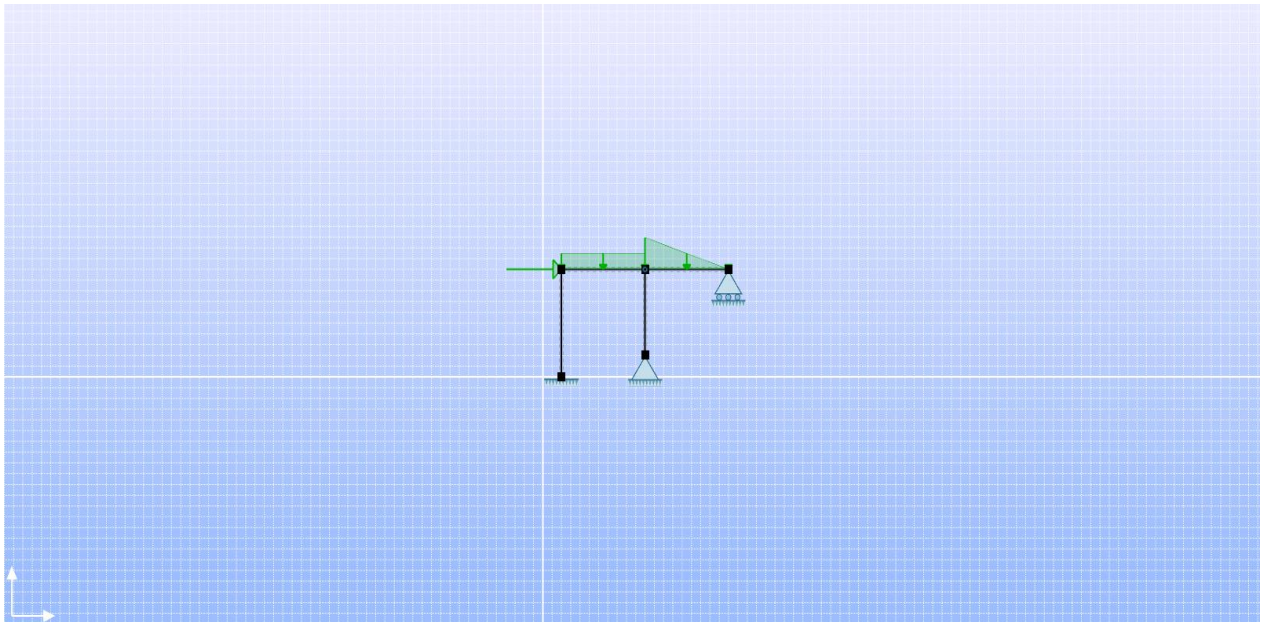
4.1.5 Zoom

Det var før ingen begrensning på hvor langt inn og ut det var mulig å zoome. Figur 7 viser rammen fra figur 3 når den er zoomet helt ut i versjon 2.0 av **fap2D**. Det er så mye zoomet ut at alt som kan sees er et opplager og noen piler. Og koordinatsystemet er visket bort.



Figur 7 - Zoom på versjon 2.0

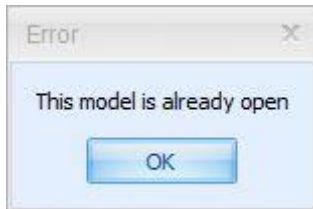
Figur 8 viser samme rammen zoomet helt ut. Det er blitt satt en verdi som det ikke er mulig å zoome lengre ut enn. Denne verdien blir bestemt ut fra hvilke størrelser man har på *grid*. Det samme gjelder når en bruker skal zoome inn.



Figur 8 - Zoom på versjon 3.0

4.1.6 Åpne samme modell

I versjon 2.0 av **fap2D** var det mulig å jobbe med en modell i flere åpne faner ved å åpne den samme modellen flere ganger. Det er lagt inn en test under åpning som sjekker om modellen en bruker forsøker å åpne allerede er åpen. Og er den allerede åpen kommer det et vindu opp som påpeker dette, se figur 9.



Figur 9 - Forsøk på å åpne en allerede åpen modell

4.1.7 Oppstart av programmet

Aasmundsrud implementerte et åpningsvindu som vises når programmet starter. Der får brukeren valgmuligheter om å åpne en modell, lage en ny modell, det er en liste med tidligere modeller der som brukeren kan klikke på for å åpne eller det går an å lukke programmet. Pga at denne er implementert ga det en mulighet til å redusere oppstarttiden til **fap2D** betraktelig. Når programmet starter er det en del som skal lastes inn. Det er særlig to funksjoner som bruker lang tid, det er `LoadCustomCursors()` og `InitSkinGallery()` som bruker 333 millisekunder og 636 millisekunder på en ganske rask pc. Ved å legge de to funksjonene inn til å lastes inn parallelt i bakgrunnen som `backgroundworkers`, som er en klasse i C#, gikk tiden de effektivt tok i lastingen av **fap2D** ned fra 969 millisekunder til 2 millisekunder.

4.1.8 Tastaturnarveier

Det er en del tastaturnarveier som brukeren kan benytte seg av i programmet. Under er det en tabell som viser alle tastaturnarveiene med funksjonaliteten knyttet til snarveiene.

Snarvei	Funksjonalitet
CTRL+T & CTRL+N	Lager en ny modell
CTRL+S	Lagrer modellen brukeren jobber med
CTRL+O	Åpne en eksisterende modell
CTRL+P	Printer bilde av modellen
CTRL+D & Del	Sletter merkede objekter i modellen
CTRL+Z	<i>Undo</i>
CTRL+Y	<i>Redo</i>
CTRL+F4	Lukker modellen
CTRL+TAB	Veksler mellom åpne modeller
CTRL+A	Merker alle objektene i modellen
ESC	Lukker en dialogboks/ Reset mouse pointer
ENTER	Trykker inn OK -knappen i dialogbokser
F1	Åpner opp brukermanualen

Tabell 3 - Tastaturnarveier

Det er og mulig å holde inne CTRL-knappen og trykke på flere objekter i modellen med muspekeren for å merke flere ting.

4.1.9 Undo/Redo

Denne funksjonen trengte en overhaling fra versjon 2.0. Det tok lang tid fra en bruker trykte på **Undo**-knappen fram til en angring var ferdig, rundt ett sekund. Dette medførte at programmet flimret litt og gjorde at det ikke føltet så responsivt. Og det var ikke alt en bruker foretok som endte opp i *undo/redo*-lista. La en til et joint fantes det ikke mulighet for å angre på dette.

Problemet med denne var at det ble konstruert 2 nye objekter av typen `FrmModel`, og dette er en ganske stor klasse. Og å konstruere et slikt objekt tar rundt 300-400 millisekunder. Å kode om slik at disse ikke ble konstruert har responstiden til en angring gått drastisk ned og flimringen har forsvunnet. Alt en bruker foretar seg med modellen blir lagt til i *undo/redo*-lista. Alt som blir lagt til, fjernet, redigert og flyttet. Dette gjør at programmet føles mer responsivt når en bruker kan angre på noe og det kun tar noen få millieskunder.

En ting som og skjer nå *undo/redo*-lista blir oppdatert, og det er at en variabel som heter `IsModified` blir satt til `True` og dermed kommer det en stjerne øverst i tittelen som sier at modellen er blitt endret. Det medfører at programmet spør om brukeren vil lagre modellen hvis den lukkes, siden den er blitt modifisert. Og en endring i modellen medfører og at fanen **Results** blir inaktiv, slik at en bruker er nødt til å kjøre analyse på nytt for å få resultater. Grunnen til det er at hvis en endrer modellen drastisk med resultater oppe, og går inn igjen for å se på resultatene så er det en mulighet for at programmet krasjer. Når det blir opprettet et objekt av typen `ComputationalModel` så tar den inn en `StructuralModel` i konstruktøren. Da peker konstruksjonsmodellen i `ComputationalModel` til samme konstruksjonsmodellen som en bruker kan endre på. Hvis denne endres drastisk kan det bli problemer med opptegningen av modellen i fanen **Results** og det er fare for krasj.

4.1.10 ComputationalModel

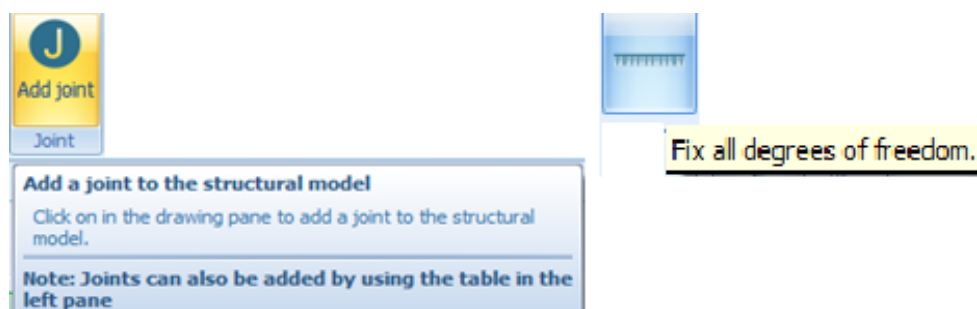
Klassen som heter `ComputationalModel` er den klassen som inneholder alt av resultater. I versjon 2.0 var det kun mulig å ha en `ComputationalModel` av gangen, da dette var et objekt som lå under hovedvinduet. Var det mange modeller åpne var det kun mulig å se på resultater av en analyse av gangen. Nå inneholder hver modell en instans av denne klassen og resultatene følger modellen. Dette medfører at det er mulig å ha et resultat per modell, istedenfor et resultat per program.

4.1.11 Error/Information

Nederst i programmet finnes det et panel som viser informasjon og *errors*. Dette fungerer veldig bra. Men det var noen av *errors* som hadde en tekst som var ganske vanskelige å tyde. F.eks. er **“Singular or near-singular stiffness matrix detected at the equation corresponding to dof 1 at node 2 during iteration 1 of load step 50”** vanskelig for en bruker å tolke. Etter god hjelp fra Bell er alle *errors* som er vanskelig å tyde skrevet om.

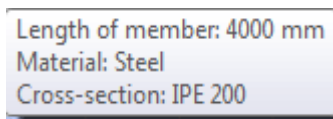
4.1.12 Tooltips

Når en bruker holder muspekeren over knapper i programmet skal det komme opp *super-tooltip* eller *tooltip* for å hjelpe brukeren. Figur 10 viser eksempel på begge, figuren er hentet fra prosjektoppgaven til Pedersen[6]. Dette har ikke vært implementert over hele programmet, og noen har vært mangelfulle. Ved god hjelp fra Bell som har skrevet teksten til alle, er dette blitt implementert over hele grensesnittet.



Figur 10 - Eksempel på *super-tooltip* og *tooltip*

Det er og blitt implementert *tooltip* for *members* i modellvinduet. Denne viser lengde, materiale og tverrsnitt når brukeren holder muspekeren rolig over en *member* i et par sekunder. Eksempel er vist i figur 11.



Figur 11 - *Tooltip for member*

4.1.13 Lagring

Det er en del mer som lagres sammen med modellen enn det gjorde i versjon 2.0. Lokale innstillinger har mange variabler, og i den gamle versjonen ble 16 av disse lagret. I versjon 3.0 er dette utvidet til 67 lokale innstillinger som lagres sammen med modellen. Elementinndelingen til *members* blir og lagret med modellen.

Det er og lagt til metoder som har gjort det mulig å lagre globale innstillinger. En bruker kan endre på noen globale innstillinger i dialogboksen **fap2D Settings** som ligger under applikasjonsmenyen. Disse blir lagret til en egen fil når brukeren klikker på **Apply**, så blir disse lagrede globale innstillingene åpnet ved start av programmet.

4.1.14 Dialogbokser

Cancel-knappene til dialogboksene som hører til fanene **Analysis** og **Results** er blitt fjernet. Det er fordi de ikke trengs på de aller fleste dialogboksene under disse fanene, og det var en del av **Cancel**-knappene som ikke hadde noen funksjonalitet. Alle dialogboksene kan lukkes ved hjelp av **Ok**-knappen, og **Esc**-knappen på tastaturet lukker i tillegg dialogbokser gjennom hele programmet.

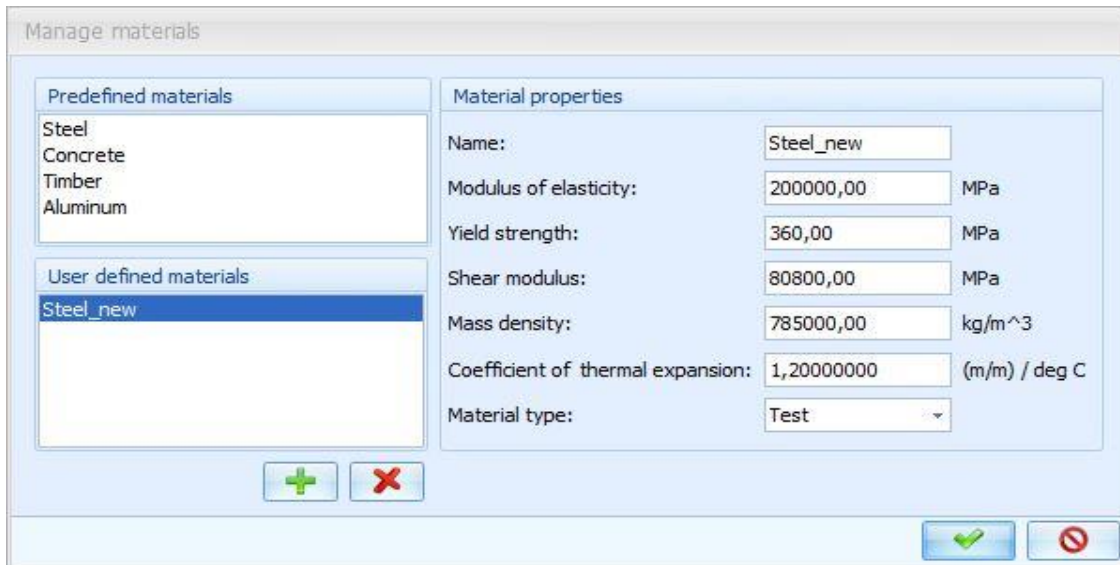
4.1.15 Antall repaints

Programmet tegner opp hele modellen ved hjelp av metoden `RePaintGLForm()`. Den ble i versjon 2.0 kalt veldig ofte, og forårsaket at programmet hadde en litt treg respons iblant. Når en bruker f.eks. åpnet programmet og trykte på **New** i applikasjonsmenyen ble denne metoden kalt 15-17 ganger, og hvis en bruker byttet mellom modeller ble samme antall *repaints* gjennomført. Redusere antall *repaints* har vært en del av jobben med å få programmet mer responsivt. Når en bruker trykker på **New** i versjon 3.0 er antall kall til `RePaintGLForm()` redusert til fire. Det gjenstår fortsatt en jobb her med å få ned antall *repaints*, men i den nye versjonen har dette antallet sunket drastisk gjennom hele brukergrensesnittet.

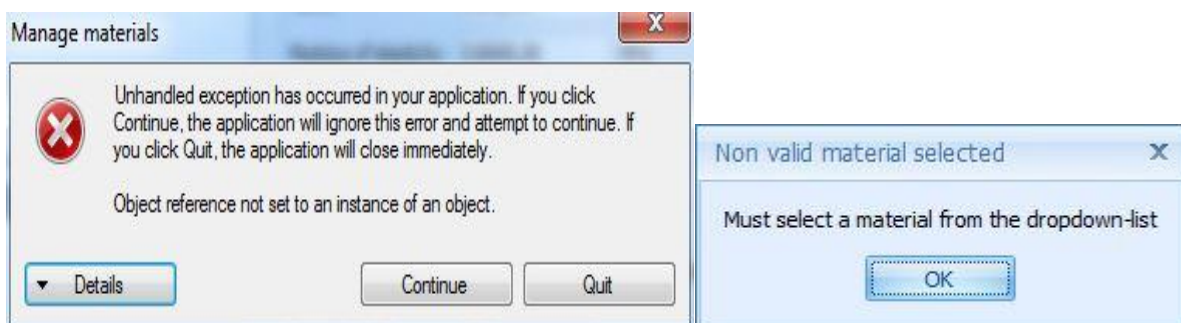
4.2 Modeling

4.2.1 Materialer

Det er mulig å skrive inn egen tekst hvor **Material type** skal velges, se figur 12. Dette medførte før en *exception*. Det er lagt inn en test her og nå kommer det istedenfor opp en boks med informasjon som sier at en bruker må velge et gyldig materiale fra listen.



Figur 12 – Dialogboks for materialer

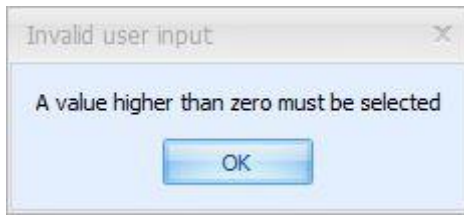


Figur 13 - Versjon 2.0 vs versjon 3.0

Figur 13 viser forskjellen mellom de to versjonene når en bruker forsøker å skrive inn en egen tekst under **Material type**.

4.2.2 Spring

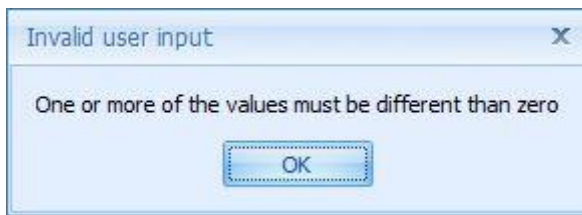
Hvis en bruker forsøker å skrive inn en negativ verdi eller verdi lik null for stivheten til en *spring* så får de en beskjed om at dette ikke er lov, se figur 14.



Figur 14 - Feil verdi for stivheten til *spring*

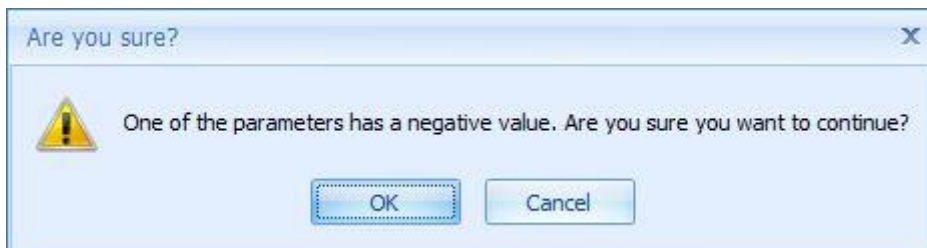
4.2.3 Point mass

Det er ikke mulig å legge til en punktmasse med verdier lik null. Da kommer det en beskjed om dette, se figur 15.



Figur 15 - Punktmasse med verdier lik null

Det er lovlig å legge til en punktmasse med negative verdier. Men siden dette er et sjeldent tilfelle og en bruker bør vite hva han holder på med hvis dette velges kommer det opp en advarsel om dette, se figur 16. Klikker brukeren **OK** legges det til punktmasse med negativ verdi, klikker brukeren **Cancel** kommer han tilbake til dialogboksen og kan endre sine verdier.

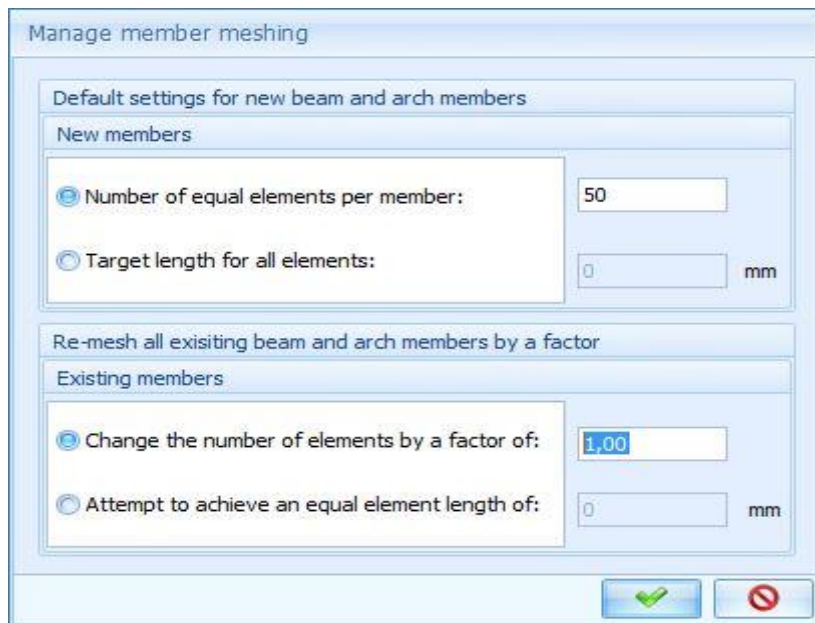


Figur 16 - Punktmasse med negative verdier

Det samme vil skje med under dialogboksen til **point damper**.

4.2.4 Manage meshing

Denne dialogboksen er blitt ny, se figur 17.



Figur 17 - Manage meshing

For nye *members* har brukeren nå to valg; hvor mange elementer hver nye *member* skal ha og hvor lang elementene skal være. Velger brukeren det første valget blir alle nye *members* inndelt med så mange elementer man velger. Velger brukeren det andre valget blir antall elementer beregnet ut fra ønsket elementlengde og lengden på ny *member*. F.eks. vil en ønsket elementlengde på 100mm og en *member* som er 5000mm lang gi 50 elementer på denne *member*.

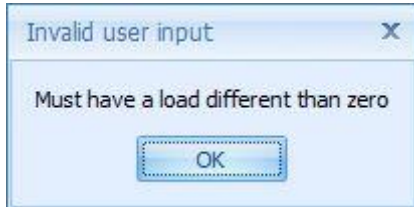
For de eksisterende *members* har man og to valg; en faktor å multiplisere alle elementinndelingene med og en ønsket elementlengde på alle elementene på eksisterende *members*. Velger brukeren den første går den gjennom alle eksisterende *members* og multipliserer elementinndelingen med den faktoren brukeren skriver inn. Velger brukeren den andre går den gjennom alle eksisterende *members* og lager en elementinndeling på lengde på *member* dividert med ønsket elementlengde.

Det er og her lagt inn sperrer slik at en bruker ikke har mulighet for å skrive inn negative verdier og elementlengde lik null. Og blir det slik at elementinndelingen til en *member* blir mindre enn to settes den til to, siden det er minimum inndeling.

4.3 Loading

4.3.1 Punktlast

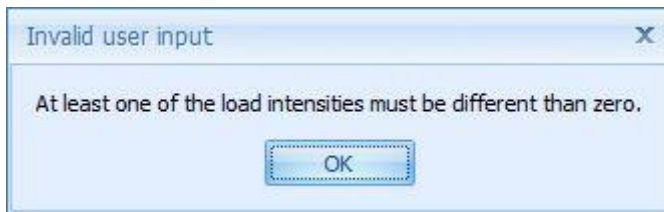
Det er lagt inn en test slik at det ikke er mulig å legge til en punktlast med verdier lik 0, da kommer beskjeden som er vist i figur 18.



Figur 18 - Forsøk på å legge inn punktlast med verdier lik 0

4.3.2 Linjelast

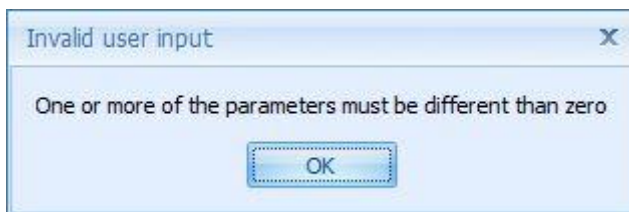
Det er lagt inn en test slik at det ikke er mulig å legge til en punktlast med intensiteter lik null, da kommer beskjeden som er vist i figur 19.



Figur 19 - Forsøk på å legge inn linjelast med intensiteter lik 0

4.3.3 Foreskrevne forskyvninger

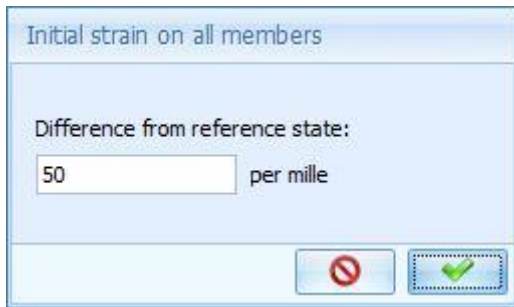
Det er lagt inn test som gir beskjed om at ikke er mulig å legge til en foreskreven forskyvning med verdier lik null, se figur 20.



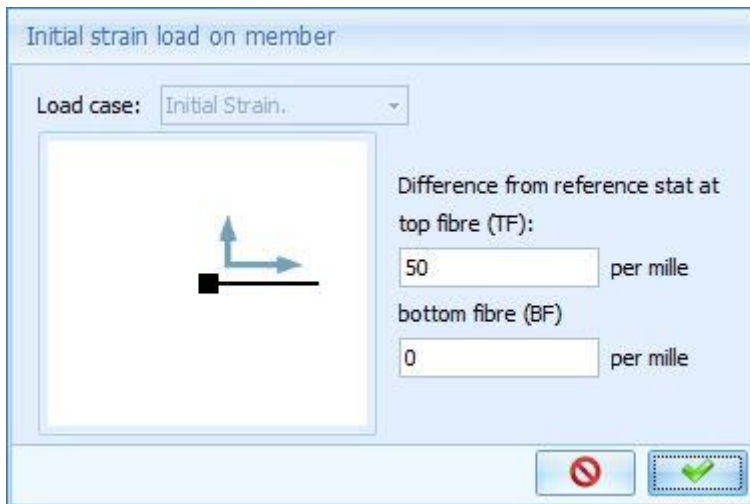
Figur 20 - Forsøk på å legge inn foreskreven forskyvning med verdier lik 0

4.3.4 Initiell tøyning/temperatur

Denne dialogboksen måtte endres en del. Hvis en bruker først la til en initiell tøyning, for så å forsøke å legge til en initiell temperatur så gikk ikke dette. Figur 21 viser et forsøk på å legge inn initiell temperatur på hele modellen når det allerede er lagt inn initiell tøyning. Da dukket dialogboksen for å endre på initiell tøyning for modellen, ikke for å legge inn initiell temperatur for modellen. Det samme gjaldt for individuelle *members*. Forsøkte en bruker å legge til en initiell temperatur på en *member* som allerede hadde initiell tøyning kom dialogboksen som er vist i figur 22.

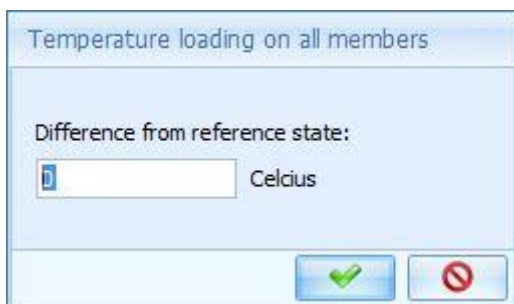


Figur 21 - Forsøk på å legge til initiell temperatur til en modell som har initiell tøyning i versjon 2.0

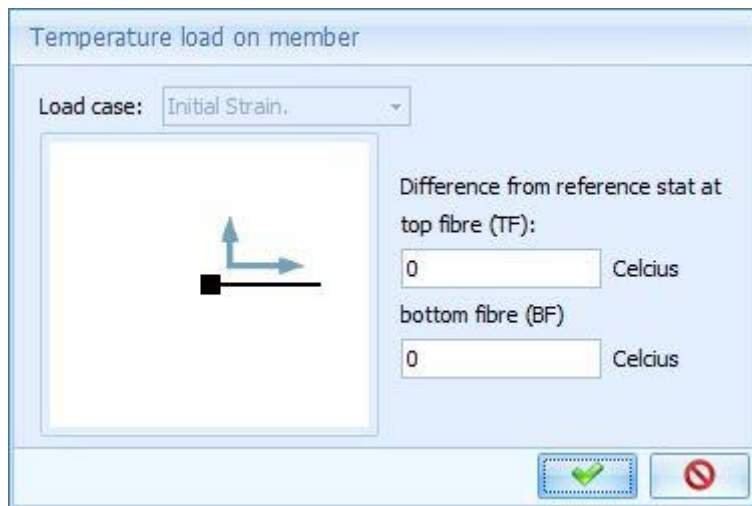


Figur 22 - Forsøk på å legge til en initiell temperatur på en *member* som allerede har initiell tøyning i versjon 2.0

I versjon 3.0 er dette endret på. Er det lagt til en initiell tøyning på hele modellen, og brukeren forsøker å legge inn initiell temperatur kommer dialogboksen vist i figur 23 opp. Det blir da lagt inn initiell temperatur for hele modellen og initiell tøyning blir fjernet. Figur 24 viser dialogboksen som kommer opp når en bruker forsøker å legge inn en initiell temperatur på en individuell *member* som allerede har initiell tøyning.



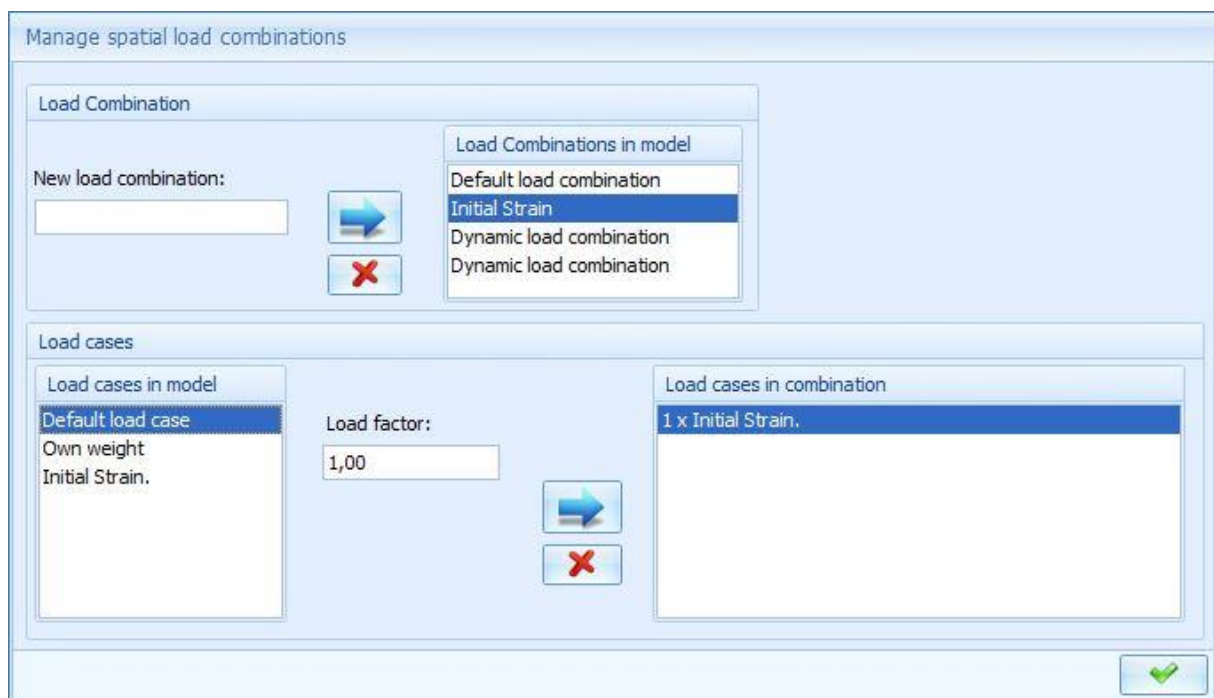
Figur 23 - Forsøk på å legge inn initiell temperatur til en modell som har initiell tøyning i versjon 3.0



Figur 24 - Forsøk på å legge inn initiell temperatur for en *member* som har initiell tøyning i versjon 3.0

4.3.5 Lastkombinasjoner

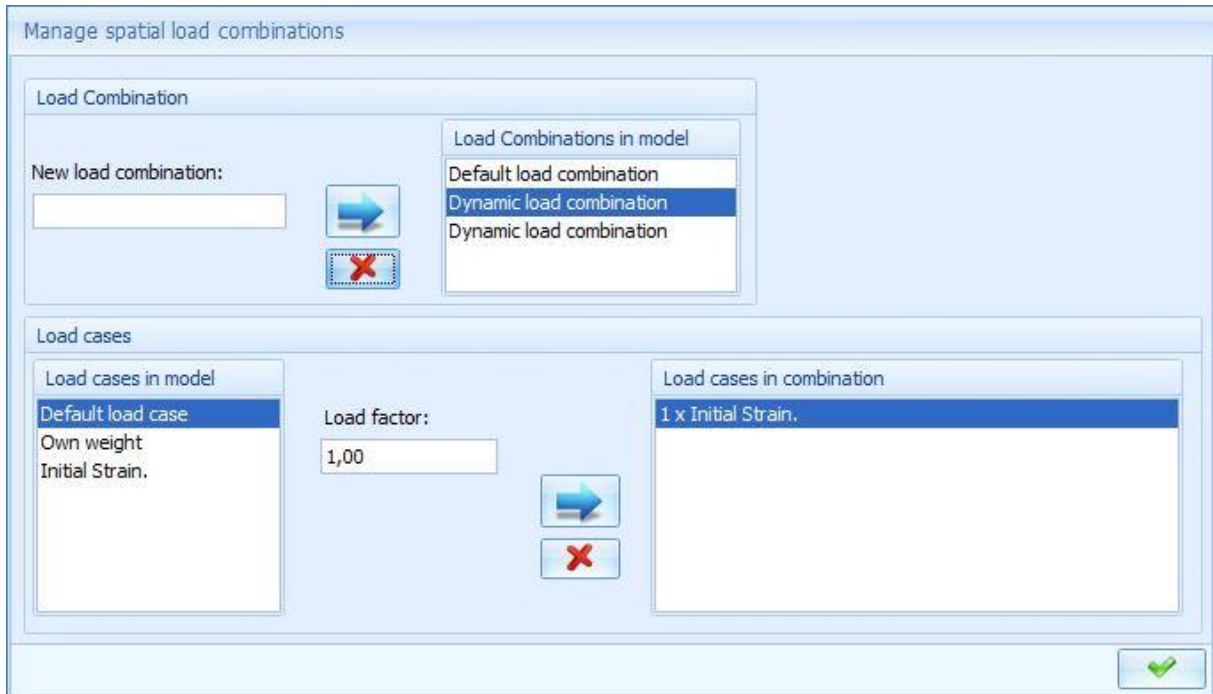
Denne dialogboksen skal vise de tilgjengelige lastkombinasjonene til modellen. I versjon 2.0, hvis en bruker la til en dynamisk lastkombinasjon og lagret modellen så la denne dynamiske lastkombinasjonen seg i denne dialogboksen, sammen med de vanlige lastkombinasjonene. Lagret en bruker modellen flere ganger la den dynamiske lastkombinasjonen seg sammen med de vanlige lastkombinasjonene like mange ganger som lagring ble kalt. Figur 25 viser en modell med to vanlige lastkombinasjoner og en dynamisk lastkombinasjon hvor modellen er blitt lagret to ganger.



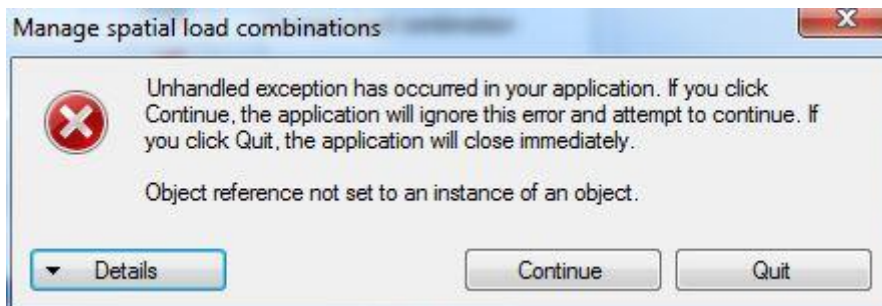
Figur 25 - Dialogboksen til lastkombinasjoner som er lagret to ganger i versjon 2.0

Figur 25 viser de vanlige lastkombinasjonene sammen med den ene dynamiske lastkombinasjonen som da vises to ganger siden modellen er blitt lagret to ganger. Dette er rettet opp i versjon 3.0 og denne dialogboksen vil nå aldri vise de dynamiske lastkombinasjonene.

I denne dialogboksen var det og mulig å slette lastkombinasjonen, men lastene som lå i denne lastkombinasjonen ble ikke slettet samtidig. Figur 26 viser samme dialogboks som figur 25, men her er lastkombinasjonen **Initial Strain** slettet. Men lasten i lastkombinasjonen er fortsatt synlig, og figur 27 viser hva som skjer om en bruker da forsøkte å slette lasten.

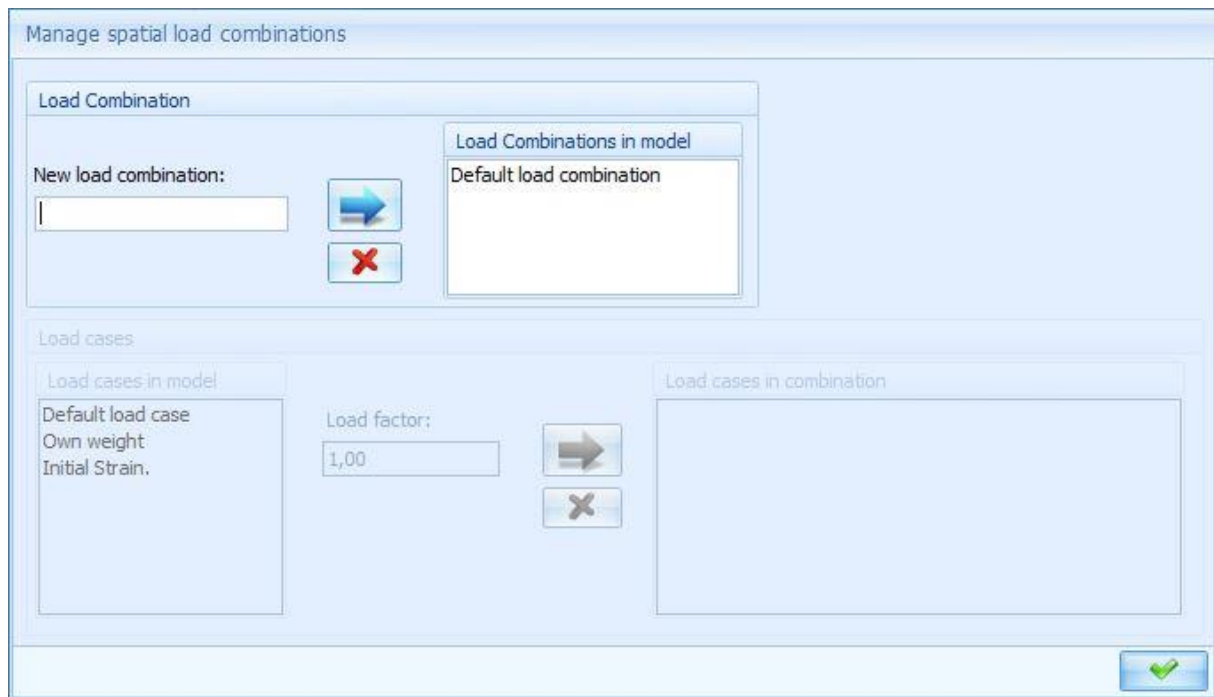


Figur 26 - Dialogboksen til lastkombinasjoner etter en lastkombinasjon er slettet i versjon 2.0



Figur 27 - Forsøk på å slette en last fra en slettet lastkombinasjon i versjon 2.0

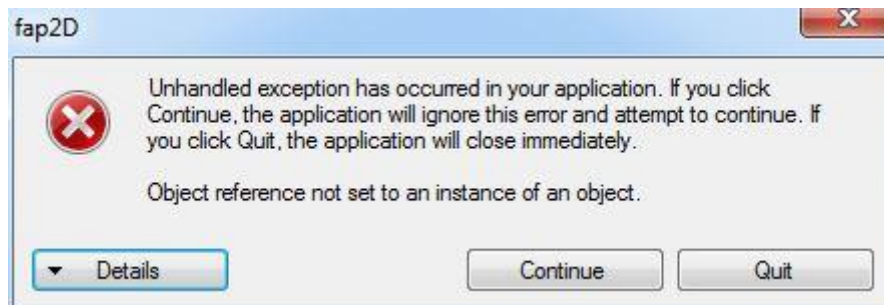
Figur 28 viser dialogboksen til lastkombinasjoner i versjon 3.0. Her er ingen dynamiske laster lagt til. Og lastkombinasjonen **Initial Strain** er nettopp blitt slettet. Da blir automatisk lastene som tilhører denne lastkombinasjonen fjernet fra dialogboksen, og enhver mulighet for å få et stopp i programmet her er fikset.



Figur 28 - Dialogboksen til lastkombinasjoner etter en lastkombinasjon er slettet i versjon 3.0

4.3.6 View loading

Når det ikke er noen modeller åpne i programmet er det enda mulig å klikke på knappene under **View loading** i resultatfanen. I versjon 2.0 så ble en exception produsert når en bruker trykte på en av disse knappene og det ikke var noen modeller åpne, se figur 29.

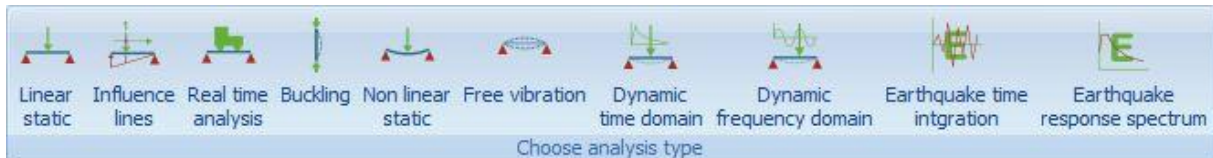


Figur 29 - Trykke på knapp under View Loading uten modeller åpne i versjon 2.0

I versjon 3.0 er dette tatt hånd om, og selv om det fortsatt er mulig å faktisk trykke på en av knappene så skjer det ingenting når en bruker trykker en av de inn.

4.4 Analysis

Den nye versjonen har fått tre nye analyser. En som gjør beregninger på lasttog og to som gjør beregninger på jordskjelv. Den nye *ribbon page group* er vist i figur 30, mens den gamle er vist i figur 31.



Figur 30 - Ny *ribbon page group*



Figur 31 - Gammel *ribbon page group*

Det er og blitt implementert slik at alle knapper i programmet blir utkoplet under en analyse, og det er lagt inn en egen animert muspeker som vises under analyse. Denne animasjonen er hentet fra nettstedet Ajaxload[12].

4.4.1 Linear static

Denne er ikke forandret siden siste versjon. Den inneholder den nye dialogboksen for lastkombinasjoner som er forklart i kapittel 4.3.5 i denne teksten.

4.4.2 Influence lines

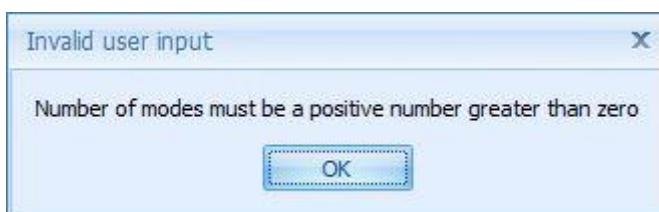
Denne har fått nye bilder til knappene **Define load path** og **Define load train**, ellers er den uforandret siden sist versjon.

4.4.3 Real time analysis

Her ligger funksjonaliteten til den nye analysen lasttog. Denne har Frans Erstad implementert.

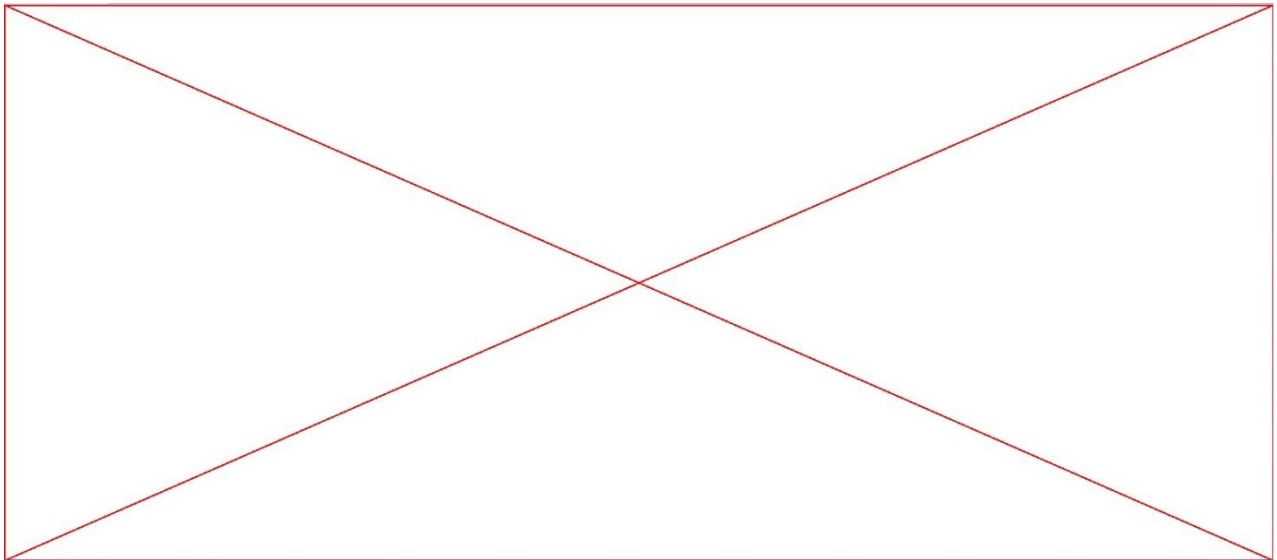
4.4.4 Buckling

Denne analysen har fått en test under **Number of modes** som tilsier at det ikke er lovlig å skrive inn negative tall eller nullverdi. Da får brukeren beskjed om dette i form av en boks som er vist i figur 32.



Figur 32 - Beskjed om number of modes i buckling analyse

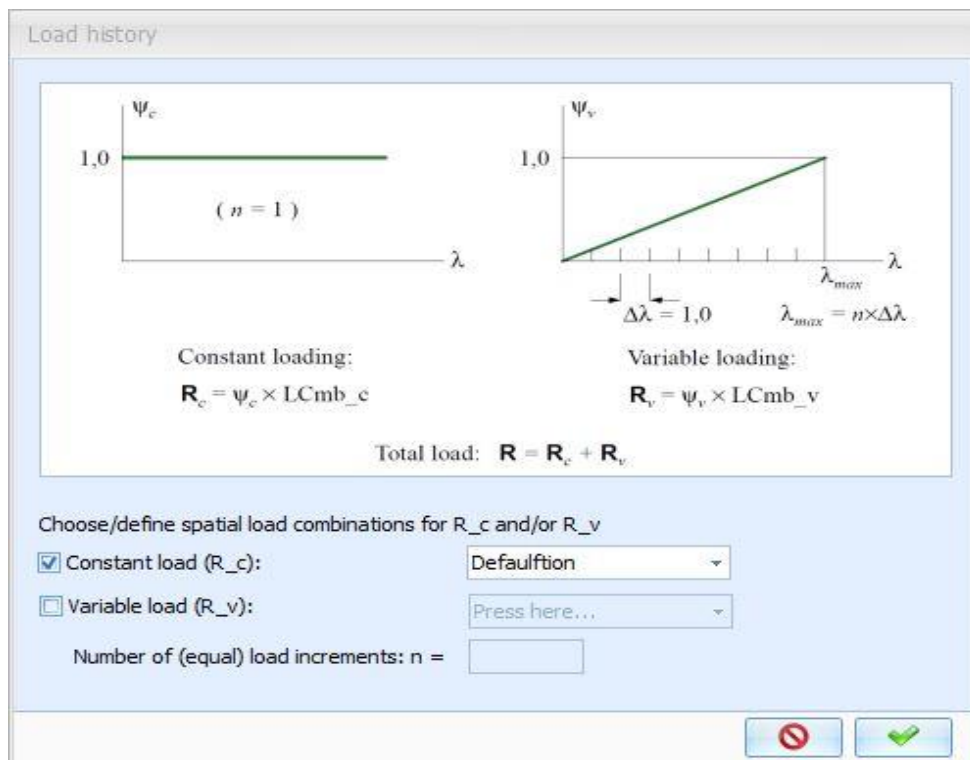
Det er og lagt inn test som gjør slik at hvis en analyse feiler og det er resultater tilgjengelig så vil **Results**-fanen bli utkoblet. Ellers er det risiko for at modellvinduet blir seende som i figur 32 under resultatene.



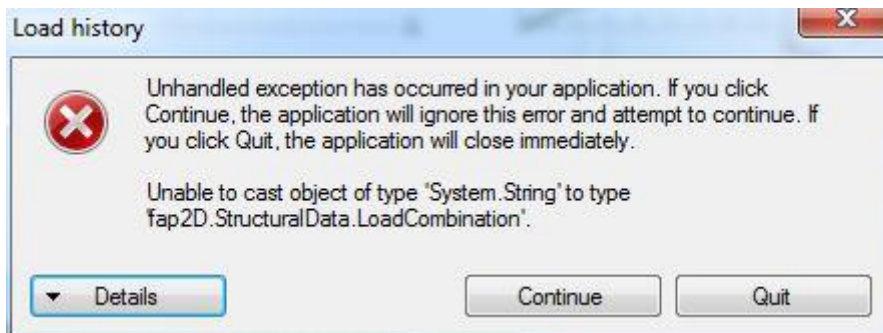
Figur 33 - Buckling analyse feilet og Results-fanen forsøkt åpnet

4.4.5 Non linear static

Hvis en bruker åpner opp dialogboksen til **Load history** og skriver inn en lastkombinasjon som ikke er definert , slik som i figur 34, fikk brukeren en *exception* i versjon 2.0 som er vist i figur 35.



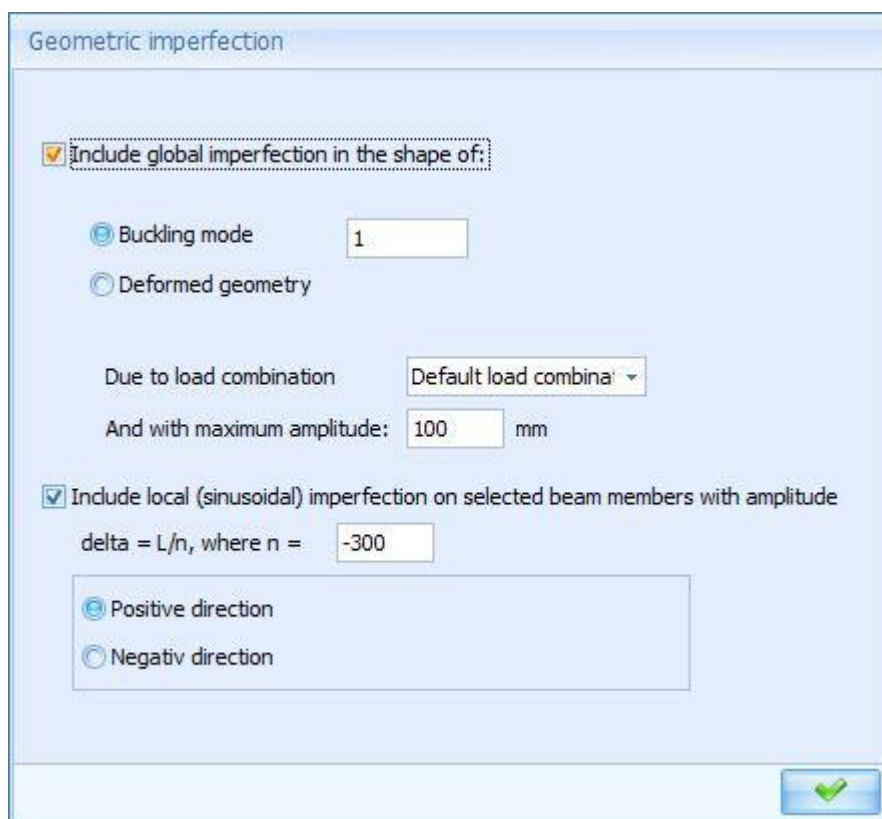
Figur 34 - Load history med lastkombinasjon som ikke eksisterer



Figur 35 - Exception ved lastkombinasjon som ikke eksisterer

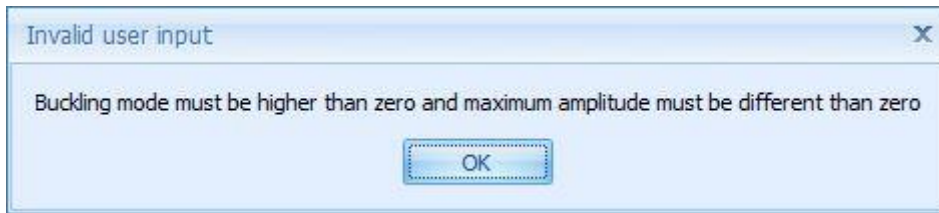
I den nye versjonen er dette rettet opp, og skriver en bruker inn en lastkombinasjon som ikke eksisterer skjer det ingenting.

Dialogboksen til **Geometric imperfection**, se figur 36, har fått implementert noen tester.

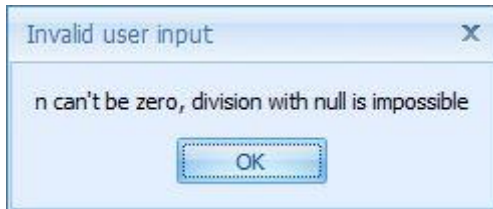


Figur 36 - Dialogboksen til *geometric imperfection*

Her var det i versjon 2.0 lov å skrive inn en lastkombinasjon som ikke eksisterer og programmet produserte en *exception* lik den i figur 35. Den blir ikke lengre produsert i versjon 3.0. Det er heller ikke lov å skrive inn en **Buckling mode** lik null eller negative verdier. Og det er ikke tillatt for en bruker å skrive en maksimum amplitude lik null heller. Da produserer dialogboksen en beskjed om dette som er vist i figur 37. Det er heller ikke tillatt for brukeren å skrive inn **n** lik null heller, da produserer dialogboksen en beskjed som er vist i figur 38.



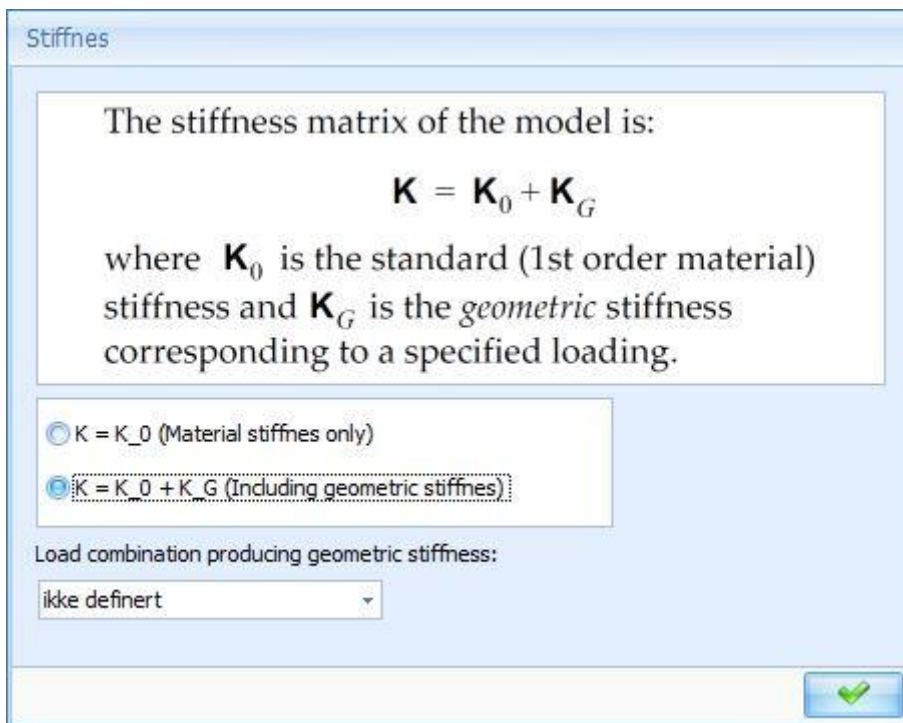
Figur 37 - Forsøk på å skrive inn en *buckling mode* lik null eller negativ verdi, eller en maksimum amplitude lik null



Figur 38 - Forsøk på å skrive inn *n* lik null

4.4.6 Free vibration

Det er lagt inn en test i dialogboksen til stivhetsmatrisen, som gjør det slik at det ikke lengre kommer en *exception* når en bruker forsøker å legge inn en lastkombinasjon som ikke er definert, se figur 39. I versjon 2.0 ble en *exception* produsert, som er vist i figur 35.

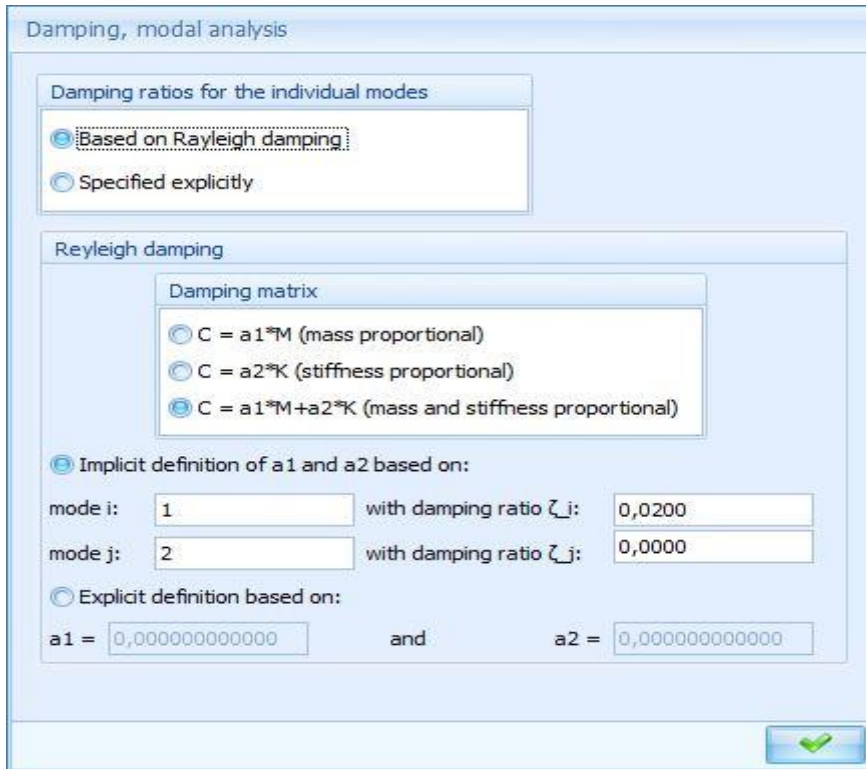


Figur 39 - Forsøk på å legge inn lastkombinasjon som ikke eksisterer

Det er og lagt inn en test som gjør at brukeren ikke er tillatt i å skrive inn **Number of modes** lik null eller negative verdier. Forsøker en bruker å gjøre dette for så å kjøre en analyse blir beskjednen som er vist i figur 32 produsert.

4.4.7 Dynamic time domain

Dialogboksene til stivhetsmatrise og **Computational model** har fått tester som gjør det slik at det ikke er tillatt for en bruker å skrive inn egen lastkombinasjon som ikke er definert. I versjon 2.0 produserte programmet en *exception* som er vist i figur 35. Dialogboksen til dempning, vist i figur 40, har og fått noen tester.



Figur 40 - Dempning for tvungen svinging i tidsplanet

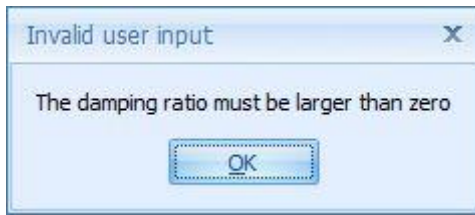
Forsøker en bruker å legge inn **mode i** eller **mode j** lik null eller negativer verdier blir det produsert beskjedene vist i figur 41 og 42. Det er heller ikke tillatt for en bruker å legge inn verdier til **damping ratio** lik null eller negative verdier. Da blir beskjeden vist i figur 43 produsert.



Figur 41 - Forsøk på å legge inn *mode i* lik null eller negative verdier

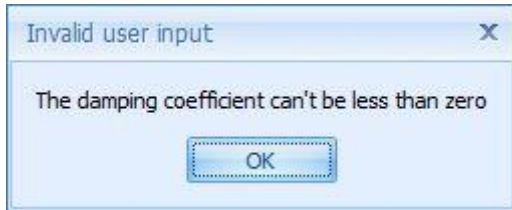


Figur 42 - Forsøk på å legge inn *mode j* lik null eller negative verdier



Figur 43 - Forsøk på å legge inn *damping ratio* lik null eller negative verdier

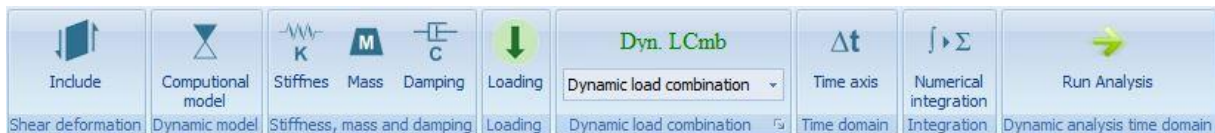
Velger brukeren en eksplisitt definisjon av dempingen er det lagt inn test som produserer beskjeden vist i figur 44 hvis en bruker forsøker å legge inn **a1** eller **a2** lik negative verdier.



Figur 44 - Forsøk på å legge inn **a1** eller **a2** med negative verdier

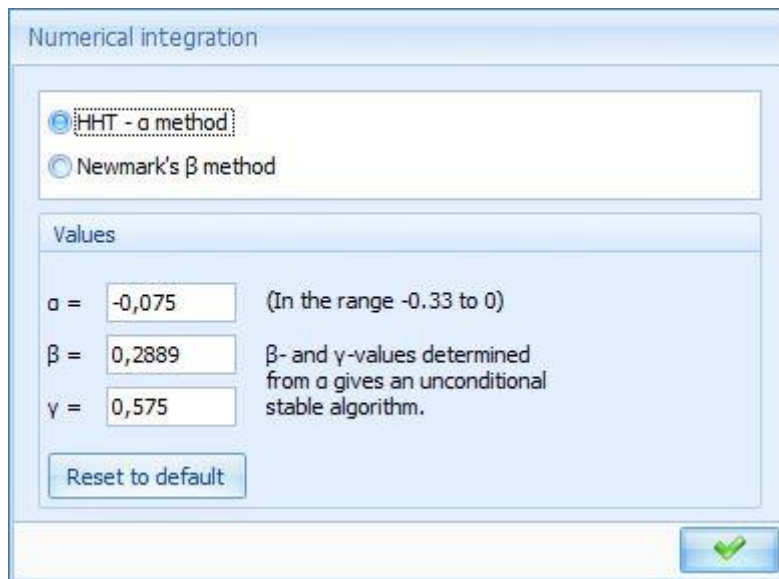
Dialogboksen for de dynamiske lastkombinasjonene slet med samme problem som dialogboksen som de statiske lastkombinasjonene som er forklart i kapittel 4.3.5. Sletter en bruker en lastkombinasjon ble ikke de tilhørende lastene slettet automatisk fra dialogboksen. For nærmere beskrivelse henvises det til kapittel 4.3.5 i denne teksten.

Det har og blitt tilført en liste som viser alle de tilgjengelige dynamiske lastkombinasjonene. Det gjør det enklere å velge en dynamisk lastkombinasjon som en bruker vil regne på. Figur 45 viser hvordan knappene under denne analysen nå er organisert.

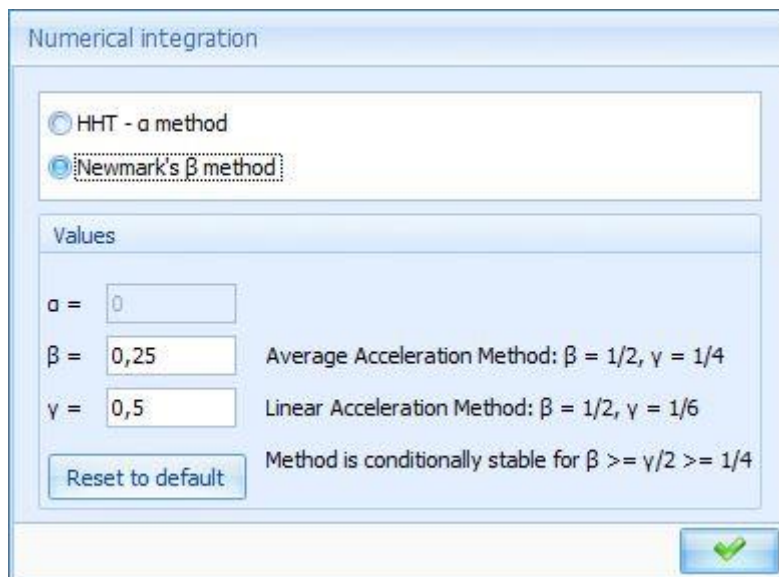


Figur 45 - Ny *layout* for dynamic time domain

Dialogboksen til **Numerical integration** har blitt pusset opp. Figurene 46 og 47 viser hvordan det nå ser ut med de to forskjellige metodene valgt. Det er en ny knapp som en bruker kan trykke på som setter verdiene tilbake til en standard verdi som vises i figurene. Det er og lagt inn en annen tekst for Newmark sin integrasjonsmetode.



Figur 46 - Numerical integration med HHT metode valgt

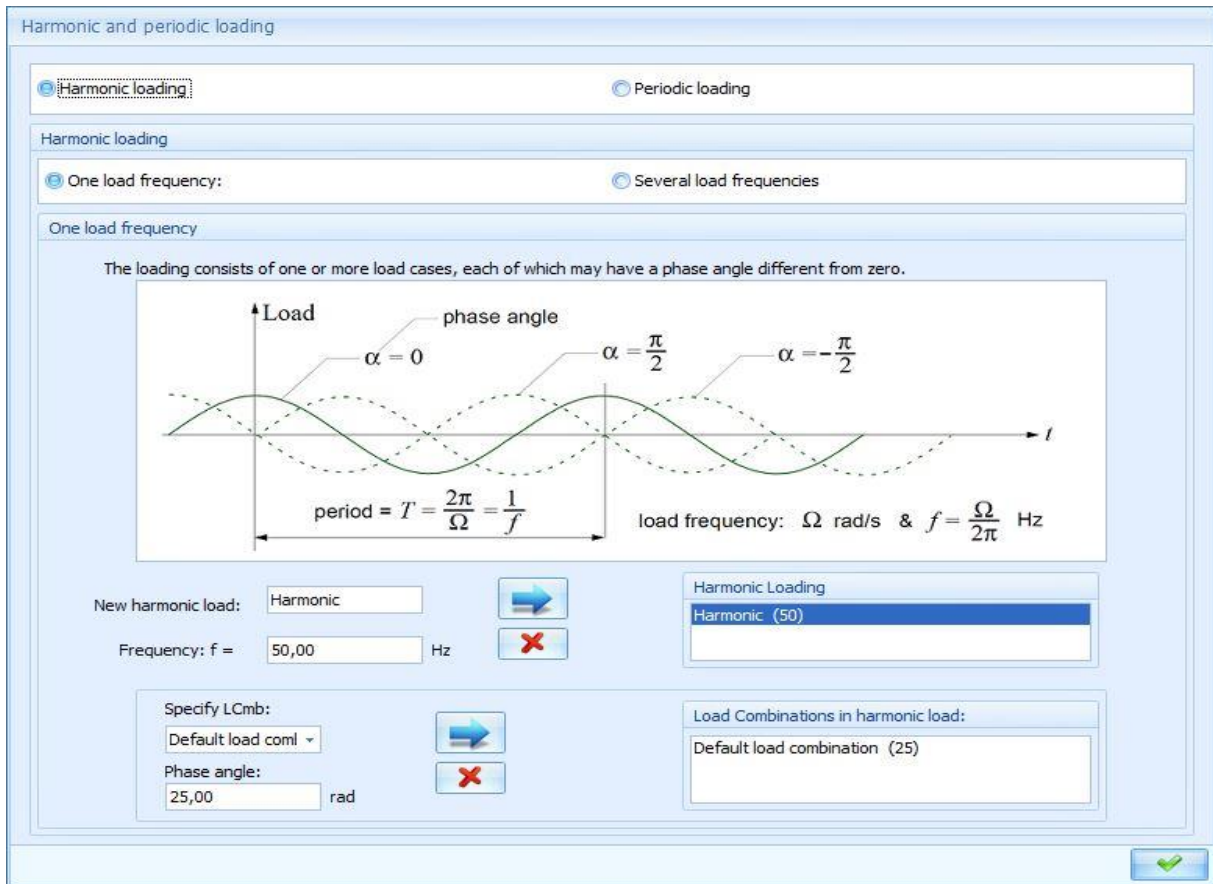


Figur 47 - Numerical integration med Newmark metode valgt

4.4.8 Dynamic frequency domain

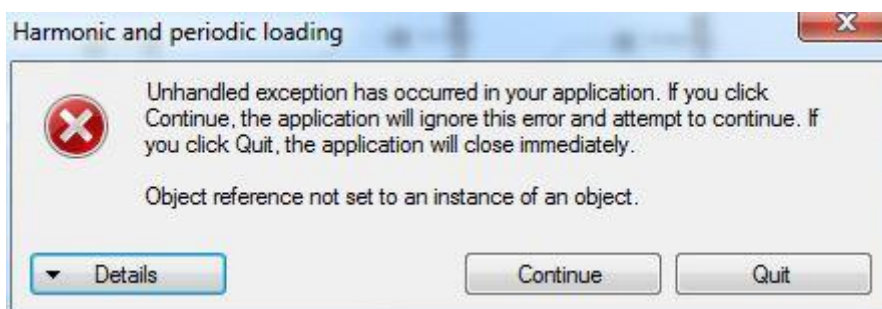
Denne analysen benytter de same dialogboksene som *dynamic time domain* hva gjelder **Computational model**, stivhetsmatrisen og dempning. På versjon 2.0 var det ikke tillatt å kjøre en analyse med harmonisk last hvor det var flere lastkombinasjoner med i den harmoniske lasten. Dette er rettet opp i versjon 3.0.

Dialogboksen for last er vist i figur 48 og det er lagt inn noen tester som gjør slik at det ikke blir produsert *exceptions* lik den gamle versjonen gjorde.



Figur 48 - Dialogboksen for last tilhørende dynamic frequency domain

Forsøkte en bruker å slette en harmonisk last når det ikke var noen harmonisk last å slette ble *exception* vist i figur 49 produsert. Den samme *exception* ble og produsert hvis en bruker forsøkte å legge til en lastkombinasjon som ikke finnes til en harmonisk last. Og den ble og produsert hvis en bruker slettet alle harmoniske lastene og så forsøkte å slette lastkombinasjon som ikke fantes fra listen **Load combinations in harmonic load**.



Figur 49 - *Exception* i harmonisk last

4.4.9 Earthquake time integration

Denne analysen har Erik Aasmundsrud implementert.

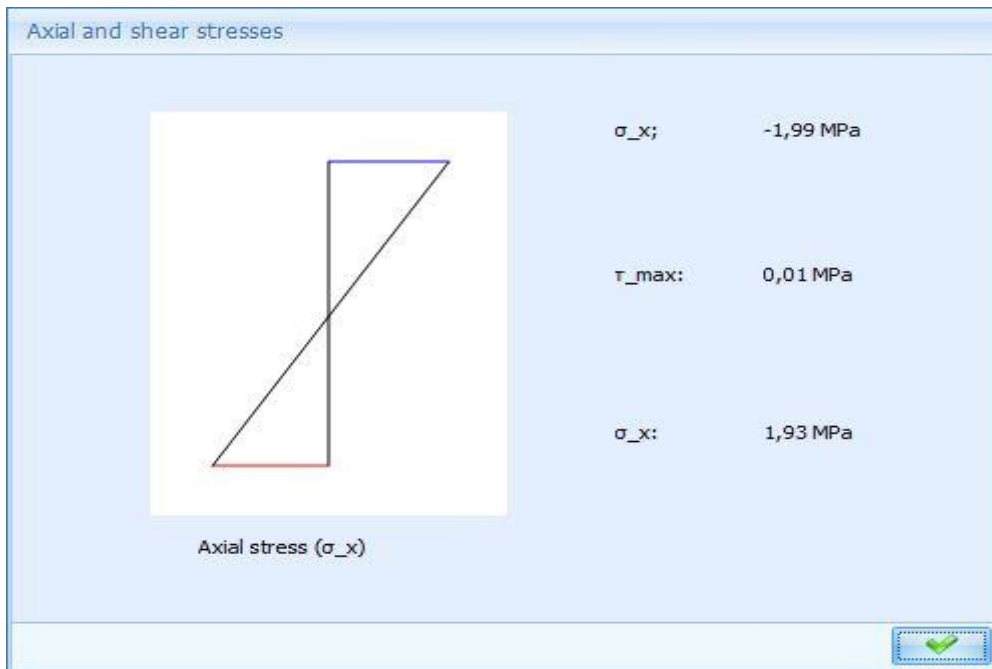
4.4.10 Earthquake response spectrum

Denne analysen har Erik Aasmundsrud implementert.

4.5 Results

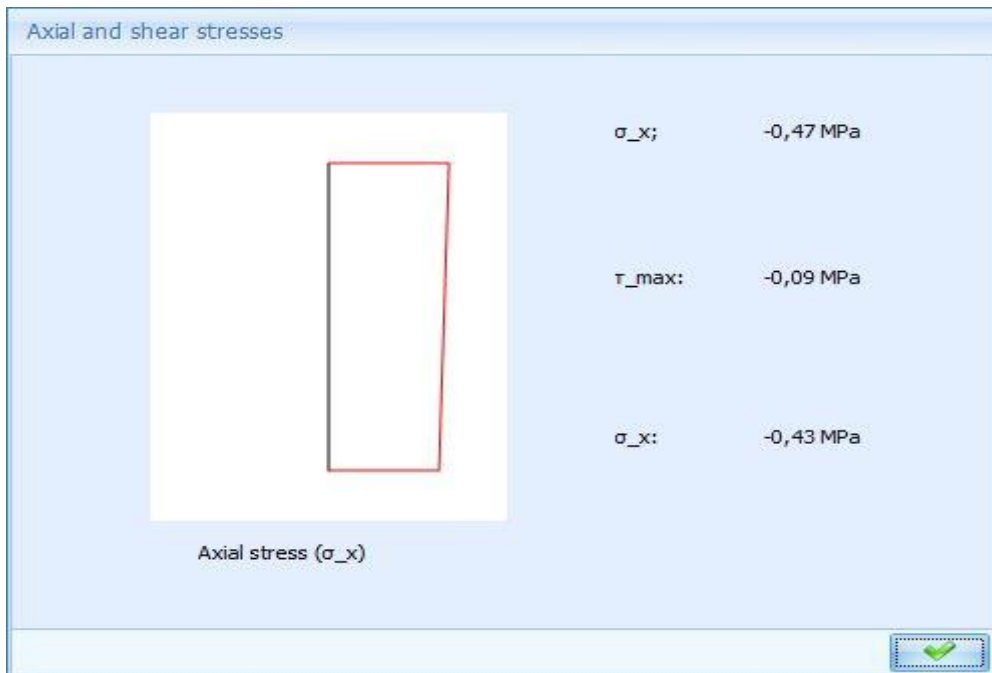
4.5.1 Axial and shear stresses

I denne dialogboksen vises det et diagram over spenningsfordelingen av et element, se figur 50.



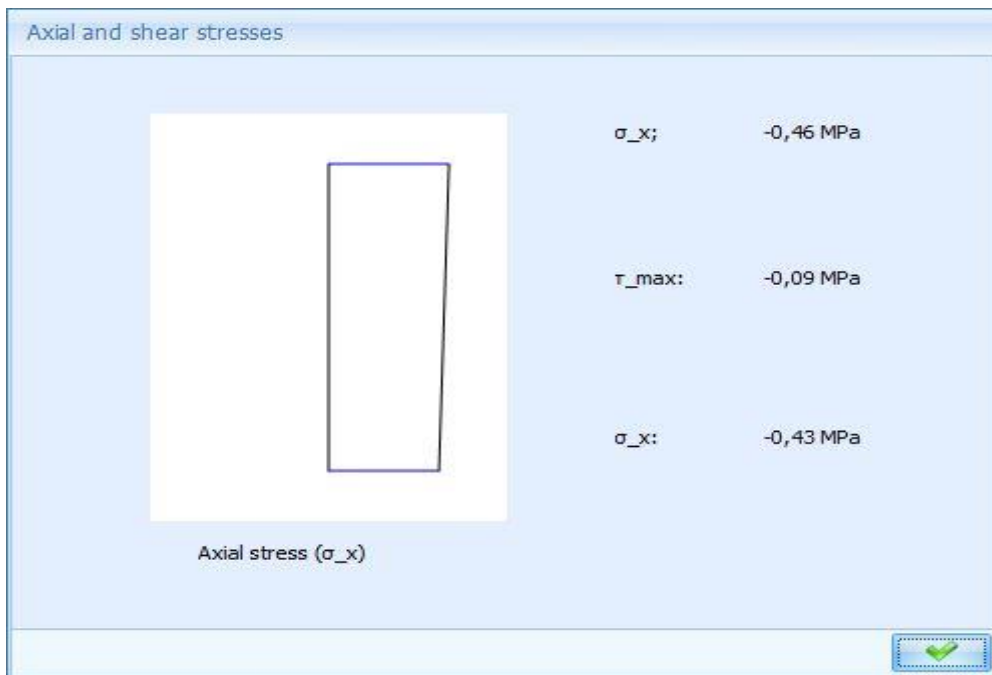
Figur 50 - Spenningsfordeling for et element

Her tegnes strekk som rødt og trykk som blått. I versjon 2.0 ble tegnet dette greit, men når et element hadde kun trykk ble alt tegnet rødt, se figur 51.



Figur 51 – Versjon 2.0 med kun trykk

I versjon 3.0 er dette endret, og når et element har kun trykk blir det tegnet opp slik som i figur 52.

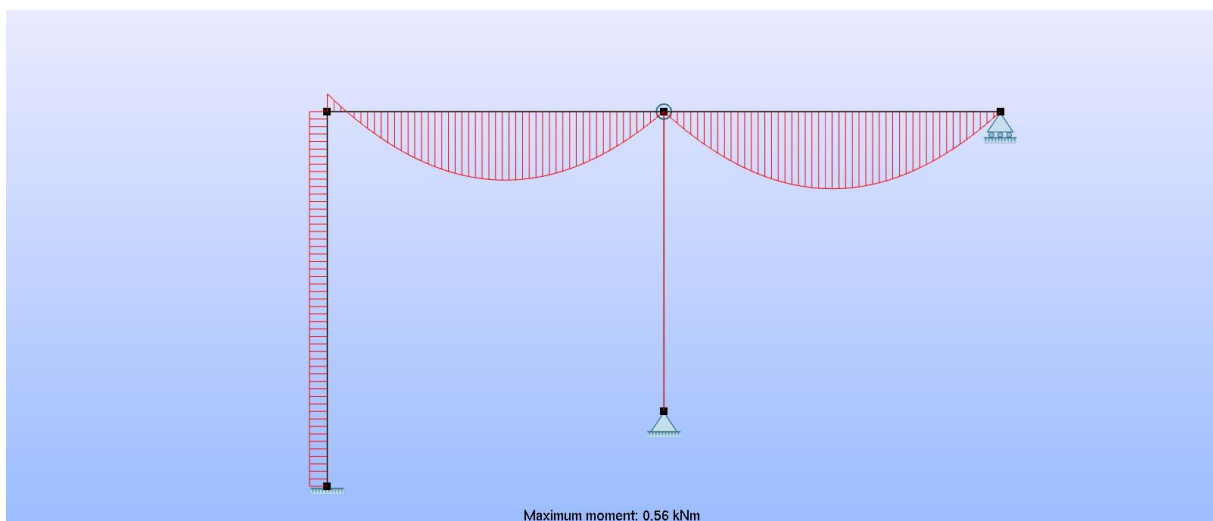


Figur 52 – Versjon 3.0 med kun trykk

Dette gjaldt og når det var kun strekk, da ble alt tegnet blått. Det er og fikset i versjon 3.0.

4.5.2 Section forces

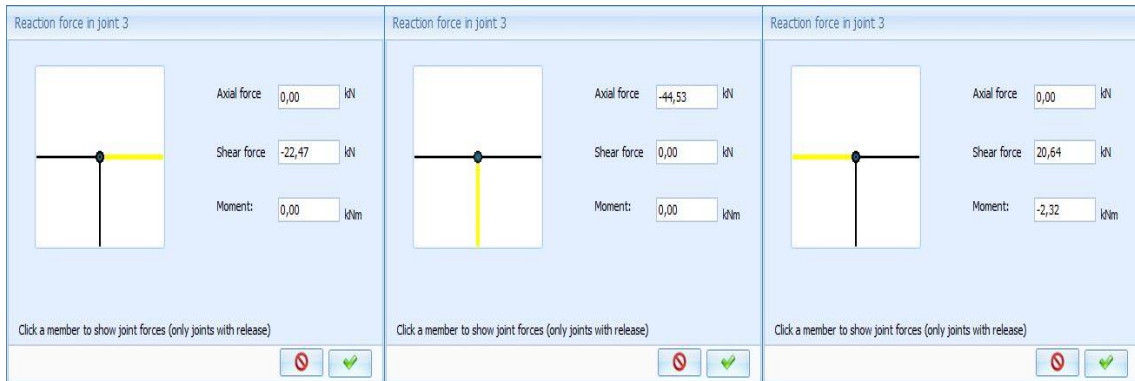
Figur 53 viser momentdiagrammet til en ramme med et momentfridd ledd i midtpunktet på øverste *member*. Det er tegnet korrekt med null moment i dette leddet. En bruker har mulighet til å høyreklikke på dette leddet og se på hvor store krefter og moment rundt leddet er.



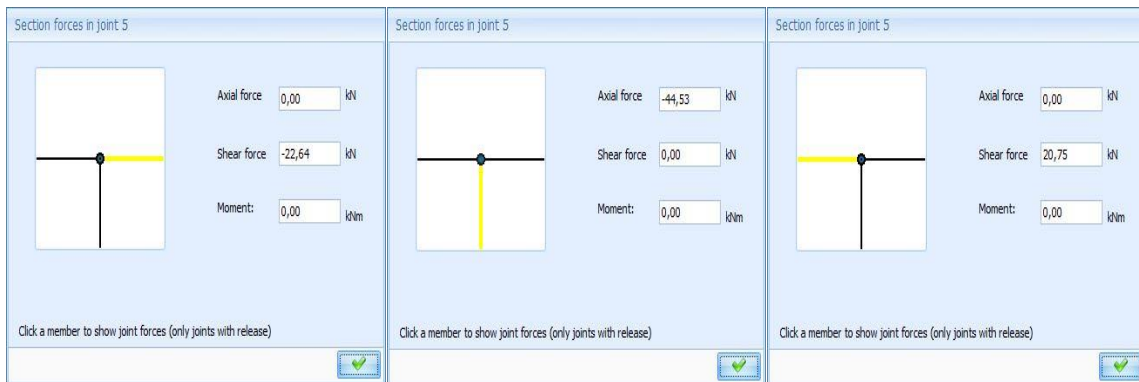
Figur 53 - Momentdiagram for ramme med momentfridd ledd

Når en bruker høyreklikker på dette leddet blir dialogboksen som er vist i figur 54 produsert. Her kan brukeren klikke med muspekeren på de forskjellige *members* for å se hvor store kreftene og momentene rundt leddet er. Det skal vise at det er null moment rundt leddet. Men i versjon 2.0 ble

det vist at leddet faktisk inneholder moment, noe det ikke gjør. Det samme leddet er vist i figur 55, hvor problemet er løst.

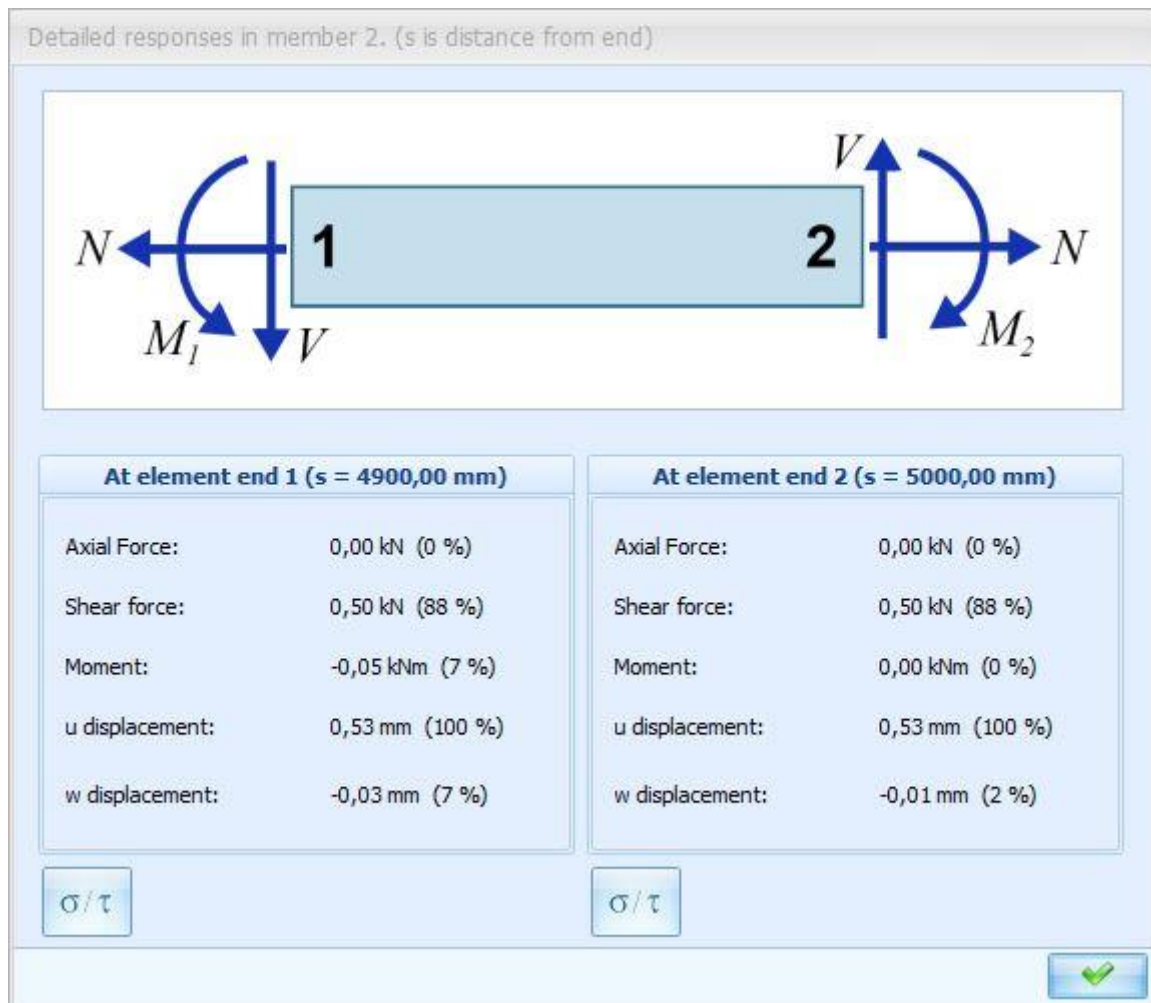


Figur 54 - Moment og krefter rundt et momentfritt ledd i versjon 2.0



Figur 55 - Moment og krefter rundt et momentfritt ledd i versjon 3.0

Versjon 2.0 viste at dette leddet hadde moment pga at elementet som ligger nærmest til venstre for leddet har forskjellig moment på begge endene sine. Figur 56 viser detaljert respons for dette elementet. Og der har elementet null moment i høyre ende, men har litt moment i venstre ende. I versjon 2.0 ble momentet i venstre ende hentet når det momentfrie leddet ble undersøkt, og dermed ble et momentfritt ledd vist som at det inneholdt moment.



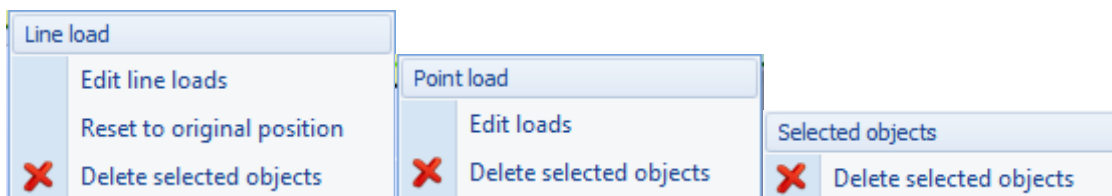
Figur 56 - Detaljert respons til element rett til venstre for momentfridd ledd

4.6 Popup menu

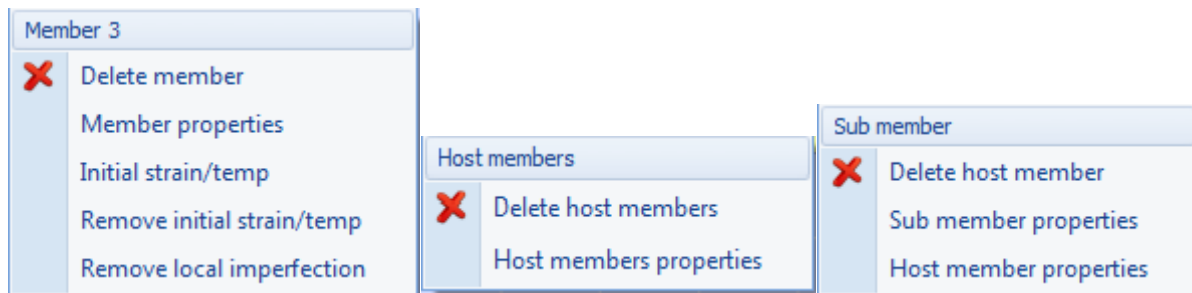
Når en bruker høyreklikker med muspekeren på en *member*, *joint* eller en last kommer det en liste med valg. Figur 57 viser hvordan høyreklikk på et *joint* eller et *internal joint* kan se ut, og hvordan høyreklikk på flere merkede *joints* ser ut. Figur 58 viser hvordan høyreklikk på en linjelast ser ut, hvordan høyreklikk på en punktlast ser ut og hvordan høyreklikk på flere ulike merkede objekter ser ut. Figur 59 viser hvordan høyreklikk på en *member* kan se ut, hvordan høyreklikk på flere *members* ser ut og hvordan høyreklikk på en *sub-member* ser ut.



Figur 57 - Popup-menyer for et *joint* og flere *joints*



Figur 58 - Popup-menyer for linjelast, punktlast og for flere merkede objekter



Figur 59 - Pop-up-menyner for en *member*, flere *members* og for *sub-member*

4.6.1 Joint

4.6.1.1 Prescribed displacement

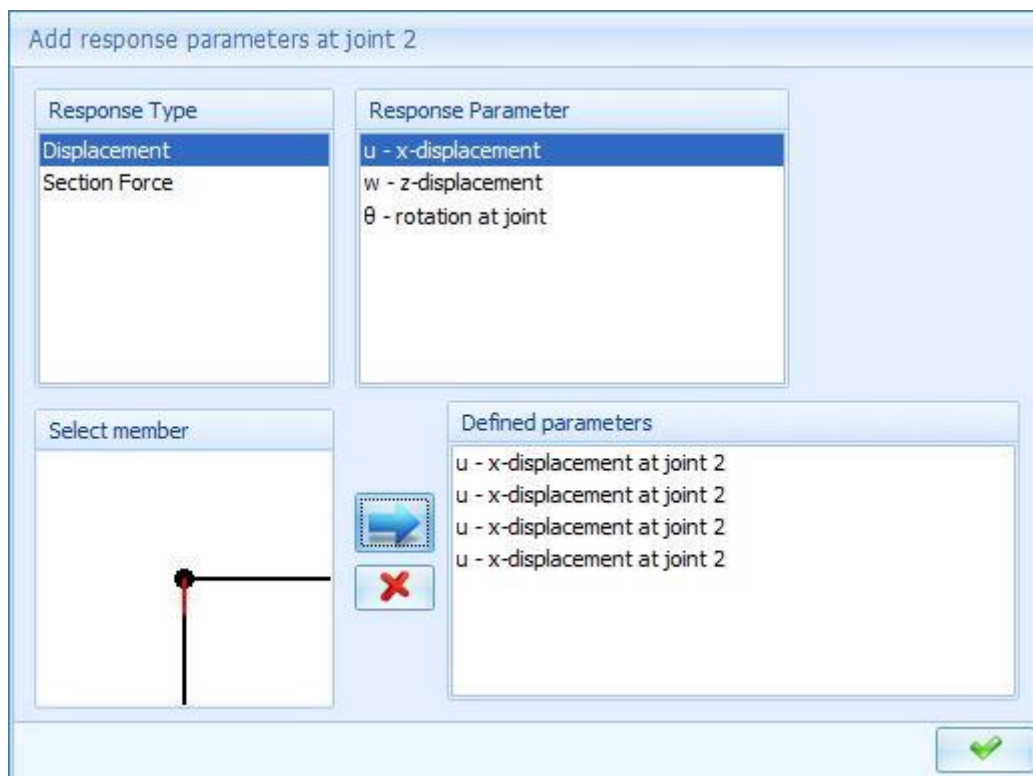
Figur 60 viser en foreskrevet forskyvning. Versjon 2.0 var det mulig å redigere denne til null, og dermed ble den foreskrevne forskyvningen borte. På den nye versjonen er ikke det mulig, siden det allerede er mulig å fjerne en foreskrevet forskyvning ved å klikke med muspekeren på **Remove prescribed displacement**.

Figur 60 - *Prescribed displacement* på et *joint*

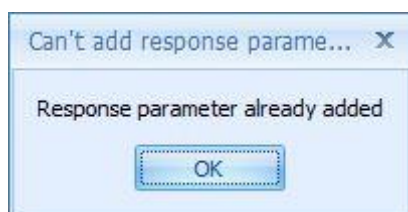
Forsøker en bruker å sette denne til null på den nye versjonen blir det produsert en beskjed lik den i figur 20.

4.6.1.2 Response parameter

I versjon 2.0 var det muligheter for en bruker å legge inn flere responsparametre av den samme typen, det kommer tydelig fram i figur 61. Forsøker en bruker å gjøre det samme i versjon 3.0 blir det produsert en beskjed, se figur 62, og brukeren får ikke lagt til flere enn en type av hver responsparameter.



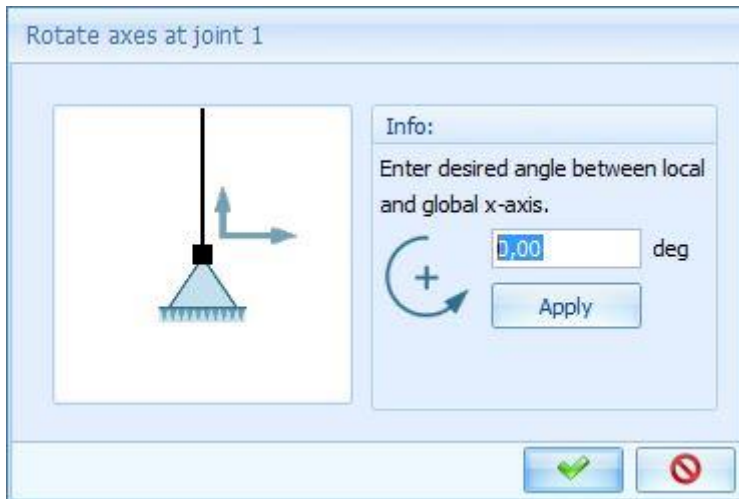
Figur 61 - *Response parameters* i versjon 2.0



Figur 62 - Forsøk på å legge inn eksisterende *response parameter* i versjon 3.0

4.6.1.3 Local axes

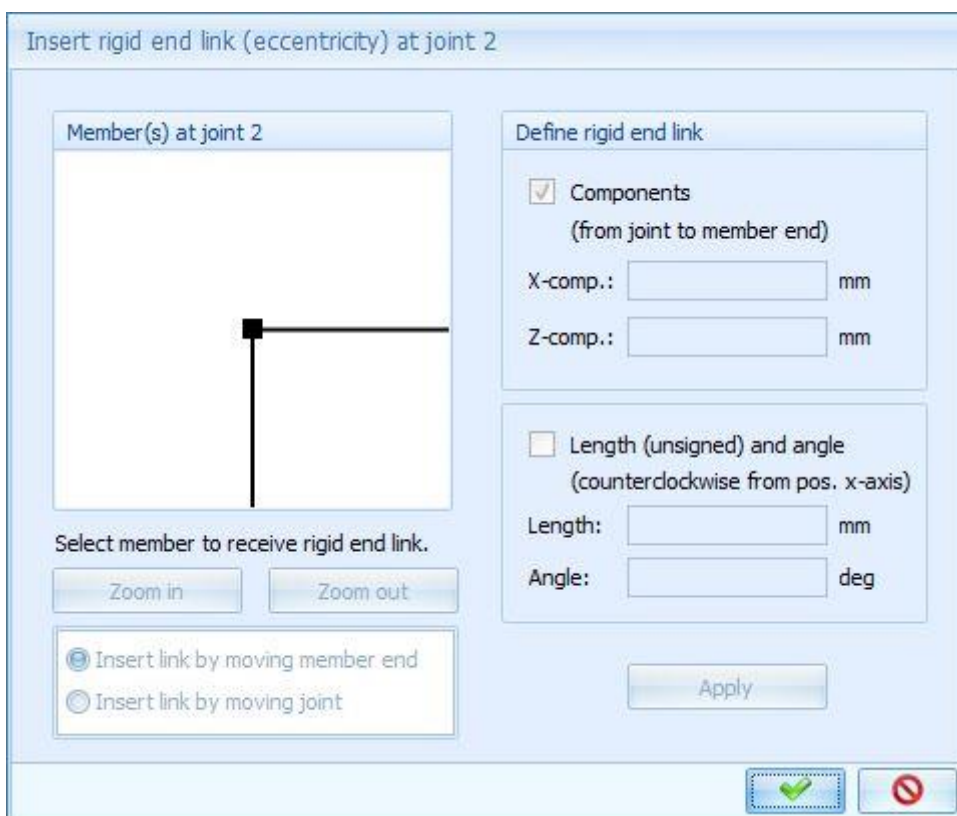
Det er mulig å legge til, redigere og legge til lokale akser på *joints*, se figur 63 for dialogboksen til dette. Forsøker en bruker å legge til en lokal akse, men angret seg vil det ikke bli lagt til noe som helst hvis brukeren går ut av boksen med **Cancel**-knappen. I versjon 2.0 ble det lagt til lokal akse hvis en bruker angret seg for så å bruke Esc-knappen på tastaturet for å lukke dialogboksen. Dette er rettet opp i versjon 3.0, og Esc-knappen sin funksjonalitet fungerer på lik måte som **Cancel**-knappen i dialogboksen.



Figur 63 - Lokal akse ved et *joint*

4.6.1.4 Eccentricity

Figur 64 viser dialogboksen for eksentrisiteter. I versjon 3.0 er zoom-knappene utkopleet frem til det foreligger en eksentrisitet å zoome ut eller inn på.

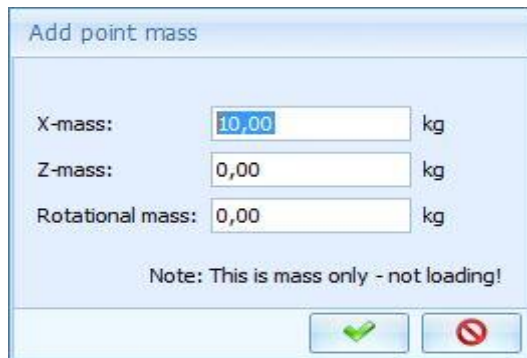


Figur 64 - Dialogboks for eksentrisiteter

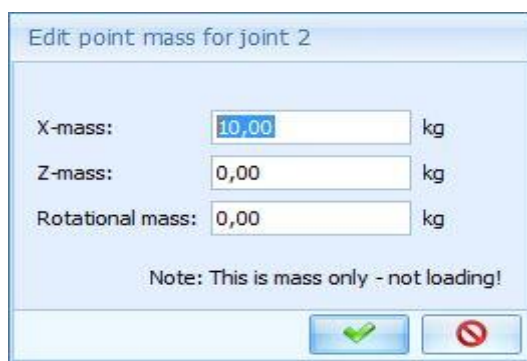
4.6.1.5 Point mass

I versjon 2.0 var det ikke mulighet for en bruker å redigere punktmasse. Hvis en bruker klikket på **Edit point mass** kom dialogboksen for en ny punktmasse opp, med verdiene lik den punktmassen brukeren klikket på. Trykket så brukeren på **Ok**-knappen fikk brukeren mulighet for å legge til en ny

punktmasse og den som brukeren forsøkte å redigere ble uforandret. Figur 65 viser dialogboksen som blir produsert når en bruker forsøker å redigere en punktmasse i versjon 2.0, og den viser at programmet gir brukeren mulighet til å legge til ny punktmasse med verdi lik den brukeren klikket på. Figur 66 viser dialogboksen som blir produsert når brukeren forsøker å redigere en punktmasse i versjon 3.0, og den viser at brukeren kan redigere punktmassen. Det er de samme testene her som er nærmere beskrevet i kapittel 4.2.3.



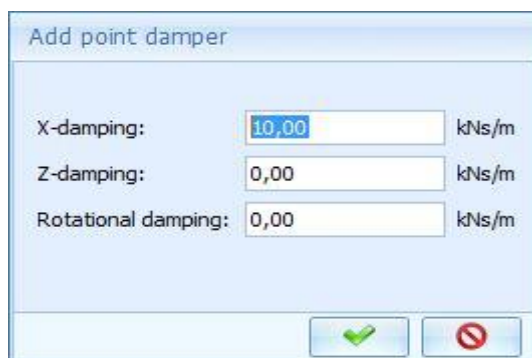
Figur 65 - Forsøk på å redigere punktmasse i versjon 2.0



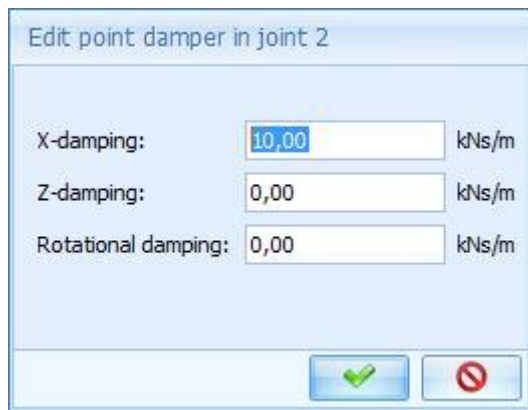
Figur 66 - Forsøk på å redigere punktmasse i versjon 3.0

4.6.1.6 Point damper

Redigering av punktdempere fungerte lik som punktmasse i versjon 2.0. Det var ikke mulig å redigere denne, kun legge til nye med verdier fylt inn lik den punktmassen brukeren forsøkte å redigere. Figur 67 og 68 viser dialogboksene i versjon 2.0 og i versjon 3.0 for redigering av punktdemper. De samme testene er lagt inn her som beskrevet i kapittel 4.2.3.



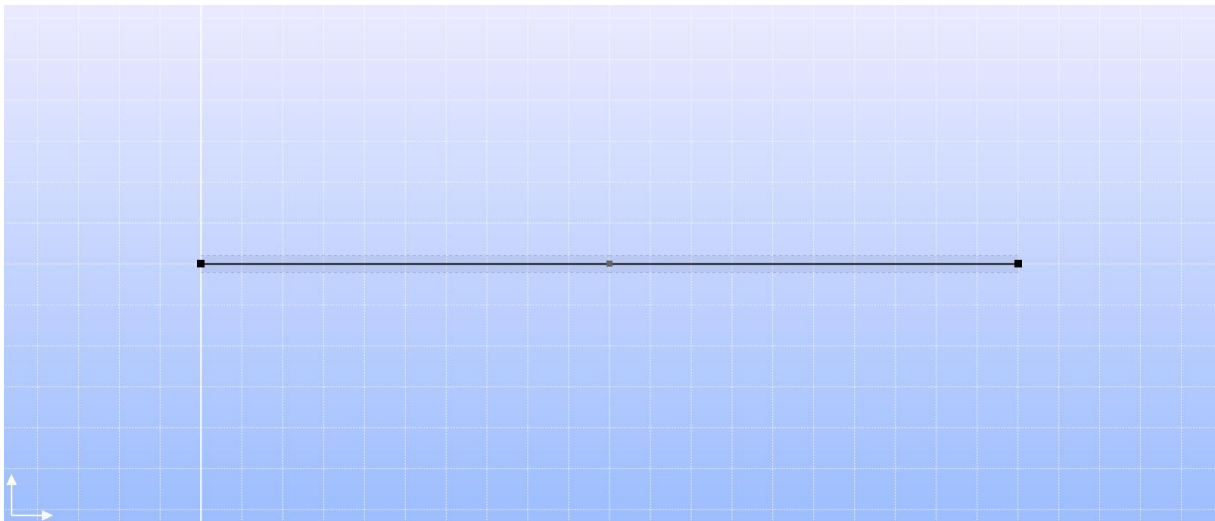
Figur 67 - Forsøk på å redigere punktdemper i versjon 2.0



Figur 68 - Forsøk på å redigere punktdemper i versjon 3.0

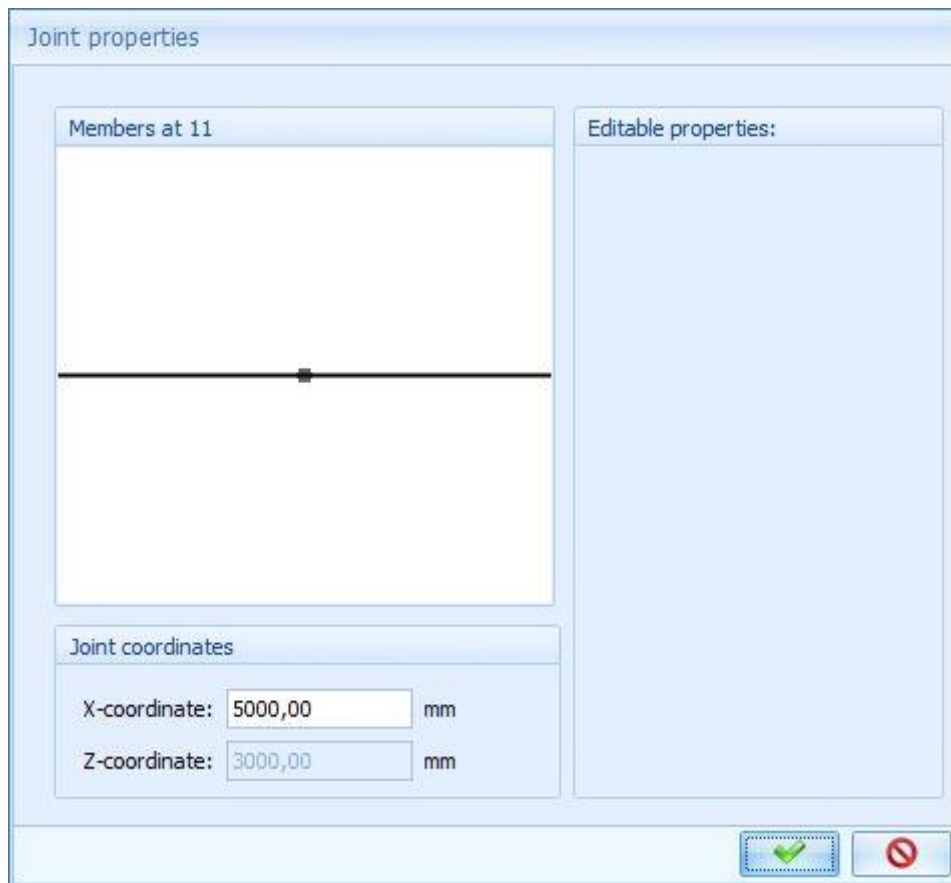
4.6.1.7 Properties

Figur 69 viser en bjelke med to *joints* og et *internal joint*. Lengden på bjelken er 10000 millimeter, og *internal joint* ligger midt på bjelken.



Figur 69 - Bjelke med lengde 10000 millimeter

Hvis en bruker høyreklikker på *internal joint* og klikker videre på **Properties** blir dialogboksen som er vist i figur 70 produsert. Her er det mulig for en bruker å skrive inn ny x-verdi for posisjonen. På den gamle versjonen var denne funksjonen mangelfull og *internal joint* flyttet seg ikke dit brukeren ville. Figur 71 viser samme bjelken etter et forsøk på å flytte *internal joint* til x-koordinat 2500, og figur 72 viser samme forflytning i den nye versjonen. Uansett hvilken x-koordinat en bruker vil flytte dette *internal joint* ender med at den legger seg på koordinat 500. Det er og lagt inn slik at en på en horisontal bjelke er det ikke mulig å skrive inn en z-koordinat, og på vertikal bjelke er det ikke mulig å skrive inn en x-koordinat.



Figur 70 - Internal joint properties



Figur 71 - Forsøk på å flytte *internal joint* til x-koordinat 2500 i versjon 2.0



Figur 72 - Forsøk på å flytte *internal joint* til x-koordinat 2500 i versjon 3.0

4.6.2 Lineload

4.6.2.1 Edit line loads

Teksten er endret fra **Show line load loads** til **Edit line loads**.

Det er nå mulig å redigere flere linjelaster samtidig. Det er kun mulig å redigere intensitetene til disse lastene, og ingenting annet. Det er lagt inn test slik at det ikke er lov å redigere intensitetene til null, da kommer det en beskjed om dette som er vist i figur 19.

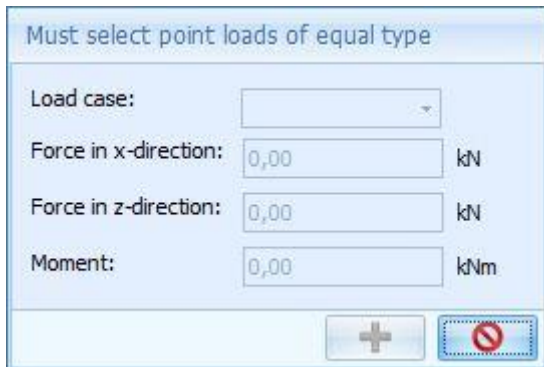
4.6.2.2 Reset to original position

Hvis en bruker har flyttet på en linjelast er det mulig å høyreklikke på lasten og klikke videre på **Reset to original position** som flytter linjelasten tilbake til den *member* den tilhører.

4.6.3 Point load

4.6.3.1 Edit loads

Det er nå mulig å redigere flere punktlaster samtidig. Det er kun mulig å endre på størrelsen til lastene. Det er og kun lov å redigere flere laster av lik type, figur 73 viser et forsøk på å redigere et moment og en kraft i z-retningen.

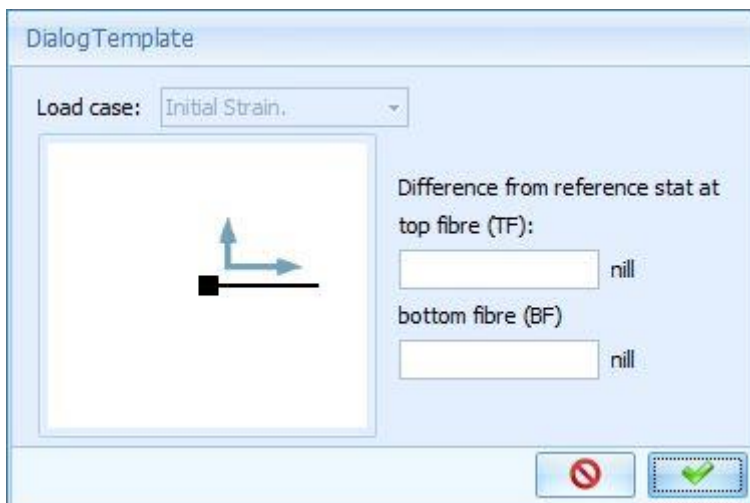


Figur 73 - Forsøk på å redigere moment og kraft i z-retning

4.6.4 Member

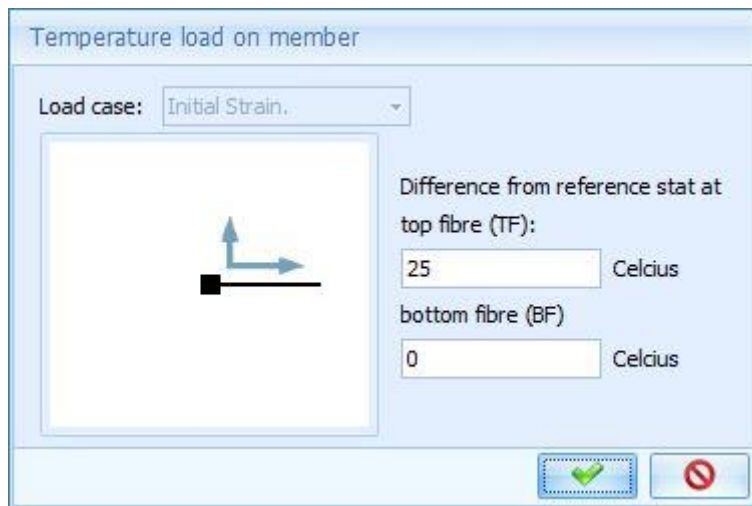
4.6.4.1 Initial strain/temp

Det er mulig å redigere initiell tøyning og initiell temperatur på *members* i modellen. I versjon 2.0 var redigering av initiell temperatur mangelfull, da dialogboksen som ble produsert ikke viste noe informasjon om størrelse eller hva slags initiell tøyning det var. Figur 74 viser dialogboksen til en bjelke som har initiell temperatur i versjon 2.0.



Figur 74 - Redigering av initiell temperatur i versjon 2.0

Denne er nå forbedret i den nye versjonen, den nye dialogboksen er vist i figur 75.

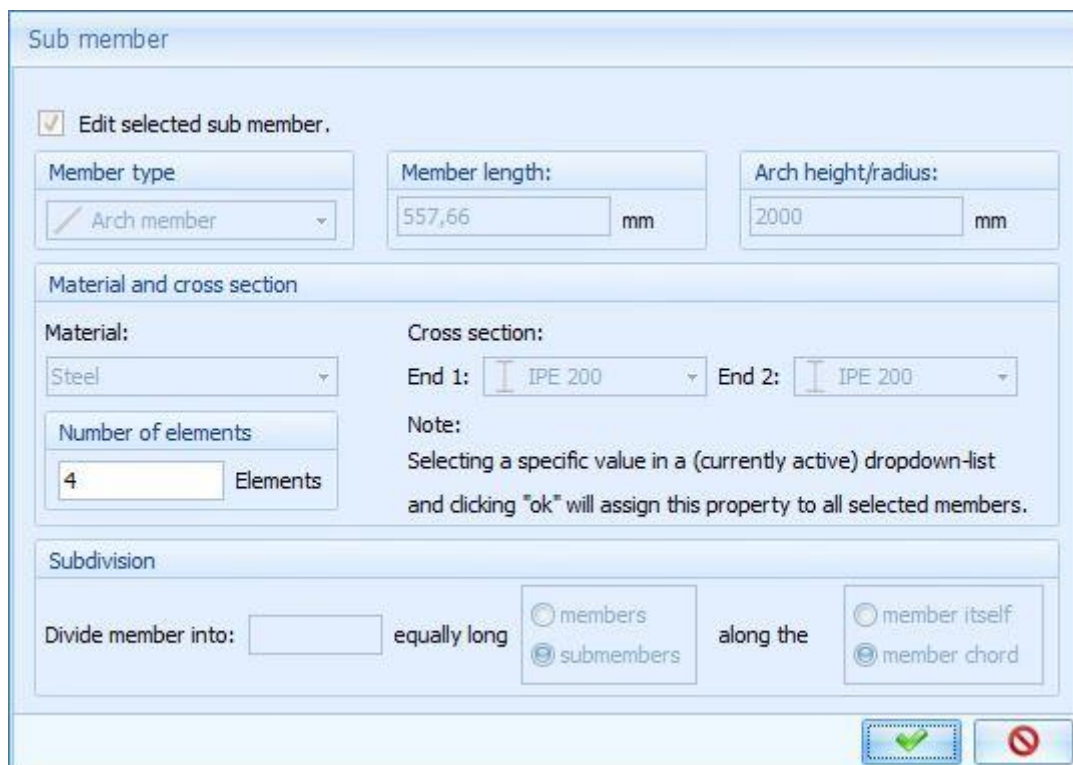


Figur 75 - Redigering av initiell temperatur i versjon 3.0

Det er og blitt lagt til mulighet for å fjerne initiell tøyning og initiell temperatur fra *members* ved å høyreklikke på en *member* for så å klikke videre på **Remove initial strain/temp.**

4.6.4.2 Sub-member properties

Dialogboksen til en *sub-member* er vist i figur 76. Dialogboksen er endret i den nye versjonen og den viser nå lengde på *members*. Dialogboksen i figur 76 viser egenskapene til en *sub-member* som tilhører en bøyd bjelke.



Figur 76 - Egenskaper til en *sub-member* som er krummet

Det er kun mulig å endre på antall elementer til en *sub-member*.

4.6.4.3 Member properties

Dialogboksen til en *member* er lik den til *sub-member*, men det er lov å redigere på flere egenskaper. Figur 77 viser dialogboksen for en *beam member*. I versjon 2.0 var det ikke mulig å dele denne inn nye *members*, det er det mulighet for i versjon 3.0. Denne dialogboksen viser antall elementer, og det er summert over alle tilhørende *sub-members*. Hvis en bruker endrer inndelingen til en *sub-member* vil antall elementer til *host-member* bli endret, dette ble ikke gjort i versjon 2.0.

The screenshot shows the 'Member 1' dialog box. It has a title bar 'Member 1' and a checked box 'Edit selected member.'. The 'Member type' dropdown is set to 'Beam member'. 'Member length' is 5500 mm and 'Arch height/radius' is <None> mm. Under 'Material and cross section', 'Material' is Steel and 'Cross section' has 'End 1' and 'End 2' both set to IPE 200. 'Number of elements' is 50. The 'Subdivision' section has 'Divide member into: 0 equally long' and radio buttons for 'members', 'submembers', 'member itself', and 'member chord'. At the bottom are 'OK' and 'Cancel' buttons.

Figur 77 - Beam member properties

Figur 78 viser dialogboksen for en *arch member*. Denne er lik som *beam member* med unntak av det er mulig å redigere høyden og det er ikke mulig å dele inn i *members*, kun *sub-members*.

The screenshot shows the 'Member 2' dialog box. It has a title bar 'Member 2' and a checked box 'Edit selected member.'. The 'Member type' dropdown is set to 'Arch member'. 'Member length' is 7064,8 mm and 'Arch height/radius' is 2000 mm. Under 'Material and cross section', 'Material' is Steel and 'Cross section' has 'End 1' and 'End 2' both set to IPE 200. 'Number of elements' is 54. The 'Subdivision' section has 'Divide member into: 0 equally long' and radio buttons for 'members', 'submembers', 'member itself', and 'member chord'. At the bottom are 'OK' and 'Cancel' buttons.

Figur 78 - Arched member properties

Figur 79 viser dialogboksen for en *bar member*. I versjon 2.0 ble denne vist som at den hadde 40 elementer, men en *bar member* vil alltid kun ha ett element. Denne vises korrekt i versjon 3.0 og den er ikke mulig å endre for en *bar member*. Vil en bruker endre denne til en *beam member* vil antall elementer bli satt korrekt og det blir gitt mulighet for å endre elementinndelingen.

Figur 79 - Bar member properties

Det var og noen mangler i versjon 2.0 hva gjaldt hvis en bruker merket flere *members* og har ønske om å endre på egenskapene. Figur 80 viser en *beam member* og en *arch member* som begge er merket og egenskapene ønskes endret. Det er her mulighet for en bruker å endre på høyden, selv om den ene *member* ikke er en *arch member*.

Figur 80 - Beam member og arch member properties i versjon 2.0

Figur 81 viser dialogboksen for en *beam member*, en *arch member* og en *bar member* i versjon 3.0. Det er blitt lagt inn restriksjoner på hva som er mulig å endre. Programmet gjenkjenner hva slags

members som ønskes redigert og gir brukeren tilgang på begrenset område som kan endres hvis det er flere *members* merket.

Figur 81 - *Beam member, arch member og bar member properties* i versjon 3.0

Alle disse *members* har ulik elementinndeling, dermed vises **<Multiple values>**. I versjon 2.0 var det ikke mulig å åpne denne dialogboksen med flere *members* hvis de hadde ulik elementinndeling. Da produserte programmet en *exception* som kan sees i figur 82.



Figur 82 - Forsøk på å åpne *properties* for flere *members* med ulik elementinndeling i versjon 2.0

Det var og et problem med at dialogboksen i versjon 2.0 viste antall materialer dobbelt opp når flere *members* var merket, dette er rettet opp i versjon 3.0.

5. OpenGL

All tegning i modellvinduet er skrevet med OpenGL sitt API. Det er flere tegnemetoder i programmet og noen av de er endret i versjon 3.0.

5.1 Joints

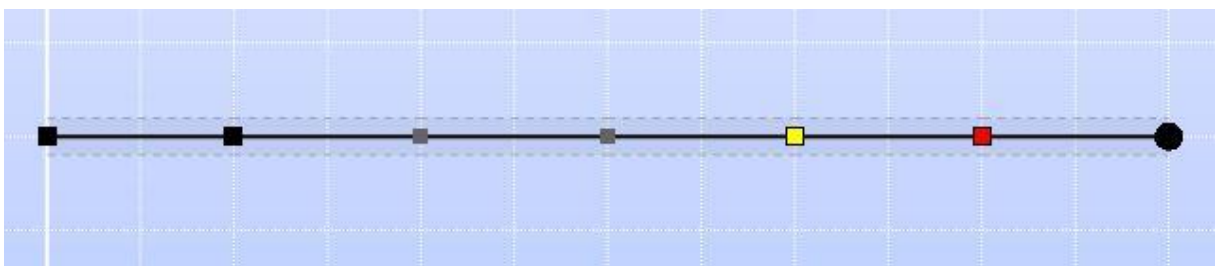
Joints er tegnet som punkter ved bruk av `Gl.Points`. Størrelsen og fargen på disse punktene er avhengig om det er et vanlig *joint* eller om det er et *internal joint*. Et *joint* får størrelsen 10 og farge med `rgba`-verdier lik 0,0,0,255, som gir sorte *joints*. Et *internal joint* får størrelsen lik 8 og farge med `rgba`-verdier lik 100,100,100,255, som gir grå *internal joints*. Det har vært litt problematisk å få disse tegnet som firkanter, siden driverne til noen skjermkort tegner punkter som firkanter og noen skjermkort tegner punkter som rundinger. Løsningen ble å skru av *multisampling* ved hjelp av kommandoen `Gl.glDisable(Gl.GL_MULTISAMPLE)`; *Antialiasing* og *blending* er skrudd av gjennom hele programmet, så disse trengtes ikke gjøres noe med.

Punktmasse blir tegnet opp med lik farge som *joints*, men med størrelse 15. Punktmasse blir tvunget til å bli tegnet som ett rundt punkt, for å enklere vise at det er punktmasse og ikke et stort *joint*. For at alle skjermkortene skal tegne punktmasse som et rundt punkt blir disse kommandoene brukt:

```
Gl.glEnable(Gl.GL_ALPHA_TEST);
Gl.glAlphaFunc(Gl.GL_NOTEQUAL, 0);
Gl.glEnable(Gl.GL_BLEND);
Gl.glEnable(Gl.GL_POINT_SMOOTH);
Gl.glHint(Gl.GL_POINT_SMOOTH_HINT, Gl.GL_NICEST);
```

Når et *joint* eller *internal joint* blir merket med muspekeren blir det tegnet et punkt på størrelse 8 og med `rgba`-verdier lik 255,255,0,102, noe som gir en gul farge. Hvis det eksisterer flere *joints* på samme plassering blir det tegnet et punkt med størrelse 8 og med `rgba`-verdier lik 255,0,0,204, som gir en rød farge. Da er det enkelt å se om et *joint* er merket eller om det ligger flere *joints* oppå hverandre.

Figur 84 viser en *member* med *joints*, *internal joints*, *joint* som er merket, to *joints* som ligger oppå hverandre og punktmasse.



Figur 83 - Tegning av *joints* og punktmasse

5.2 Members

Members blir tegnet som linjer ved bruk av `Gl.Lines`. Hvis knappen **Show member outline** er trykt inn i **Toolbox** blir det tegnet stiplede linjer på oversiden og undersiden av *members* ved bruk av `Gl.Lines` og de blir tegnet opp stiplede ved hjelp av kommandoen `Gl.glEnable(Gl.GL_LINE_STIPPLE)`; . Tabellen nedenfor viser de forskjellige fargene *members* blir tegnet i, og hvor disse fargene er i bruk.

RBGA-verdier	Farge	I bruk
0,0,0,13	Sort	<i>Beam</i> og <i>arch beam members</i>
128,0,255,13	Lilla	<i>Bar, strut</i> og <i>cable members</i>
255,255,0,102	Gul	En <i>member</i> som er merket
255,204,0,13	Oransje	En <i>member</i> som har en lokal formfeil og brukeren er inne på fanen Analysis , og knappen Non linear static er trykt ned
0,179,0,51	Grønn	En <i>member</i> som har en definert en load path , enten gjennom influenslinjer eller lasttog, og brukeren er inne på fanen Analysis og en av knappene Influence lines eller Real time analysis er trykt ned

Tabell 4 - Farge på *members*

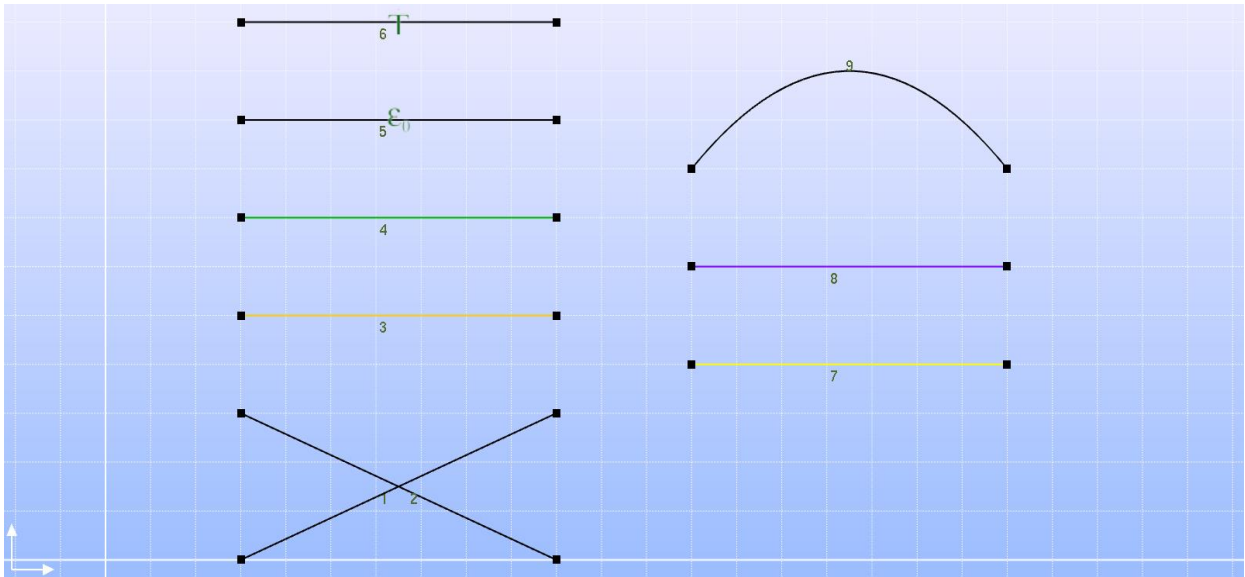
En *member* får og tegnet opp et symbol som viser om den inneholder initiell tøyning eller initiell temperatur. Tegnetometoden som viser id-numrene til *members* har fått en ansiktsløftning ved at tallene ikke har sirkler rundt seg, og de blir tegnet opp med rbga-verdier 85,107,47,255, som gir en mørk grønn farge. Det var en svakhet i versjon 2.0 at *members* som stod i kryss fikk id-numrene tegnet på samme plass, og *arch members* sine id-numre ikke ble tegnet opp langs krumningen. Dette er blitt endret på, og tegning av id-numre for *members* fungerer bedre i versjon 3.0.

Figur 84 viser en modell med flere *members* som har id-nummer og forskjellig farger etter hva de er. Og tabellen nedenfor gir en forklaring til hva de forskjellige *members* er.

Id	Type member
1	<i>Beam member</i> i kryss med <i>member</i> nummer 2
2	<i>Beam member</i> i kryss med <i>member</i> nummer 1
3	<i>Beam member</i> med lokal formfeil
4	<i>Beam member</i> med definert <i>load path</i>
5	<i>Beam member</i> med initiell tøyning
6	<i>Beam member</i> med initiell temperatur
7	<i>Beam member</i> som er merket
8	<i>Bar member</i>
9	<i>Arch member</i>

Tabell 5 - Forklaring til figur 83

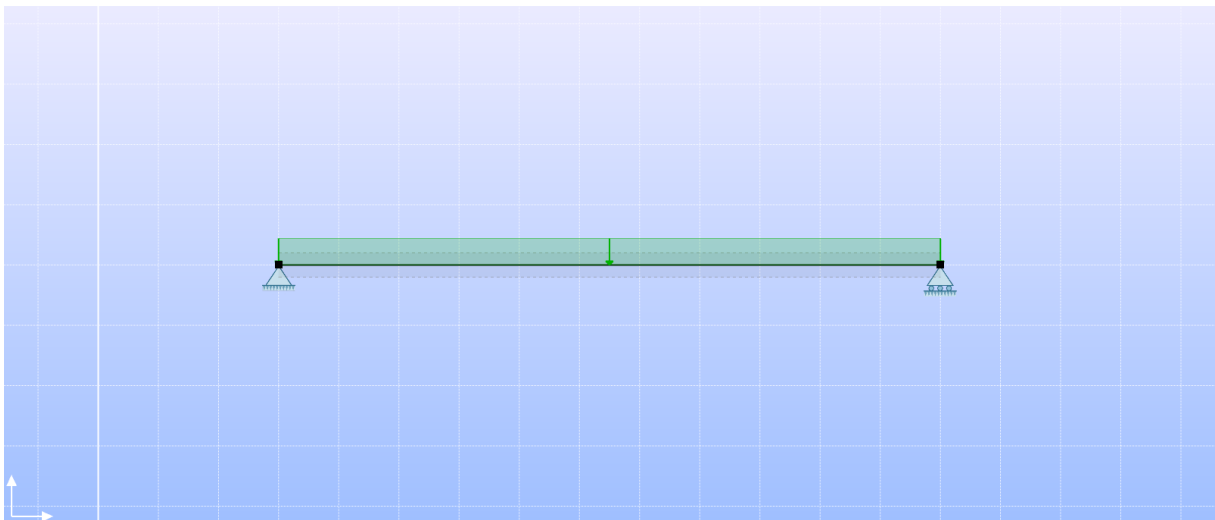
De forskjellige fargene, på *joints* og *members*, er blitt hentet fra nettsiden C# Color Table[13].



Figur 84 - Diverse members

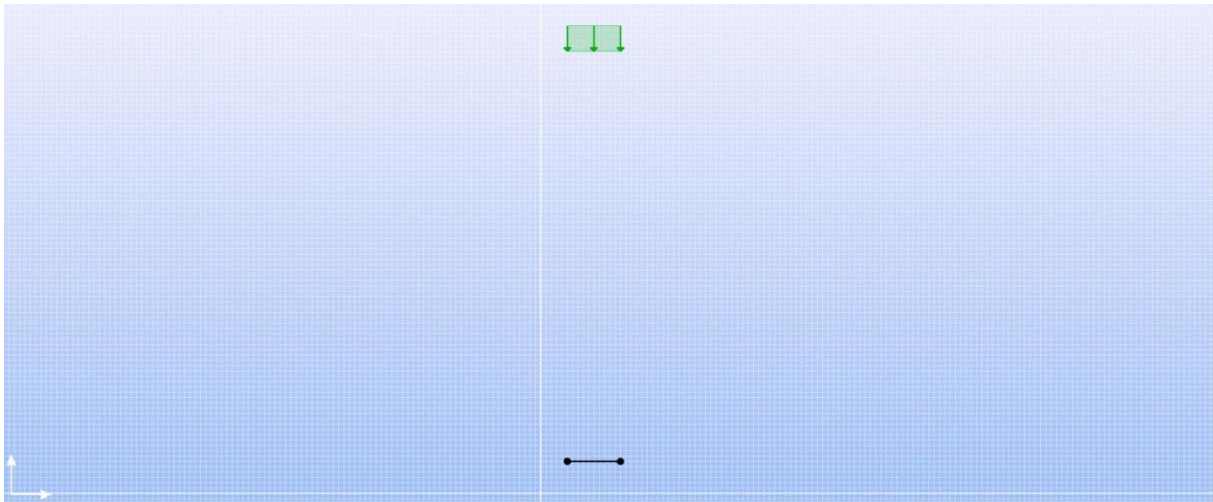
5.3 Linjelaster

Det var et problem i versjon 2.0 med flytting av linjelaster. Tok en bruker tak i en linjelast og flyttet på den forsvant den langt avgårde. Og ble linjelasten så flyttet ut av skjermen forsvant den langt oppover fra modellen. Figur 85 viser en modell av en fritt opplagt bjelke med en linjelast.



Figur 85 - Fritt opplagt bjelke

Tok en bruker tak i denne linjelasten og flyttet den noen hakk oppover resulterte det i forflytningen som er vist i figur 86. I versjon 3.0 er dette korrigert. Det er i tillegg lagt inn slik at det ikke finnes mulighet for å flytte linjelaster, og punktlaster, ut av modellvinduet.



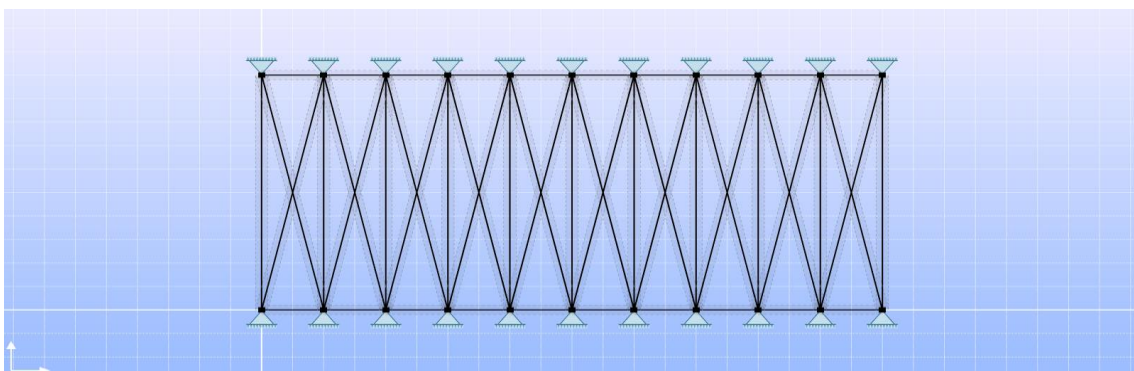
Figur 86 - Flytting av linjelast i versjon 2.0

5.2 Finne-metoder

Det eksisterer flere metoder i koden som går ut på å finne ut hva slags objekter som er under muspekeren. Den som blir oftest kalt er metoden som finner objekter i modellvinduet. Denne tegner opp *joints*, *members* og laster. Så finner den ut hvilket objekt som brukeren da klikker på. Denne metoden ble og kalt i metoden som kalles når muspekeren «hviler» i modellvinduet. Problemet med sistnevnte er at `mouseHover` blir kalt ganske ofte, og det medførte at modellen ble tegnet opp på nytt hele tiden når muspekeren var inne i modellvinduet og det medførte at programmet ble tregere og tregere jo større modellen ble.

`FindObject()`-metoden er blitt fjernet fra `mouseHover` i den nye versjonen. `MouseHover` blir brukt til å generere *tooltips* for *members* i modellvinduet. Det er istedenfor blitt implementert en metode som itererer over alle *members* i modellvinduet og finner ut hvilken *member* muspekeren er over. Siden `mouseHover` kun blir brukt til å vise *tooltips* for *members*, ble tiden som `mouseHover` bruker kraftig redusert og resulterte i et raskere og mer responsivt program.

Figur 87 viser en modell med en del *members* (56 stk), *joints* (22 stk) og opplagre (22 stk). På versjon 2.0 brukte et kall til `mouseHover` rundt 160 millisekunder. På den nye versjonen bruker et kall til `mouseHover` 0-1 millisekunder. Og siden `mouseHover` blir kalt ganske ofte er dette en ganske drastisk forskjell.



Figur 87 - Modell med mange *members*

Det er og blitt implementert nye metoder for å finne objekter under muspekeren ved musklikk i modellvinduet. Dette gjelder for å finne *joints* og *members* i fanene **Modeling**, **Loading** og **Analysis**. I fanen **Results** er det blitt implementert en ny metode for å finne *joints* og elementer ved musklikk. Metoden som finner *joints* og *members* går gjennom alle *joints* og *members* i modellen og undersøker om punktet som ligger under musklikket inneholder noen av disse objektene. Det samme skjer når metoden skal finne *joints* og elementer. Dette kutter tegnetiden drastisk gjennom hele programmet, siden musklikk er noe en bruker gjør ofte.

For å sammenligne musklikk fra versjon 2.0 til versjon 3.0 viser tabellen under tiden det tar for et musklikk på modellen vist i figur 86, det er da 56 *members* og 22 *joints* i modellen.

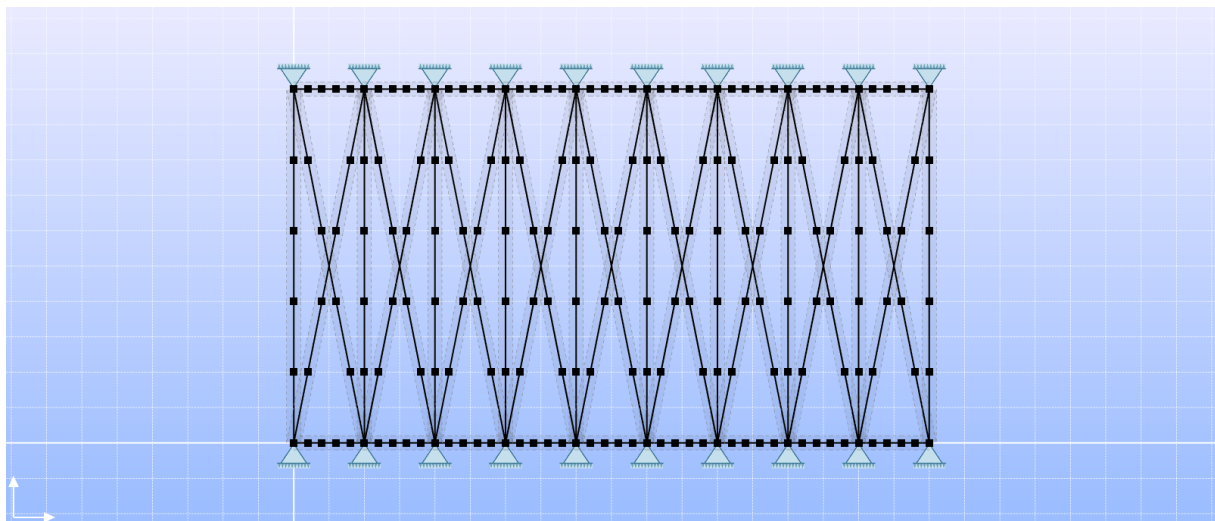
Musklikk	Tid i versjon 2.0	Tid i versjon 3.0
Venstre	158 millisekunder	1 millisekund
Høyre	229 millisekunder	2 millisekunder

Tabell 6 - Tid for musklikk i modell vist i figur 87

Tabell 7 viser musklikk på samme modellen vist i figur 88. Dette er samme modell fra figur 87, men hver *member* er delt inn i fem nye *members* slik at modellen nå består av 230 *members* og 204 *joints*.

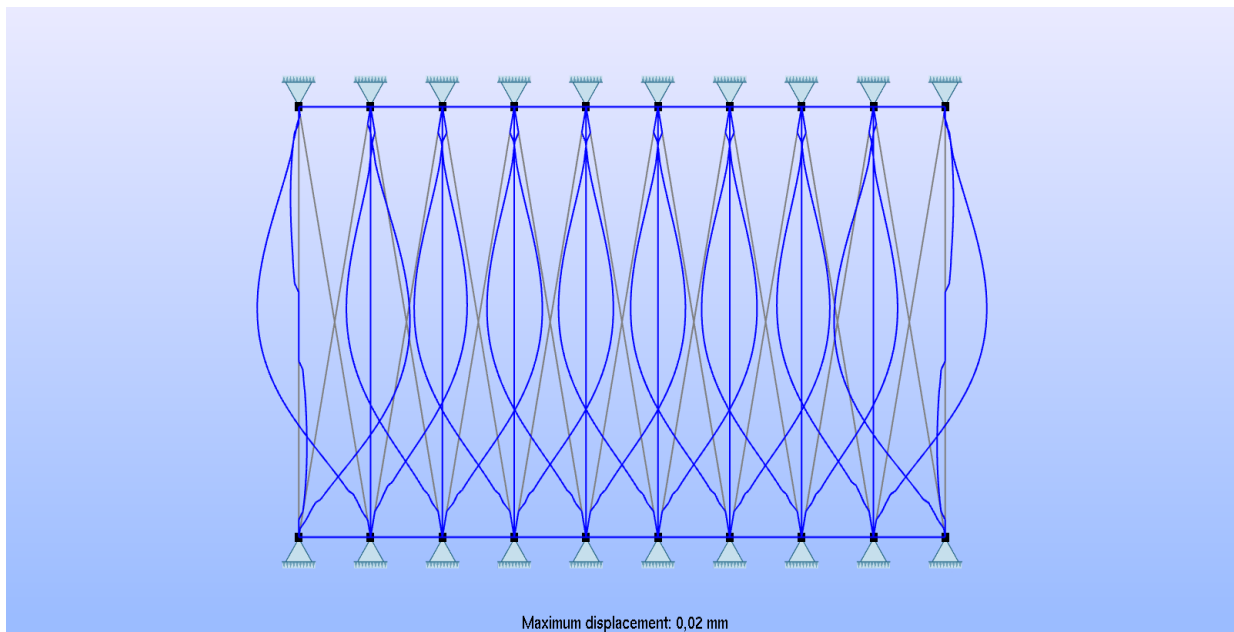
Musklikk	Tid i versjon 2.0	Tid i versjon 3.0
Venstre	220 millisekunder	3 millisekund
Høyre	355 millisekunder	8 millisekunder

Tabell 7- Tid for musklikk i modell vist i figur 88



Figur 88 - Modell fra figur 87 endret til å inneholde flere *members*

Figur 89 viser forskyvningdiagram for modellen i figur 87, regnet med egenvekt og med lineær statisk analyse.



Figur 89 - Forskyvningdiagram for modellen i figur 87

For å sammenligne musklikk fra versjon 2.0 til versjon 3.0 viser tabellen under tiden det tar for et musklikk på forskyvningdiagrammet. Det er en elementinndeling på 50 per *member*, det er da totalt 2800 elementer med i diagrammet. Det er og tatt tiden for en elementinndeling på 100 elementer per *member*, da totalt 5600 elementer med i diagrammet. Det er fjernet mulighet for venstreklikk i fanen **Results**, siden det kun er høyreklikk som har en funksjon. Det er enda mulighet for å klikke på *joints* for å merke de (denne funksjonaliteten ligger i et annen *event*), bruke **Zoom to rectangle** og **Move model** som ligger i **Toolbox**.

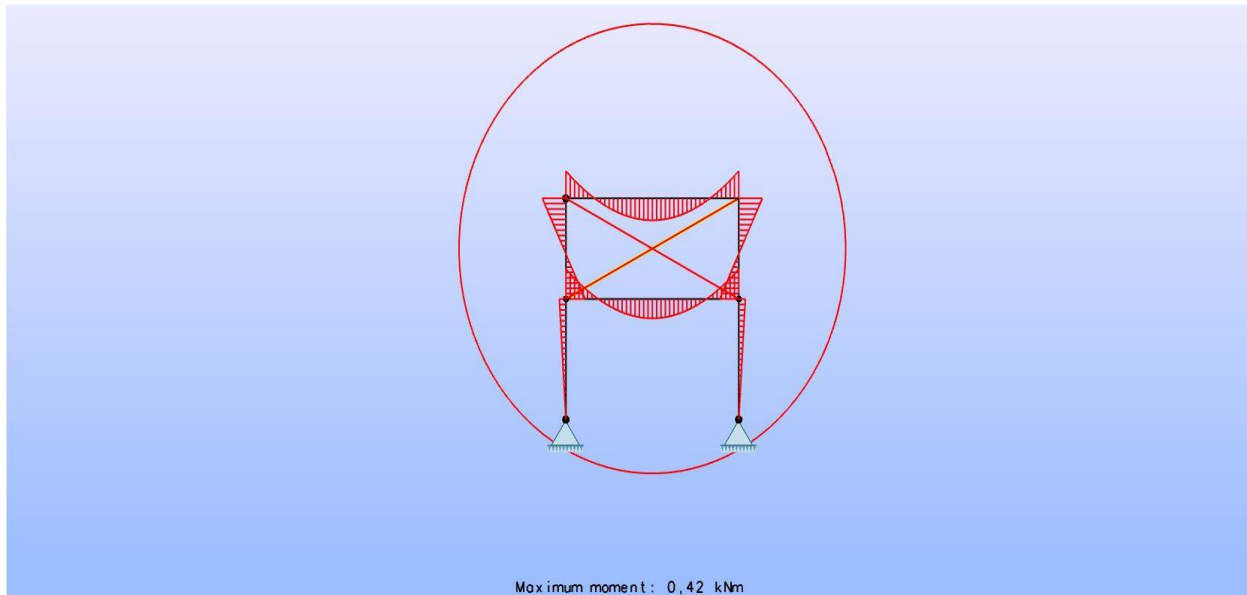
Musklikk	Tid i versjon 2.0	Tid i versjon 3.0
Venstre med 2800 elementer	301 millisekunder	0 millisekunder
Høyre med 2800 elementer	534 millisekunder	32 millisekunder
Venstre med 5600 elementer	500 millisekunder	0 millisekunder
Høyre med 5600 elementer	1120 millisekunder	56 millisekunder

Tabell 8 - Tid for musklikk i forskyvningdiagram vist i figur 89

Tiden for et musklikk har gått drastisk ned fra tidligere versjon. Dette hjelper til å få programmet til å virke mye mer responsivt, da spesielt for større modeller med mye objekter.

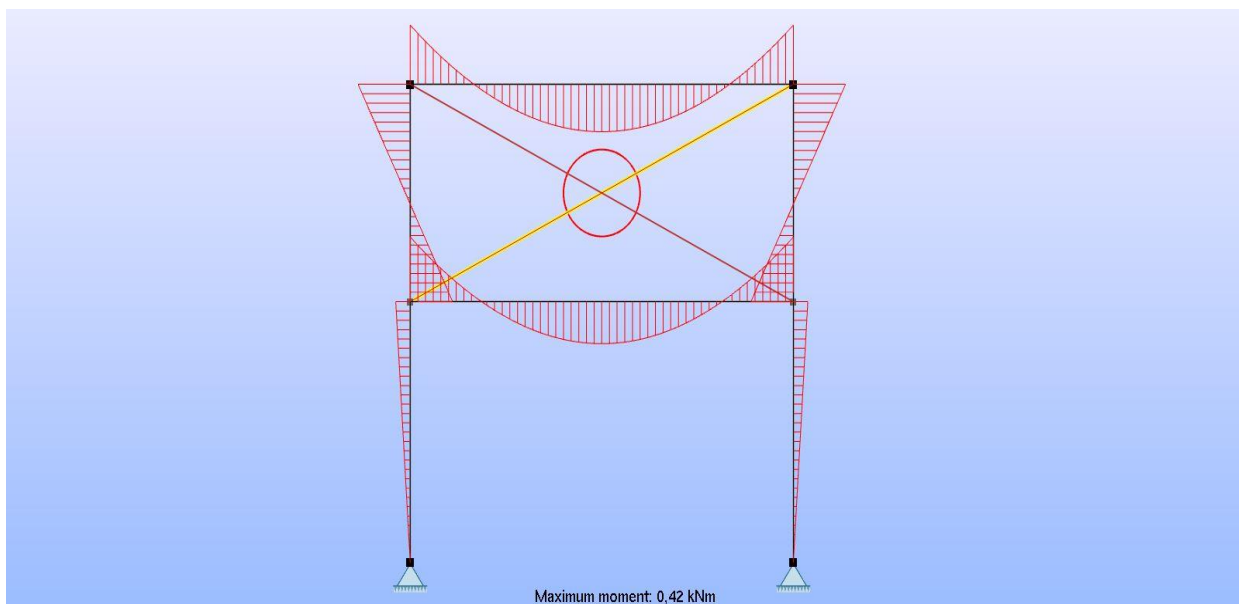
5.3 Results

Når en bruker høyreklikker på et element i resultatvinduet vil det tegnes en liten sirkel rundt det elementet brukeren klikker på. Størrelsen på denne sirkelen bestemmes ut fra hvor stort elementet er. En *member* av typen *bar*, *strut* og *cable* har kun ett element og denne sirkelen ble meget stor i versjon 2.0. Se figur 90 for eksempel.



Figur 90 - *Bar member* høyreklikket på i momentdiagram i versjon 2.0

Figur 91 viser den samme rammen, og hvor stor denne sirkelen er i den nye versjonen.



Figur 91 - *Bar member* høyreklikket på i momentdiagram i versjon 3.0

Muligheten for å flytte og slette *joints* er fjernet i **Results**. Det er fordi det ikke skal være mulig for en bruker å endre på modellen i denne fanen.

6. Videre arbeid

6.1 Lagring av innstillinger

Det bør taes en gjennomgang om det skal lagres enda mer innstillinger. Lokale innstillinger lagres nå sammen med modellen. Globale innstillinger blir lagret til en fil som blir åpnet ved start av programmet. Det er de globale innstillingene som ligger under **fap2D Settings** som blir lagret. Burde og ha en gjennomgang om brukeren skal ha mulighet til å endre på, og lagre, enda mer globale innstillinger.

6.2 Lagring av resultater

Selv om resultatene nå følger modellen blir ikke resultatene lagret sammen med modellen. Dette er noe en bør ta en diskusjon rundt og undersøke om det er verdt å implementere. Med dagens datamaskiner og regnekraft tar ikke en analyse mange sekundene, og det bør diskuteres om det er fordelaktig om dette arbeidet er verdt det.

6.3 Dialogbokser

Cancel-knappene er fjernet fra alle dialogbokser fra fanene **Analysis** og **Results**. En jobb er å implementere slike her, og legge ved funksjon som gjør at all endring en bruker har gjort i dialogboksen blir ugjort. Alle dialogboksene i hele programmet må og taes en gjennomgang på og sørge for at alle dialogboksene gjør det de skal.

6.4 Lisensiering

Muligheten for lisensiering er undersøkt våren 2013. Men det ble ikke tid til å gjennomføre dette. Det ble enighet om å ha lisenser liggende på en server eid av NTNU som det er mulig å kople seg mot gjennom VPN. Ved oppstart av programmet henter det en gyldig lisens fra denne serveren. Dette vil gjøre at det kun er studenter og ansatte ved NTNU som har mulighet til å benytte seg av programmet.

6.5 Undo/Redo

Dette har blitt forbedret stort våren 2013, og en *undo/redo* går veldig raskt, spesielt ved bruk av en SSD-harddisk. Men det er enda en svakhet ved at programmet henter ut midlertidige modeller og åpner disse ved hver *undo/redo*. Dette kan bety en stor omlegging av koden, og det bør diskuteres om dette er verdt arbeidet som må legges ned.

6.6 Tegning

Den største svakheten ved programmet per dags dato er tegnemetodene. Spesielt klassen `MultiSampleGLControl` bør gjennomgås av en som er kyndig med OpenGL. Metoden `FindObject(int x, int z)` som ligger i klassen `FrMModel` er blitt endret, å finne *joints*, *members* og elementer har fått egne metoder som er mer effektive enn de var. Men metoden `FindObject(Rectangle area)` er ikke blitt endret, denne bør undersøkes om kan effektiviseres. Det er i tillegg finne-metoder i flere klasser som bør gåes gjennom og som bør effektiviseres.

I versjon 3.0 av fap2D er det OpenGL 2.1 som er blitt benyttet, siden rammeverket Tao Framework[14] er blitt benyttet. Om dette rammeverket kommer i en ny versjon bør det oppdateres. OpenGL har gått gjennom en del endringer siden versjon 2.1 og programmet vil nok tjene stort på å ha en nyere versjon av OpenGL.

6.7 Member data panel

Dette panelet er blitt urørt av utviklerne våren 2013. Og det bør vurderes om denne skal få en oppfriskning. Panelet som viser frem *joints* er en del bedre enn det for *members*, og derfor brukes nesten ikke panelet for *members* noen gang av brukerne av programmet.

6.8 Beta-testing

Å finne en gruppe med studenter som er villige til å teste programmet på daglig basis og få tilbakemeldinger vil nok være nyttig. Hvis programmet er tenkt til å brukes i undervisningen vil det være lurt å få tilbakemeldinger fra studenter som kommer til å benytte seg av programmet. Programmet har iblant oppført seg ulikt på forskjellig maskinvare, så det vil nok og være gunstig å se om programmet fungerer likt på flere typer maskinvare i tillegg.

6.9 Kopiering

En funksjon som kan være nyttig er muligheten til å kopiere objekter i modellvinduet. Spesielt *members* kan være en fin ting å kunne kopiere, da det vil gjøre det raskere å lage en stor modell.

6.10 Stabilitet og sikkerhet

Våren 2013 har gjort **fap2D** til et ganske stabilt og sikkert program. Men det er enda mer å gjøre på denne biten. Finne frem til feil i koden og rette de opp vil nok være nyttig, da det vil føre til enda færre krasjer av programmet og gjøre opplevelsen av **fap2D** enda bedre enn det er per dags dato.

6.11 Slå user controls sammen

Flere dialogbokser inneholder bokser som viser et *joint* med tilhørende *members*. Disse boksene kalles *user controls*. Disse boksene gjør mye av det samme, og det bør undersøkes om det er mulig å slå dette sammen til en klasse. Per dags dato gjør klassene UC_CouplingSpring, UC_Eccentricity, UC_Joint, UC_JointProperties, UC_NonLinearElasticSpring og UC_Release mye av det samme. Å ha dette i en klasse vil gjøre det mer oversiktlig når feil skal finnes og det vil kutte ned en del kode.

6.12 Nummerering av members

Nummerering av *members* er endret på våren 2013, men den har fortsatt noen svakheter. Denne bør utbedres enda litt, slik at nummerering av *members* blir mer nøyaktig. Spesielt på store modeller med mange *members* kan det bli rotete med slik det er per dags dato.

6.13 Events for mus

I modellvinduet (klassen `FrmModel` i koden) har seks *events* for musa. Dette er `MouseClicked`, `MouseDown`, `MouseHover`, `MouseLeave`, `MouseMove` og `MouseUp`. Disse bør gjennomgås og se om de kan effektiviseres. Det er blitt en del endringer i disse i versjon 3.0 fra versjon 2.0, spesielt antall *repaints* er redusert i disse. `MouseClicked` vil heller ikke gjøre noenting ved venstre musklick i resultatfanen, siden venstre musklick ikke har noen funksjon her. Programmet slipper da å kjøre gjennom unødvendig kode og responsen øker.

Ved en gjennomgang av disse bør det undersøkes om hvor musklick, draging med mus nede, når musklicket slutter osv. kan gjøres slik at koden kun blir kjørt gjennom der de forskjellige *events* har en funksjon. Pga at tegnetiden er redusert og finne-metoden for modellvinduet er effektivisert vil disse *events* ikke ta like lang tid som det gjorde i versjon 2.0, men det er enda behov for å effektivisere disse slik at responsen i programmet øker.

7. Referanseliste

1. **Dolven, Jan Kristian.** *Further development of GUI for fap2D.* NTNU : Masteroppgave, Inst. for konstruksjonsteknikk, 2010.
2. **Vågenes Aase, Daniel og Årvik, Brita.** *fap2D - Forbedring og videreutvikling av GUI.* NTNU : Masteroppgave, Inst. for konstruksjonsteknikk, 2011.
3. **Dolven, Jan Kristian og Larsen, Fredrik.** *GUI for fap2D.* NTNU : Prosjektoppgave, Inst. for konstruksjonsteknikk, 2009.
4. **Bell, Kolbein.** *fap2D - A WINDOWS-based program for static and dynamic analysis of 2D frame type structures.* NTNU : Inst. for konstruksjonsteknikk, 2013.
5. **Microsoft.** Office products. *Office.* [Internett] Microsoft Corporation, 2010.
<http://office.microsoft.com/nb-no/products/?CTT=97>.
6. **Pedersen, Kristian.** *Kritisk vurdering av programmet fap2D med tanke på enkel og sikker bruk.* NTNU : Prosjektoppgave, Inst. for konstruksjonsteknikk, 2012.
7. **Collaborative.** Visual C# forum. *Microsoft developer network.* [Internett] Microsoft Corporation.
<http://social.msdn.microsoft.com/Forums/en-US/csharpgeneral/threads>.
8. **Zuckerberg, Mark.** Facebook. [Internett] Facebook, Inc, 2004. <https://www.facebook.com/>.
9. **Division, Microsoft Skype.** Skype download. *Skype.* [Internett] Microsoft Corporation, 2013.
<http://www.skype.com/no/download-skype/skype-for-computer/>.
10. **Dropbox, Inc.** Dropbox download. *Dropbox.* [Internett] Dropbox, Inc, 2013.
<https://www.dropbox.com/downloading?src=index>.
11. **Microsoft.** *Notepad.* Redmond : Microsoft Corporation, 1985.
12. **Croissant, Yannick og Kath.** Ajaxload. [Internett] <http://ajaxload.info>.
13. **Newcomer, Joseph M.** C# Color Name Table. [Internett] Floundercraft Ltd., 1998.
http://www.flounder.com/csharp_color_table.htm.
14. **Ridge, Randy.** Tao Framework. [Internett] Sourceforge, 2013.
<http://sourceforge.net/projects/taoframework/>
15. **Microsoft.** *Windows 7.* Redmond : Microsoft Corporation, 2009.
16. —. *Windows 8.* Redmond : Microsoft Corporation, 2012.
17. —. *Windows Vista.* Redmond : Microsoft Corporation, 2006.
18. —. *Visual Studio 10.* Redmond : Microsoft Corporation, 2010.
19. —. Visual Studio 12 Ultimate. *Visual Studio.* [Internett] Microsoft Corporation, 2012.
<http://www.microsoft.com/visualstudio/eng/products/visual-studio-ultimate-2012>.

20. **Subversion, Apache.** VisualSVN Download. *VisuaSVN*. [Internett] Apache Software Foundation, 2013. <http://www.visualsvn.com/visualsvn/download/>.
21. **DevExpress.** DXperience. *DevExpress*. [Internett] Developer Express Inc., 2012. <http://www.devexpress.com/Subscriptions/DXperience.xml>.
22. **Tatham, Simon.** PuTTY Download Page. *chiark home page*. [Internett] Team PuTTY, 2013. <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
23. —. PuTTY Download Page. *chiark home page*. [Internett] Team PuTTY, 2013. <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
24. —. PuTTY Download Page. *chiark home page*. [Internett] Team PuTTY, 2013. <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
25. **Kemp, Tim og Küng, Stefan.** TortoiseSVN. *Tigris*. [Internett] TortoiseSVN, 2013. <http://tortoisesvn.tigris.org/>.
26. **Software, Flexera.** Installshield: Resources. *Installshield*. [Internett] Flexera Software, 2012. <http://www.flexerasoftware.com/products/installshield/resources.htm>.
27. **Microsoft.** Microsoft .NET Framework 4.5. *Download center*. [Internett] Microsoft Corporation, 2012. <http://www.microsoft.com/nb-no/download/details.aspx?id=30653>.
28. —. Microsoft Download Center. *Access 2007 Download: Access Runtime*. [Internett] Microsoft Corporation, 2007. <http://www.microsoft.com/en-us/download/details.aspx?id=4438>.
29. **Bell, Kolbein.** *fap2D note 3*. NTNU : Inst. for konstruksjonsteknikk, 2009.

A. Logg

Loggen er skrevet etter endt arbeidsdag. Og noen ganger er det blitt skrevet ned noen tanker som er gjort rundt programmet mens arbeidet pågår. Loggen er skrevet med mye muntlig språkbruk og formateringen beskrevet i tabell 1 er ikke brukt i loggen. Det er skrevet de aktuelle klassene som er arbeidet med den dagen. Bortsett fra under opprydding og kommentering, for da er det blitt jobbet med nesten alle klassene i prosjektet.

Uke 3

15/01-2013

Slet med å installere DevExpress til VS 2012, det ble mye feil under kompilering av prosjektene. Løsning ble å fjerne alle referansene til DevExpress, for så å legge de inn igjen. Siden VS 2012 ikke støtter å lage egne installasjonsfiler måtte setup-prosjektet fjernes. Brukt programmet og forsøkt og reprodusere en del feil de andre har meldt inn.

16/01-2013

Ryddet litt kode og sletta mange variabler som ikke er i bruk. En metode som heter `OpenGLControl_MouseHover` i klassen `UC_Graph` som har «return» som første linje, dvs at den faktisk ikke gjør noe som helst. Etter å ha sett gjennom mye av koden har det gått opp for meg at det er mye rot og at det kommer til å ta en stund å få ryddet opp i alt.

17/01-2013

Oversatt mye OpenGL-kode fra TaoFramework til OpenTK. Fordelen ved å bruke OpenTK er at den bruker syntaksen til .Net, har en mye bedre intellisense og den støtter opptil OpenGL 3.0 mens Tao kun har støtte for opptil OpenGL 2.1. Ulempen er at `glu` og `glut` ikke er inkludert, men `glu`-funksjonaliteten kan finnes igjen i `Matrix4`-klassen til OpenTK. Det er en innebygd metode i `glu` som ikke er inkludert, nemlig `PickMatrix()` og jeg måtte lage en egen metode som gjør det samme. Teksten i programmet er brukt av en metode i `glut`, så foreløpig må jeg la den være i fred til jeg finner en løsning.

18/01-2013

Ferdig med å oversette all OpenGL-koden fra Tao til OpenTK, men har støtt på et stort problem. Trenger en context for å kunne bruke OpenTK, og bruke OpenGL-kall til å tegne. Ifølge forfatterne bak OpenTK er det mulig å lage en `DummyContext` hvis man skal oversette til OpenTK, men den fungerer ikke slik den skal. Har forsøkt flere omveier rundt denne, men får kun `ArgumentOutOfRangeException` og programmet stopper opp. Får heller la OpenTK ligge og fortsette med Tao. Forfatteren bak Tao har sagt at det skal komme ny versjon denne våren som skal støtte OpenGL opptil versjon 4.0, så får håpe at dette faktisk skjer.

19/01-2013

Har fjernet alle variabler som ikke er i bruk. Etter å ha gått gjennom mye av koden har jeg lagt merke til at det er mye kommentering som mangler og som ikke er godt nok skrevet. Siden dette ikke er prioritet for øyeblikket skriver jeg det ned og ser på det senere. Har og satt meg inn i bruken av OpenGL, siden alt av tegning benytter metoder fra OpenGL-biblioteket. Det er særlig klassen `MultiSampleGLControl` jeg må dykke dypere inn i, denne er stor og komplisert.

Uke 4

21/01-2013

Gått gjennom noen tutorials for OpenGL og sett på hvordan klassene i fap2D er koplet med hverandre. En del opprydding i koden og fjerning av unødvendige mellomrom for å kutte ned på antall kodelinjer. Har og delt inn en del kode inn i regioner. Ferdig ryddet i prosjektene fap2D.Computations og fap2D.Design.

22/01-2013

Syk idag, så lite prosuktiv. Startet såvidt med å rydde opp i fap2D.Gui.

23/01-2013

Ryddet en del opp i fap2D.Gui. Mye som ikke var delt inn i regioner og ser ut til at det er mye som er programmert i høst. Så må ta en grundig gjennomgang når opprydningsarbeidet er ferdig.

24/01-2013

Ny feil oppdaget når en modell skal eksporteres til bilde. Bildet blir ikke noe bra, ser nesten gjennomsiktig ut. Feilen er nok i SaveImage()-metoden i FrmMainGui. Må og gjøre noe med knappene import og print under applikasjonsmenyen, de har ingen funksjon per dags dato. Hvis man bruker CTRL+A for å merke hele modellen blir ikke modellen merket før en rører på musa og metoden RePaintGIForm kalles. Bør og legge til en snarvei til CTRL+S for å lagre modellen. Ferdig med å rydde opp i fap2D.Gui og fap2D.Storing.

25/01-2013

Ryddet opp i fap2D.StructuralData. Her ligger det fra gammelt av en del kommentarer hvor det står TODO, ting som da må gjøres. Må gå tilbake til disse senere og finne ut av om de faktisk er gjort og de TODO-linjene ikke er fjernet eller om det er noe som bør gjøres. Ferdig med å rydde opp i fap2D.Utilities og, unntatt MultiSampleGIForm siden den er såpass kompleks og jeg må sette meg mer inn i den. All kode er hvertfall ryddet opp i og kan starte med å kode.

27/01-2013

La til noen kodelinjer i konstruktøren til FrmMainGui, som finner oppløsningen til skjermen og så setter størrelsen til programmet lik dette. Det gjør at programmet starter opp en del raskere

```
int width = Screen.PrimaryScreen.Bounds.Width;
int height = Screen.PrimaryScreen.Bounds.Height;
this.Size = new Size(width, height);
```

Har og lagt til en RePaintGIForm etter at CTRL+A trykkes inn slik at modellen blir tegnet opp merket etter man trykker inn. La og til snarvei til CTRL+S for å lagre modellen. Og lagt til sjekk til CTRL+Z som kaller Undo(). Det sjekkes om det faktisk er mulig å ta en undo, for trykker man inn CTRL+Z når man han en tom modell så krasjer programmet.

Aktuelle klasser: FrmMainGui

Uke 5

28/01-2013

Møte idag, to uker siden sist. Har installert Putty og koplet opp til serveren der den delte koden ligger. Har sett litt på koden de andre har implementert, og har delt inn alle metoder inn i regioner private, internal, protected og public. Jordskjelvfilene til Erik gir meg noen problemer siden de bruker en absolutt path istedenfor en relativ path. Programmet prøver å aksessere jordskjelvfilene under `c:\fap2D\`, og det er en folder jeg ikke har.

29/01-2013

Har jobbet med `FrmMemberProperties` idag, det var før ikke mulig å dele en member inn i flere members, kun submembers. Det er nå mulig. Men det er ikke mulig å legge inn internal joints på de nye members som blir lagd, det kommer en exception og programmet krasjer.

Pga tidsseriene som blir lagt inn i programmet tar tegnemetodene lang tid, siden de blir tegnet opp på nytt hver gang en repaint blir kalt. Må se på om det er mulig å kun tegne tidsseriene opp en gang, og så ikke tegne de på nytt.

Aktuelle klasser: `FrmMemberProperties`

30/01-2013

Har brukt dagen på å finne ut hvorfor programmet lagrer så mye, og problemet ligger i metoden `UpdateUndoRedoBuffer()` i `FrmModel`. Vi trodde at problemet lå i tegnemetoden, men den bruker i realiteten ikke mange millisekundene på å kjøre. `UpdateUndoRedoBuffer()` er metoden som kalles hvis det er noe som skal legges i undo/redo-lista, så jeg var nødt til å fjerne alle stedene denne ble kalt. Det er ikke lengre mulighet å ta en undo/redo, men det skal jeg uansett kikke på nærmere ved et senere tidspunkt.

Aktuelle klasse: `FrmMainGui`, `FrmModel`, `FrmLoadCaseDialog`, `FrmManageTimeFunctions`, `Member`, `MultisampleGIControl`

31/01-2013

Har sett på alle warnings som VS 12 produserer og har fjernet alle sammen. Har og rettet opp metoden `Savelmage()` som lagrer bilde av modellen. Denne lager uklare bilder pga det blir lagret i png-format og de tar vare på transparency. Transparency blir ikke fjernet fra bilde før det lagres, lagrer nå i jpg-format som ikke tar vare på transparency og nå ser det mye bedre ut.

Har sletta alt som har med ghost-members i programmet. Disse er ikke implementert og det kommer ikke til å bli gjort heller. Det frigjør en del minne og man slipper å få meldingen «not implemented» hvis man prøver å legge ut en ghost member.

Det er noen eksempler her og der på dårlig programmering. Et godt eksempel er konstruktøren til `FrmMainGui` som kaller metoden `InitPages()`. Etter dette blir metoden `InitMainForms()` kalt, og denne metoden kaller `InitPages()` på nytt. Lite poeng i å kalle samme metode flere ganger i konstruktøren.

Aktuelle klasser: `FrmMainGui`, `FrmModel`, `StructuralModel`, `Member`

01/02-2013

Har sett litt på metodene som tegner tekst. Det er brukt glut til å skrive tekst i modellvinduet, slik som tekst i resultatvinduene og nummerering av members og joints. OpenGL har ikke direkte støtte for tekst så må bruke glut. Men her finnes det to metoder å tegne tekst på, og det er Stroke Fonts og Bitmap Fonts. Det er førstnevnte som er blitt implementert. Det er en enkel metode å bruke, men det finnes kun en font i denne metoden og det er roman. Det er ikke en veldig pen skrifttype. Bitmap Fonts er litt mer vrien å implementere, men her finnes det flere skrifttyper.

Klassen som tegner opp grafer, UC_Graph, fungerer ikke så bra. Den tegner blant annet opp jordskjelvene til Erik, og det går treigt. Zooming er heller ikke noe særlig her, så denne må gjennomgås grundig. Har bedt Erik og om å se på den og vi skal nok finne ut av det.

Har gjort slik at Bottom Panel er litt tynnere slik at Toolbox og Left Panel går helt ned. Og har gjort det slik at gruppene Analysis og Scaling inne i Toolbox ikke vises før man er inne på Result-ribbon. Disse var synlige når programmet startet opp før og fjernet seg ikke før det ble lagd en ny modell.

Aktuelle klasser: MultisampleGLControl, FrmMainGui

Uke 6

04/02-2013

Lagt til funksjonalitet til Print-knappen, og snarvei til CTRL+P for å printe. Den lager en bitmap av modellvinduet og så printer det i landskapsmodus som fyller et A4-ark. Må jobbe litt mer på denne senere, den «kapper» modellvinduet litt slik at hele modellvinduet ikke blir med på arket. Og lagt til flere fil-formater på Export-knappen. Kan nå eksportere til fire formater; jpg, png, bmp og gif.

La merke til en ny bug, men dette ligger i en metode som allerede skal endres på. Det er der man deler inn en member inn i flere members. Hvis man legger til en punktlast på en member blir denne delt inn i 2 submembers, men hvis man går inn i egenskapene til denne og deler den inn i members så forsvinner lasta. Selve lasta blir ikke borte, kun tegningen av den. Og hvis man da forsøker å tegne opp flere laster blir heller ikke disse tegnet opp.

Maintainability Index er ganske lav på enkelte klasse. Spesielt på de store. En måte jeg skal øke denne på er å endre måten noen metoder håndterer retur-verdien.

```
Private int ReturnNumber(int numberOne, int numberTwo)
{
    If(numberOne > numberTwo)
        Return numberOne;

    Else
        Return numberTwo;
}
```

Dette er en metode som tar inn to tall og returnerer det største av de. En annen måte å skrive denne på, som øker maintainability index på er slik. Innføres slik skrivemåte på alle metoder vil dette øke maintainability index betraktelig.


```

Private int ReturnNumber(int numberOne, int numberTwo)
{
    Int numberToReturn;

    If(numberOne > numberTwo)
        numberToReturn=numberOne;

    else
        numberToReturn=numberTwo;

    return numberToReturn;
}

```

Aktuelle klasser: FrmMainGui

05/02-2013

Møte idag I 3 timer. Der kom vi til enighet om å slette all kode som ikke er i bruk. Det er en del klasser og metoder som ikke er i bruk noen plasser. Det er like greit å slette det, det kutter ned på kodelinjer, frigjør minne og det blir mindre forvirrende for fremtidige utviklere. Har kikket på advarselene som Microsoft kommer med når man kjører en Code Analysis på prosjektet. Kjører nå på Analysis Settings som Microsoft anbefaler, og skal prøve å fikse på disse. Det går mye på at variabler skal gjøres private og heller kun ha aksess gjennom Properties. Og en del advarsler rundt dette med importering av fortran-dll.

Endrer litt på print ogekspport-metodene. Siden de bruker en del av de samme koden for å hente et bilde av modellen, så har jeg lagt dette i en egen metode. Så bruker print og eksport-metodene denne metoden når de skal hente et bilde av modellen.

Aktuelle klasser: FrmMainGui

06/02-2013

Startet på et lite prosjekt idag. Hver gang programmet får en exception skal denne skrives ned. Hva slags type exception det er, hvilken klasse den oppstod i, hvilken metode/event/property den oppstod i og grunnen til at den kom. Har bedt de to andre sende meg sine også. Så skal jeg gå gjennom alle sammen om noen uker og forsøke og fjerne de.

Har undersøkt nærmere en Property i Joint-klassen om heter Position. Den blir kalt hele tiden etter at joints har blitt lagd i en modell, og det skal ikke skje. Den blir kalt fra metoden openGlControl_MouseMove, som er en event som tar for seg hva som skjer når muspekeren beveger på seg inne i modellvinduet. Dette eventet kaller metoden RePaintGlForm() som tegner alt opp igjen. Det betyr at hele modellen tegnes på nytt hele tiden når muspekeren beveger seg. Slettet denne og la til kallet der den trengs, f.eks når man tegner opp en ny member eller skal merke flere ting med en firkant.

Aktuelle klasser: GlobalSettings, FrmMainGui, FrmMemberProperties, FrmModel

07/02-2013

Da er inndelingen av en member inn i nye members fikset. Feilen var en enkel programmeringsfeil, men en feil som tok veldig lang tid å finne ut av. Metoden som finner ut av hvor de nye jointene skal stå bruker en metode fra Frame2D som heter Divide(). Det ble sendt inn en variabel som ikke var definert inn i denne metoden, og da returnerte den ingenting. Det var høyden til en member som ikke var sendt inn. La til en sjekk om det er en rett member som skal deles inn, og da sendes det inn høyde lik 0. Det var 2 kodelinjer som skulle til, og timesvis med debugging.

Har fikset på F1-knappen. Tidligere så åpnet det en boks hvor det stod «Display help», nå åpner F1-knappen brukermanualen. Har og satt grenser for hvor mange FPS det kan være på animasjonen til en Free Vibration analyse. Satte den nedre grensa til 2 og den øvre til 100. Det er kanskje litt ekstremt det og, men det kan alltid justeres senere.

Har flyttet på en del kode. I stedet for å opprette nye events som går direkte på forms, slik som create(), resize(), close(), mouse_click() osv så er det bedre å ha de i override-events. Da benyttes funksjonalitet som allerede er der, istedenfor å opprette nye events. Det er bedre for minne og vil gjøre alt litt kjappere.

Aktuelle klasser: FrmModel, Member, CurveUtils, FrmMemberProperties, FrmMainGui, GlobalSettings, ModelSymbol, OpenGLUtils

08/02-2013

Gikk gjennom konstruktøren og sjekka hvor lang tid hvert element tok å laste inn, for innlastingen av programmet tar lang tid. De to verste synderne er LoadCustomCursors() og InitSkinGallery(). På min pc brukte de 333 millisekunder og 636 millisekunder. Alle var enige om at dette tok altfor lang tid, programmet må starte raskere. Så jeg har implementert noen backgroundWorkers. Siden det blir startet opp en ny tråd for hver av disse som da jobber i bakgrunnen tar de 1 millisekund hver i konstruktøren. Det hjalp på veldig mye. Det var faktisk så effektivt at jeg skal gå gjennom flere klasser og undersøke muligheten for å legge inn noe som en backgroundworker, siden det gir et inntrykk av at programmet er mye raskere. Når man nå hopper mellom modeller eller åpner/lager en ny så starter man på ribbon-siden Modelling.

Aktuelle klasser: FrmMainGui, FrmModel, CustomCursorList

Uke 7

11/02-2013

Har satt meg inn i tilleggspakken Installshield. Det er en veldig god måte å lage installasjoner til Windows på med mye funksjonalitet. Den fulle versjonen koster en del kroner så må bruke gratisversjonen. Men med dette går det enda an å lage en veldig fin installasjon for fap2D. Har lagd en en som legger seg i c:\Program Files(x86)\NTNU\Fram Analysis Program\. Med snarveier på skrivebordet og i start-menyen.

Har kikket på metoden Move() i Joint-klassen. Den blir kalt hver gang et joint flyttes i modellvinduet. Hvis man høyreklikker på et internal joint og skriver inn ny x-verdi så flytter den seg feil. Hvis man derimot skriver inn ny x-verdi i left panel så flytter den seg korrekt. Men etter å ha gjort det så flytter ikke det internal jointet seg etter en host member når den flytter seg, og det skal den gjøre. Så begge måtene flytter internal joints feil. I Move() så beregnes det en ny relativ distanse for et internal joint

og den får denne distansen fra `GetClosestAllowedMove()` som ligger i `Member`-klassen. Denne tar inn 4 verdier; `bool xDefined`, `bool zDefined`, `Joint MovingPoint`, `Point2D newPosition`. Det ble sendt inn feil verdier til denne metoden før, men det er nå blitt rettet opp.

Når man så på animasjonene til `Free Vibration` så er det fine animasjoner. Men hvis man pauset animasjonen og startet opp igjen så virket det som om animasjonen «bremset» opp. Det er satt en `maximum skaleringsfaktor` som beregnes på nytt hvis man pauser og starter en animasjon. Denne synker stadig hvis man pauser og starter flere ganger. Har satt denne skaleringsfaktoren til å være konstant. Har og slette flere metoder som ikke er i bruk.

Aktuelle klasser: `FrmMainGui`, `FrmJointProperties`, `Joint`, `UC_Joint`, `FrmModel`, `Point2D`, `Point3D`, `Utilities`

12/02-2013

Møte idag. Ble enige om å fjerne muligheten til å lage flere joints oppå hverandre. Det skal enda være mulighet til å kunne flytte et joint oppå et annet, men da skal brukeren få beskjed om at det er flere joints på samme plass ved å tegne de røde. Jeg la derfor inn en metode i `StructuralModel` som sjekker om et punkt inneholder et joint, og returnerer da `true` hvis det finnes joint på punktet. Denne metoden blir nå brukt i `FrmModel` for å undersøke om det er et joint på det punktet man prøver å lage et nytt joint, og da får man ikke lov til å lage det jointet på det punktet. Og det er heller ikke mulig å lage en member som går fra og til samme punkt/joint, for det var mulig før. Dvs at en member gikk i en u-sving fra og til samme plassering.

Aktuelle klasser: `FrmModel`, `StructuralModel`, `Joint`, `GlobalSettings`

13/02-2013

Har kikket på mulighetene for å gi brukeren beskjed om at det er flere joints på samme plassering. Løsningen ble å lage en `bool`-verdi som heter `OnSamePlace`. I tegnetoden til `StructuralModel` så sjekkes det om det er flere joints på samme plass, og gir da jointet `bool`-verdien `OnSamePlace` satt til `true`. Og vil da bli tegnet rødt.

Har og kikket på muligheter for tooltips til joints og members. Det er implementert fra før av en grei metode i klassen `FrmModel.InfoText` som lager en tooltip og så viser det. Måtte bare forbedre den litt og så implementere denne i metoden `MouseHover` i `FrmModel`. Så nå kommer det opp informasjon om joint og member hvis man holder muspekeren over i noen sekunder. Må høre med Kolbein hva slags informasjon som skal vises.

Det er og fra før av lagd en metode som gir brukeren mulighet til å bevege modellvinduet rundt. Men man er nødt til å holde inne midtre musknapp for å ha mulighet til dette. Så jeg skal lage en egen knapp for dette, så slipper man å ha en ekstern mus for å kunne bruke denne metoden.

Aktuelle klasser: `Joint`, `StructuralModel`, `GlobalSettings`, `FrmModel`, `FrmMainGui`

14/02-2013

Lagt til en knapp i toolbox med bilde av en hand. Når denne bli trykt inn så får man en hand som muspeker og man kan bruke venstre musknapp(holde den inne) til å bevege seg rundt i modellvinduet.

Aktuelle klasser: `Mouse`, `FrmMainGui`, `CustomCursor`, `CustomCursorList`, `FrmModel`

Uke 8

18/02-2013 & 19/02-2013

Har sittet med installshield i 2 dager nå og prøvd og fått til en bra installasjon. Savner en del av funksjonaliteten til fullversjonen, men gratisversjonen gjør nytte av seg den og. Nå sjekker installasjonen om pcen har .Net 4.5, og er det ikke installert så blir det lastet ned og installert. En ting jeg ikke har fått til enda er at Microsoft Office Access Database Runtime blir installert om det ikke allerede finnes på pcen. Det er nødvendig å ha dette installert, hvis man ikke har Microsoft Office installert. Har sliti med å få et program som fungerer på Kolbein sin pc. Jeg har testet det på 3 pcer hjemme og fungerer fint på alle 3. Og Frans prøvde på sin pc, og det fungerte der også. Så det er et lite mysterium.

Uke 9

25/02-2013

For å forstå hvorfor programmet ikke fungerer på pcene til Kolbein har jeg formatert en gammel pc jeg har liggende, for så å installere windows på nytt og se hva som kreves for at fap2D skal kunne kjøre. Å formatere en gammel pc tar sin tid, men det er den eneste måten å finne ut av ting. Og det ga resultater, for jeg fant ut at TaoFrameWork må være installert. Dette er litt merkelig, da det ikke var en nødvendighet på den gamle versjonen. Foreløpig så få installasjonen komme med TaoFrameWork og må bli installert sammen med fap2D.

26/02-2013

Har prøvd noen forskjellige løsninger på å installere Tao samtidig som man installerer fap2D. Det enkleste hadde vært og hatt fullversjonen til Installshield, da man kan legge tao som en temporary fil på pcen, installere denne og så slette den. En midlertidig løsning nå er at programmet går inn i registeret ved oppstart og sjekker om nøkkelen TaoFramework ligger under CurrentUser\Software. Gjør den ikke det så kjører fap2D opp installasjonen til Tao og installerer det før det starter opp. Det krever at tao.exe må ligge sammen med resten av programmet under installasjonen og installasjonsfila er blitt mye større. Tror nok anskaffelsen av fullversjonen av Installshield er veien å gå. Men denne løsningen få stå inntil videre, og så se hva som skjer om noen måneder.

27/02-2013

Har sett på dette med inndelingen av elementer på members og submembers. Det er en del rart som skjer her. Har man f.eks en member med 2 submemers og går inn på Manage Meshing og sier at antall elementer skal multipliseres med et tall så oppdateres kun den ene submember. Går man inn på Properties til den ene submember og endrer elementinndelingen så endres ikke tilhørende hostmember sin elementinndeling. Programmet mesher nå med tanke på submemers, så dette bør fungere optimalt. Har man en member delt inn i 10 submembers med 5 elementer hver, og ganger antall elementer med 100 så endres ikke den ene submember. Under resultater ser det da ut som om det er en bjelke med 4505 elementer, ikke 5000.

Når man nå går inn på Manage Meshing og skriver inn en verdi som alle members skal multipliseres med går den gjennom alle submembers og multipliserer opp. Har testet på noen eksempler og det virker lovende.

Har og forandret på Unit-klassen, alle float-verdier er blitt til double-verdier. Det innebærer mye forandringer i koden, men tror det skal være på stell nå. Fikk et par NullReferenceException siden den forandre 0,001 til 0 et par plasser, men det er ordnet opp i.

Aktuelle klasser: FrmMainGui, GlobalSettings, FrmMemberProperties, FrmManageMeshing, FrmModel, FrmPointLoadDialog, Unit, Units, ComputationalModel, FrmArchedMemberDialog, StoringUtilities

28/02-2013

Har startet jobben med å implementere slik at ComputationalModel følger FrmModel, istedenfor FrmMainGui. Per dags dato er det slik at det kun er mulig å ha ett resultat oppe av gangen. Siden det kun er en instans av klassen FrmMainGui, så finnes det og kun en instans av ComputationalModel. Hvis det isteden følger FrmModel er det mulig å ha ett resultat per modell. Har foreløpig kun gjort dette med Linear Static og det fungerer ganske bra. Har og lagt inn noen tester. Hvis man har oppe en modell som ikke har noen resultater så er ikke Results-fanen synlig, den blir først synlig når man har noen resultater å vise frem. Hver gang en ny modell er lagd går den direkte til Modelling-fanen, uansett hvilken fane man befant seg i før. Samme skjer hvis man åpner en modell. Et problem før er at hvis man og kikket på noen resultater og byttet modell så var enda resultatene oppe, istedenfor bilde av den modellen man byttet til. Det er nå fikset på.

Aktuelle klasser: FrmMainGui, FrmModel

29/02-2013

Nå følger resultater modellen. Det er mulig å ha ett resultat per modell. Har og lagt til noen tester når man bytter modellbilde, så er noen resultater satt til «standard» visning. Så hvis en kikker på momentdiagram på en modell og bytter så vises «Show All» på modellen man bytter til. Har implementert dette på Linear Static, Non Linear Static, Dynamic Time Domain og Dynamic Frequency Domain. Buckling og Free Vibration har kun en visning så det er ikke noe vits i å gjøre noe med de.

Aktuelle klasser: FrmMainGui, FrmModel, FrmExtremeEarthquake, FrmNationalParameterDialog, FrmUtilizationDialog, FrmSectionForces, FrmXtrmDialog

Uke 10

04/03-2013

Tooltips for joints og members fungerer ikke nå i Results-fanen, har lagt inn test som sørger for dette. Har brukt en stund på å finne ut hvor mange ganger alt blir tegnet om igjen. Det viser seg at når man trykker med musa blir alt tegnet om igjen 2 ganger. Det er fordi C# kaller metoden Mouse_Leave når musa blir trykt ned, og der blir alt tegnet om igjen. Denne er overflødig og er slettet. Et annet problem er at det blir kalt en RePaint mange ganger når det blir lagd en ny modell. Den blir og kalt en del ganger når man bytter modellvindu. Hvis man lager en ny modell og har en åpen modell så vil den åpne modellen bli tegnet 6 ganger. Samme skjer når man bytter modellvindu, den man bytter fra vil bli tegnet om igjen 6 ganger. Synderen her er metoden InitAnalysisPage() i FrmMainGui. Den kaller en repaint hver gang den kalles. Den er fjernet og antall repaints er gått drastisk ned.

Aktuelle klasser: FrmModel, FrmMainGui,

05/03-2013

Har sett nærmere på diverse tegnetoder og de metodene som heter FindObjects() og GetDrawableItems, alle de metodene som finner objekter i modellvinduet. Det er en del av de som har gjenbrukbar kode og som kan samles på en plass istedenfor å ha masse metoder som har nesten lik innmat.

06/03-2013

Kolbein har lagt ut nytt bilde til å bruke som hand når man beveger seg rundt i modellvinduet. Har endret tegnetoden til Joint litt, siden Kolbein får runde joints på sin pc mens vi andre får firkantede joints. Har og fjernet tooltips for joints, og har fjernet member-id fra tooltip til member. Og gjort slik at det er 2 gjeldende siffer etter komma for lengden til member i tooltip. Tegnetoden til initiell tøyning/temperatur oppført seg rart. En member med initiell tøyning skal få et symbol med epsilon-symbol over seg, men temperatur gir et symbol med en stor T. Legger man først til tøyning, så endrer denne til temperatur blir fortsatt symbolet et epsilon. Og legger man til tøyning, så temperatur, så tøyning osv så forsvinner etter hvert symbolene og ingenting vises. Det er nå endret på, det fungerer nå å legge til både tøyning og temperatur og det vises korrekt. Har og fjernet resize()-metoden til FrmMainGui siden den kaller ResizeGl(), det samme som resize-metoden til FrmModel gjør. Trenger ikke to metoder som gjør akkurat det samme.

Har endret tegnetoden til Joint. Før ble alle joints tegnet opp 2 ganger. Når et joint er oppå et annet joint, eller et joint er trykt på får det en annen farge. Måten dette blir gjort på er å tegne opp det opprinnelige jointet, så et mindre joint, evt med en annen farge. Lagt til en sjekk om det er nødvendig å tegne opp joint nummer to.

Metoden MouseHover i FrmModel blir kalt når muspekeren står stille. Det er her metoden for tooltip blir kalt. Men problemet med dette er at for å finne ut hvilket objekt som er under muspekeren så tegnes alt opp igjen for å sjekke hva som er under muspekeren. Dette tar ganske lang tid, og drar en del ressurser når alt blir tegnet om igjen hele tiden når muspekeren står stille. Med tanke på at MouseHover kun skal brukes for å vise frem tooltip for en member må det finnes en annen måte å gjøre dette på.

Aktuelle klasser: Joint, FrmModel, GlobalSettings, FrmMainGui, FrmInitStrainTemp, FrmTempStrainAllMembers, FrmTempStrainIndividualMembers, UC_CouplingSpring, SubMember, GraphNode

07/03-2013

Har brukt dagen på å lage en ny metode som sjekker om to punkter ligger på en submember. Denne blir brukt på tooltips for members i modellen. Istedenfor å tegne opp alt i modellen går den nå gjennom alle members i modellen for å sjekke om koordinatene til muspekeren tilhører noen av dem. Denne metoden tok sin tid å lage. For rette members var det greit, men for bøyde var det en annen sak. Men det er i orden nå, og en stor forbedring i tid brukt. Den gamle metoden med å tegne opp alt brukte mellom 60 og 80 millisekunder. Den nye bruker 0-1 millisekunder, så en stor forbedring. Og vil forhåpentligvis føre til at programmet føles mer responsivt

Aktuelle klasser: FrmModel, SubMember, CurveUtils

Uke 11

11/03-2013

Da fungerer tooltips for members ordentlig, både på rette og bøyde. Hadde tidligere et problem med å sjekke submembers for bøyde members. Men det er rettet opp og alt med tooltip for members fungerer optimalt. Har og endret tooltip for en del ab knappene i Modelling-fanen, etter at Kolbein ga meg noen ark med tekst som skal stå der. Og endret på dialogvinduet til Local Axis. Gjør man en endring i lokal akse, men trykker på cancel-knappen så går det tilbake til slik det var før man åpner vinduet og gjorde endringer. Lagt til en repaint til popup-klassen, så høyreklikker man på noe og endrer så blir det tegnet opp umiddelbart. Før måtte man trykke med musknappen f.eks for at ting skal bli oppdatert.

Flytting av lineload fungerer virkelig dårlig. Og tegning av support tar lang tid. Har man en modell med mye randbetingelser bruker tegnet metoden veldig lang tid. Dette må gjøre noe med

Aktuelle klasser: FrmMode, Submember, FrmCouplingSpring, FrmEccentricityDialog, FrmJointProperties, FrmMainGui, FrmLocalAxes, FrmSpringDialog, FrmManageMeshing

13/03-2013

Før gikk det an å flytte på objekter i Results-fanen, det er nå fikset. Lagt til en test i starten av metoden MoveStructuralItems(). Har og fikset på flyttingen av lineloads. Dro man den før ut av modellvinduet forsvant den langt oppover. Har og lagt inn en test i MoveStructuralItems() som gjør at man ikke kan dra laster ut av modellvinduet. Hvis man merker flere laster, uansett om det er pointload eller lineload, og høyreklikker på «show»/»edit» så er den lasten man da klikker på som kommer opp i dialogboksen. Og kan kun endre den lasten, ikke alle som er merket. Enten så må alle lastene som er merket kunne bli redigert, ellers så må det være slik at man kun kan merke en last av gangen. Det må være konsekvent.

Aktuelle klasser: ModelSymbol, LineLoad, StructuralModel, FrmReleaseDialog, FrmModel, Load, FrmSpringDialog, FrmModel

14/03-2013 – 17/03-2013

Har sittet nå i et par dager med et ganske stort problem som Erik fant. Lager man en modell, kjører en analyse, ser på resultater, går tilbake og endrer modellen og så går tilbake på resultater -> Krasj. Grunnen er at en ComputationalModel tar inn StructuralModel i konstruktøren sin. Og ComputationalModel og StructuralModel har da de samme pekerne til objektene i modellen. Har derfor brukt lang tid på å lage en Klonemetode i StructuralModel, og det viser seg veldig vrient. Siden alle objektene i modellen har egne pekere kan man ikke bare kopiere pekerne. Er nødt til å kloner absolutt alt i modellen, og det er mye som skal klaffe sammen. Har klart å lage en klonemetode som kloner alle joints, members, submembers og internal joints. Var en del trøbbel med internal joints, siden det er mye som skal koples sammen. Stort problem med det er de dynamiske analysene skal kjøre animasjoner. Siden Reset-knappen kjører analysen på nytt, må det ordnes en test som ikke kloner en tom modell.

Aktuelle klasser: FrmLineLoadDialog, FrmMainGui, StructuralModel, Joint, FrmModel, Member, PointLoad, Submember, FrmCrossectionDialog

Uke 12

18/03-2013

Måtte gi opp prosjektet med å klonе StructuralModel. Det er altfor mange pekere, lister med pekere og mye annet som må stemme overens i den klonede modellen. Det kommer til å ta lang tid med å få det til. Har designet om dialogboksen til egenskaper for members. Nå viser den i tillegg lengde på valgte member/submemer. Denne er ikke mulig å endre på, den viser kun lengde.

Aktuelle klasser: StructuralModel, PointLoad, Member, SubMember, Joint, FrmMemberProperties

19/03-2013

Møte idag. Ble enige om å bruke variabelen isModified til å enable/disable Results-fanen. Setter denne til true hver gang det er en endring i modellen, og når denne er true så er Results-fanen disabled. Da er det umulig å få noe feil hvis man endre modellen for så å gå tilbake til Results-fanen. Denne variabelen brukes også når en modell lukkes, hvis den er satt til true spør programmet om modellen skal lagres.

Har og endret på dialogboksen til Point loads. Høyreklikker man på en point load og klikker på edit så er dialogboksen noe endret, combo boxen er endret til en label som viser aktuell load case. Og det er og muligheter for å klikke på flere point loads og endre de på samme måte som en. Og lagt inn en test som gjør at det er umulig å legge til en point load med verdi 0.

Aktuelle klasser: FrmMainGui, ComputationalModel, FrmPointLoadDialog, FrmModel

20/03-2013

Startet jobbem med isModified idag, den skal settes til true når noe blir lagt til, endret eller slettet i modellen. Og når den blir satt til true så skal tittelen oppdateres med en * sin tilsier at man har en modell om er modifisert og ikke lagret. Metoden UpdateUndoRedoBuffer() gjør disse to tingene i tillegg til å legge ting i undo/redo-bufferet slik at man kan ta en undo/redo. Denne metoden blir nå kalt hver gang man legger til noe eller sletter noe. Og når isModified er satt til true er Results-fanen inaktiv. Enda ikke lagt til denne metoden ved høyreklikk eller når noe flyttes.

Det er en del dialogbokser som tillater negative verdier og verdier lik 0. Har gått gjennom alle dialogboksene hvor dette er tillatt og skal ta opp med Kolbein over hva som skal være lovlige verdier i alle boksene.

Aktuelle klasser: FrmLineLoadDialog, StructuralModel, FrmMainGui, FrmPointMassDialog, FrmPrescribedDisplacementDialog, FrmModel

21/03-2013

Har gått gjennom en del dialogbokser og satt hva som er lov å sette verdier til. Fant og en ny bug, lager man en vertikal member av typen Bar, fast innspent i bunn og kjører en analyse på det så stopper programmet opp,uten advarsel. Debuggeren til VS gir heller ingen informasjon om hvorfor dette skjer.

Aktuelle klasser: ModelSymbol, FrmMaterialDialog, FrmPointMassDialog, FrmPrescribedDisplacementDialog, FrmSpringDialog, FrmPointDamperDialog

22/03-2013

Fikset opp i flere dialogbokser, med hva som er lovlige verdier. Popup-menyen når man høyreklikker på en submember inkluderer også Remove Initial Strain/Temperature, for den var kun inkludert på members før. Har og lagt inn restriksjoner på godkjente verdier til Free Vibration og Buckling analysene.

Aktuelle klasser: FrmPointLoadDialog, FrmMainGui, FrmTempStrainAllMembers, FrmTempStrainIndividualMembers, FrmModel, UndoRedo, FrmImperfection, FrmStiffness, FrmInitStrainTemp

Uke 13

25/03-2013

Da er jobben ferdig med UpdateUndoRedoBuffer, den blir kalt hver gang noe blir lagt til modellen, fjernet, redigert eller flyttet på. Det er slik på alt under Modeling og Loading fanene, med unntak av responsparametre, tidsfunksjoner og jordskjelv. Flytte-metoden var litt vrien å få til, siden MoveStructuralItems() blir kalt kontinuerlig når man flytter på modellen eller deler av modellen. Men fant en måte hvor UpdateUndoRedoBuffer() blir kalt kun første gangen MoveStructuralItems() kjøres. Eneste dialogboksen som ikke er fikset opp med tanke på lovlige verdier er dempningen, nærmere bestemt Rayleigh Damping. Resten har fått regler og advarsler for null-verdier og negaiver verdier. Har og piffet opp dialogboksen til Numerical Integration litt. Teksten i boksen git litt mer informasjon og har implementert en knapp som heter «Reset to default», som setter verdiene lik de vi har satt som default. Nå er det påskeferie, så blir nok ikke gjort noe før jeg er tilbake fra ferien.

Aktuelle klasser: ComputationalModel, FrmPointLoadDialog, FrmResponseParameterDialog, FrmModel, FrmMainGui, FrmNumericIntegration

Uke 14

02/04-2013

Har brukt dagen på å sette meg inn i undo/redo-metodene. Grunnen til at de går så treigt er fordi det blir lagd 2 nye FrmModels, og det er unødvendig. De 2 kallene tar ca et halvt sekund hver, og det er ganske mye. Har fjernet disse og endret litt på metodene, så en undo/redo går nå så kjapt at det ikke er noe flimring mer.

Aktuelle klasser: FrmMainGui, FrmModel, UndoRedo

03/04-2013

Er ferdig med metodene for undo/redo idag. Begge metodene går nå som de skal. Det var et problem med buffer-lista hvor filnavnene ligger i. Tok man en undo tilbake til starten og så startet å modellere ble ikke denne slettet. Så da ble det krøll hvis man tok en undo igjen, da hentet den frem «gamle» modeller. Det er nå rettet opp. Har og fikset dialogboksen til dempning. Satt hvilke verdier under Rayleigh Damping som er lovlig og ikke lovlig.

Aktuelle klasser: FrmDampingDialog, FrmMainGui, FrmModel, UndoRedo

04/04-2013

Har idag fikset dialogboksen til Member properties. Der var det en del rart som skjedde. Hvis man merket flere members så ble number of elements lik antall elementer totalt på de merkede members. Det gikk an å sette material til hva man ville uten å skrive inn noe, som medførte krasj. Og hvis man merket flere members så ble material-lista lasta inn flere ganger. Det var mulig å sette høyde når en bøyd member var merket sammen med noen rette. Bar, strut og cable stod oppført med 40 elementer, de skal kun ha 1. Alt dette er rettet opp i.

Aktuelle klasser: FrmMemberProperties, StructuralItemsCollection, Member

05/04-2013

Nå går det an å redigere flere line loads samtidig. Det er kun lov å redigere intensitene til lastene, ikke noe annet. I egenskapene til en internal joint er det ikke lov å redigere x-verdien hvis den tilhører en vertikal member, og ikke lov å redigere z-verdien hvis den tilhører en horisontal member. Har endret på tegnemetoden til PointMass, den blir tegnet opp som en stor sirkel uansett. Når man høyreklikket på en bar, strut eller cable member i Results-fanen så ble rundingen rundt de ekstremt store siden det kun er ett element i de, det er endret til en betraktelig mindre sirkel. Endret og litt på dialogboksen til response parameters, det er ikke lov å legge inn samme response parameter flere ganger lengre.

Har og startet med å endre på errors. Det er ganske bra bygd opp, men noe av informasjonen som kommer er av og til forvirrende. Spesielt for en bruker som ikke har noe informasjon om tabellene til computationalmodel. De gamle utviklerne har bare kopiert hva som står i dokumentasjonen til Kolbein i dokumentet Frame2D_error.pdf. Og det er en del her som bør skrives om.

Aktuelle klasser: FrmLineLoadDialog, Joint, FrmJointProperties, FrmModel, Element, FrmResponseParameter, UC_Joint, ComputationalModel, Frame2D.Error

Uke 15

08/04-2013

Har laget en ny installasjon til Kolbein. Det var et problem før at man mistet skrive-rettigheter til logg-fila etter en installasjon. Det er fordi Windows automatisk setter den til Read Only etter installasjonen. Jeg har inkludert et eget manifest til prosjektet som krever at en bruker må ha administrasjonsrettigheter for å kjøre programmet, for da får man skrive-rettigheter til loggen. Har og skrivd om metoden Save() til at elementinndelingen til members lagres med modellen. Jeg skal bruke tiden fremover med å lagre alt som ligger i LocalStructuralSettings med modellen. Det er 59 variabler i den klassen, og per dags dato er det kun 18 av de som lagres.

Aktuelle klasser: FrmMainGui, fap2D, ModelDataSet, StoringUtilities, LocalStrucuralSettings

09/04-2013

Møte idag. Har laget en ny tegnemetode for joints. Siden Nvidia sine grafikkort tegner GL_POINTS som rundinger istedenfor firkanter brukes nå GL_POLYGONS og tegner opp fire streker til å lage en firkant. Har og lagt til en ny knapp under applikasjonsmenyen som åpner opp logg-fila.

Aktuelle klasser: Joint, FrmMainGui, GlobalSettings

10/04-2013

Laget en ny installasjon til Kolbein som ikke krever administrasjonsrettigheter, siden Kolbein fikk problemer med pcen han har på jobb. Har og slettet den nye tegnetoden til joints, siden den er sensitiv for zooming. Har skrivd om errors og warnings etter at Kolbein har skrevet om de som var forvirrende. Lagt inn en sperre på zooming på dialogboksen til eksentrisiteter. Finnes det ingen eksentrisiteter så er zoom-knappene inaktive. Har og lagd ny dialogboks til Manage Meshing.

Aktuelle klasser: Frame2D.Error, UC_Eccentricity, UC_Joint, Joint, UC_JointProperties, GlobalSettings, FrmMainGui, FrmSpringDialog, FrmEccentricityDialog, FrmManageMeshing

11/04-2013

Har lagd ferdig dialogboksen til Manage Meshing, med kode som blir utført når man trykker på OK-knappen. Har og modifisert ModelDataSet slik at alt i LocalStructuralSettings blir lagret med modellen. Har og fjernet Import-knappen fra applikasjonsmenyen, siden denne ikke er i bruk.

Aktuelle klasser: FrmManageMeshing, FrmMemberProperties, FrmModel, StoringUtilities, LocalStructuralSettings, ModelDataSet, FrmMainGui

12/04-2013

Har endret litt på ModelDataSet slik at flere ting lagres. Og har lagt inn tooltips for knappene under Loading-fanen.

Aktuelle klasser: StoringUtilities, LocalStructuralSettings, FrmMainGui, ModelDataSet

Uke 16

16/04-2013

Har endret lagringsmetoden til å inkludere imperfections som fins under ikke-lineær statisk analyse. Og har endret lagringsmetoden til cross sections. Hvis man lagret en modell uten cross sections så gikk det ikke an å åpne den igjen. Den funker litt bedre nå. Har og gjort slik at Joint list i left panel blir oppdatert når man gjør en undo/redo. Og internal joints oppdaterer seg i left panel med koordinater når man flytter på tilhørende hostmember. Det er slik at egenvekt blir beregnet og brukt som last under statisk analyse hvis man ikke har noen andre laster. Men den ble før ikke fjernet når man gikk tilbake og la til laster, måtte manuell gå inn i Load Combination sin dialogboks og fjerne den. Den blir nå fjernet hvis lastkombinasjonen har beregnet en egenvekt og så legger til vanlige laster i denne. Det er fortsatt mulig å manuell legge inn egenvekt i lastkombinasjonene.

Aktuelle klasser: FrmLoadCombination, StoringUtilities, LocalStructuralSettings, ModelDataSet, StructuralModel, ComputationalModel, FrmModel, FrmMainGui, LoadCombination, UndoRedo

18/04-2013

Har fikset en del xml-kommentering idag. Det var en del parametre som mangler på flere metoder og konstruktører, og noen som var overfløydige. Og en del parametre som var blitt skrevet feil, og det er blitt skrevet feil i henhold til referanser til andre variabler i kommenteringen. Det var blitt skrevet «<param ref name=» istedenfor «<paramref name=» som er den korrekte måten. Det meste av kommenteringen er ferdig og det eneste som mangler er noen variabler som er satt til public.

Har og lagt inn en regel som gjør at hvis man kjører en buckling analyse som feiler, så blir Results-fanen borte. For gikk man da inn på Results igjen så krasjet det. Og alt under Analysis-fanen blir disabled hvis en lukker alle modellene. Det var mulig før å trykke på Run-knappen blant annet uten noen modeller åpne, og da krasjer det. Det lå og en feil i lukke-koden til programmet, den prøvde å lukke den åpne modellen, men har man ingen åpen modell så krasjet det. Det er og rettet opp.

Aktuelle klasser: FrmMainGui

19/04-2013

Har skrevet om tegnetoden for tekst i programmet. Det var ingen som likte den gamle teksten noe særlig, og nå ser den mye bedre ut. UpdateUndoRedoBuffer() blir og nå kalt når man legger til et joint ved hjelp av left panel. Den var litt vrien, og måtte ty til litt dårlig programmering-etikk. Innførte en counter som blir satt til 0 ved initiering av programmet. Så fyres eventet cellValueChanging, og sjekker denne counteren. Er den 0 blir bufferet oppdatert, er den ikke så skjer ingenting. Og counteren telles opp på slutten av eventet. Etter man har skrivd inn ferdig koordinat i left panel blir eventet cellValueChanged fyrt og denne setter counteren til 0 igjen. Har og endret på tegnetodene til nummerering av joints og members, pga den nye teksten måtte de skrives om litt.

Aktuelle klasser: OpenGLUtils, ComputationalModel, GlobalSettings, FrmMainGui, Joint, Member

20/04-2013

Har gått gjennom alle dialogboksene i fanene Results og Analysis og fjernet Cancel-knappene på alle sammen. Cancel-funksjonen er satt til ESC-knappen, og det var i mange plasser at cancel-knappen ikke hadde noen funksjon. Vi ble enige om at det er bedre å ikke ha en knapp der.

Aktuelle klasser: FrmBucklingDialog, FrmComputationalModel, FrmImperfection, FrmLoadFD, FrmStiffness, FrmLoadHistory, FrmTimeDomain, FrmNationalParameter, FrmReactionForceDialog, FrmDockPanelsToShowDialog, FrmResultsRSpectrumEarthquake, FrmInfluenceLineLoadTrainDialog, FrmRealTimeLoadTrainDialog, FrmInfluenceLineLoadPathDialog, FrmRealTimeLoadPathDialog

Uke 17

22/04-2013

Har skrevet om tegnetoden for punktmasse. Den metoden jeg skrev tidligere var veldig ømfintlig ved zooming. Derfor måtte jeg skrive den om og den fungerer mye bedre. Har og skrivd om tegnetoden til nummereringen av members, de sirklene rundt selve nummeret var veldig små og tallene «krasjet» med sirkelen iblant. Har lagt inn noen restriksjoner på zooming, satt en grense for hvor langt inn og ut man kan zoome før det stopper opp.

Har og skrevet litt om på metoden som lagrer modellen som et bilde, det ble ganske dårlig før hvis man lagde et mindre vindu av programmet og så eksporterte/printet modellen. Har og lagt inn funksjon til print slik at den skalerer modellvinduet før det printes til å passe på et A4-ark. En ting som tar lang tid, det er opptegningen av grid. Det var før definert til å tegne opp hele 7998 linjer hver gang en repaint ble kalt. Jeg har kuttet denne ned ganske drastisk. Og får man en så stor modell at grid ikke dekker hele modellen er det bare å definere om bredde og høyde i toolbox.

Aktuelle klasser: Joint, FrmMainGui, Member, OpenGLUtils, FrmModel, StructuralSettings, Grid

23/04-2013

Endret på fargene til member-ID og internal joints. De blir nå tegnet opp grønt, uten sirkel rundt, og mørk grått. Det er da litt enklere å se forskjell på joints og internal joints. Har og lagt inn en test i left panel sitt event RowCellStyle. Det er en linje der som aksesserer en liste, og iblant forsøkes det å aksessere index nummer -21345 og får da en exception, den er nå fikset. Har sett en del på lisensiering av programmet idag. Det vi ble enige om er å ha en del lisenser liggende på en server på NTNU, så må brukere av programmet være koplet på denne serveren for å kunne bruke programmet.

Aktuelle klasser: Joint, Member, GlobalSettings, FrmMainGui

24/04-2013

Har fikset problemet med at Harmonic loading ikke fungerer hvis det er flere laster i lastkombinasjonen som det skal regnes på. Det ble forsøkt å kopiere en liste over i en annen med forskjellig størrelse, og fikk da en ArgumenOutOfBoundsException. Har og endret PaintId-metoden til members. Ved buede members så ble id tegnet opp likt som ved rette, de blir nå tegnet opp langs buen. Lagt til noen tester i dialogboksen til dynamic time domain. Før kunne man slette laster i en lastkombinasjon etter å ha slettet selve lastkombinasjonen, det er nå rettet opp.

Aktuelle klasser: ComputationalModel, Member, FrmLoadTD, FrmMainGui

25/04-2013

Har fikset på en del dialogbokser som hadde litt merkelig oppførsel. Det er LoadCombination, LoadTD og LoadFD hvor det oppstod en del NullReferenceExceptions, de er fjernet. Lagd ny release til Kolbein idag.

Aktuelle klasser: FrmLoadCombinationDialog, FrmLoadTD, FrmLoadFD, FrmMainGui

Uke 18

30/04-2013

Har lagt inn noen tester i noen dialogbokser som gjør at det ikke er mulig å skrive inn egen tekst hvor det skal velges lastkombinasjon. Gjorde man det før så kom det en exception. Har og endret tegnetoden til submember litt. Hvis «Show member outline» ikke var trykt inn ble ikke member merket gult når man trykte på den, det gjør de nå. På release-versjonen dukket det opp en exception som ikke kommer under debugging, det gjelder redigering av flere line loads. Det er fikset på. Har og lagt inn flere tooltips for knapper under Analysis-fanen.

Aktuelle klasser: FrmMainGui, FrmLineLoadDialog, FrmPointLoadDialog, FrmBucklingDialog, FrmLoadHistory, FrmStiffness

01/05-2013

Sett nærmere på det her med lisensiering. Det er en del som skal inn i koden, og det må testes grundig. Dette er det ikke tid til, for det nærmer seg slutten. Så tok en avgjørelse om at senere utviklere må ta seg av dette.

02/05-2013

Startet med arbeidet på rapporten. Ferdig med første kapittel, innholdsfortegnelsen og startet på andre kapittel.

03/05-2013

Endring av ModelDataSet idag. De tre «show» knappene har egne verdier under StructuralSettings, og de blir nå lagret med modellen. Er «show selected LCmb» trykt inn når man lagrer og lukker modellen er det den som er trykt inn når man åpner den igjen. Har og lagd ny release til Kolbein. Siden det er en del eldre maskiner som sliter med den nye tekst-metoden så ligger det 2 versjoner ute, en med gammel type tekst og en med den nye.

Aktuelle klasser: OpenGLUtils, ComputationalModel, FrmMainGui, StoringUtilities, LocalStructuralSettings, StructuralSettings, ModelDataSet

Uke 19

06/05-2013

Har lagt inn en test når man åpner en fil. Den sjekker om fila allerede er åpnet, og er den det kommer det en beskjed opp om at det ikke går an å åpne samme fila flere ganger. Før var det mulig å åpne samme fila flere ganger. Har og fikset slik at lastkombinasjoner-lista under Loading-fanen kun viser de vanlige lastkombinasjonene, ikke de dynamiske. Har skrivd om tegnetoden til StructuralModel. Det blir tegnet opp i denne rekkefølgen: Supports, Members, Loads, Joints. Symbolene til strain/temp blir først tegnet opp i members, og det blir satt i loads om de skal tegnes opp eller ikke. La derfor inn en ekstra metode som sjekker om disse skal tegnes opp før tegnetoden til members kalles opp slik at strain/temp blir tegnet opp korrekt.

Aktuelle klasser: FrmMainGui, OpenGLUtils, FrmModel, StoringUtilities, ModelDataSet

07/05-2013

Møte idag hvor jeg fikk nye ark med tooltips som skal inn i programmet. Nå er det kun knappene under Results-fanen som ikke har tooltips. Har og fikset på fargene til UC_Stress sin dialogboks, for fargene der var litt feil. Har og fikset opp dialogboksen til Joint forces og, når det er en release tilstede. Over et element er det forskjellig moment på endene. Og hvis man høyreklikket på et joint med release under Resulta-fanen så hende det den viste feil moment, for den skal være lik 0. Den er det, men det ble noen plasser skrivet inn momentet til nærmeste element som hadde verdi.

Aktuelle klasser: Joint, FrmMainGui, FrmModel, UC_Stresses, FrmReactionForceDialog

08/05-2013

Endret litt på FrmLoadCombination, og metoden som lagrer modellen. Når det lagres så legger dynamiske lastkombinasjoner i listen til de vanlige lastkombinasjonene, det er blitt rette opp. Fått skrivd en del på rapporten idag og.

Aktuelle klasser: FrmLoadCombinationDialog, StoringUtilities

09/05-2013

Lagt til en dropdown-list under Dynamic Time Domain, som viser de tilgjengelige dynamiske lastkombinasjonene. Det går an å velge en av de som det skal regnes på, slik som under vanlig statisk analyse. Members som har local imperfection vil bli tegnet opp gult når man er i Non Linear Static i Analysis-fanen, lettere å se hvilke members som har local imperfection på denne måten. Og lagt inn en test til OK-knappen til FrmLoadFD, som gjør at det er mulig å gå ut av dialogboksen ved å trykke på OK selv om man ikke har lagt inn verdier til analysen. Slettet en del metoder i fap2D.Utilities som ikke er i bruk. Skrivd inn en del i rapporten idag

Aktuelle klasser: FrmLoadFD, SubMember, StructuralModel, FrmMainGui, FrmModel, GlobalSettings, FrmLoadTD, DrawingUtils, Camera, SmartBoundingBox, StringEnum, ModelSymbol, ComputationalModel

12/05-2013

Endret litt på tab index i FrmLoadFD, og lagt til 2 variabler som tar vare på hvilken lastkombinasjon som skal regnes med under Frequency Domain.

Aktuelle klasser: FrmLoadFD, FrmMainGui, StructuralModel

Uke 20

13/05-2013

Har lagt inn en egen animert muspeker, den er tenkt brukt når det kjøres en analyse. Har og lagt inn slik at den åpne modellen ikke lukkes når man prøver å lukke hele programmet, men trykker på CANCEL-knappen på påfølgende dialogboks. Skrivd mye på rapporten idag.

Aktuelle klasser: FrmMainGui, CustomCursor, CustomCursorList, Mouse, FrmModel, AutoWaitCursor

14/05-2013

Når det nå kjøres en analyse så blir alle knappene i programmet disabled, og den nye animerte muspekeren vises. Dette for å lettere vise en bruker at programmet er inaktivt mens en analyse kjører. Når analysen er ferdig går ting tilbake til normalt igjen. Skrivd inn mye på rapporten idag.

Aktuelle klasser: FrmMainGui, FrmModel

15/05-2013

Før ble ikke point loads vist når man trykte på Show All under Loading-fanen, det gjør de nå. Har og fikset det slik at zoom to model fungerer skikkelig når man har en modell med u-formede members. BoundingBox hentet feil verdier når det var slike members i modellen. Ny god skrive dag.

Aktuelle klasser: FrmMainGui, StructuralModel, Member

16/05-2013

Lite møte idag med Kolbein hvor han ga meg fem nye ark med tooltips som skal legges inn. Nå er det langhelg med 17.mai og pinse så det får vente til neste uke.

Uke 21

21/05-2013

Lagt inn arkene med tooltips som Kolbein ga meg på torsdag. Har gjort teksten over listen til de dynamiske lastkombinasjonene litt mindre slik at y-en vises. Har og fjernet en variabel som ikke er i bruk i Camera-klassen. Og lagt inn en liten test i dialogboksen for Computational Model under de dynamiske analysene. God skrive dag, tross sykdom.

Aktuelle klasser: FrmMainGui, Member, Camera, FrmComputationalModel

22/05-2013

Vært syk i over en uke nå, så får ikke vært så produktiv som ønsket. Skrivd noen sider i rapporten. Og har lagt inn slik at når man merker flere lineload/pointloads sammen med andre ting så kommer det kun «Delete selected objects» når man høyreklikker på de.

Aktuelle klasser: FrmModel

23/05-2013

Har lagt inn slik at det ikke er mulig å ta en undo/redo i Results-fanen. Det er heller ikke mulig å trykke på ctrl+a for å merke alle tegnbare objekter her og det er heller ikke mulig å slette noen tegnbare objekter. Og har lagt inn et lite bilde under Show geometric imperfections, under resultater for en ikke-lineær statisk analyse. Ellers en fin skrive dag.

Aktuelle klasser: FrmMainGui

24/05-2013-26/05-2013

Disse dagene er blitt brukt til å skrive på rapporten.

Uke 22

27/05-2013

Endelig blir joints tegnet som firkanter. Løsningen ble å skru av multisampling. En enkel løsning, som tok lang tid å finne ut av. Ellers skrevet på rapporten.

Aktuelle klasser: Joint

28/05-2013

Skrevet om noen tooltips som inneholdt noen skrivefeil. Ellers skrevet på rapporten. Har skrevet alt inn, nå starter jobben med redigering.

Aktuelle klasser: FrmMainGui

29/05-2013

Møte idag. Vi ble enige om hva vi skal fremføre på fredag for instituttet. Det blir spennende å vise frem programmet og høre hva folk mener. Har og skrevet om noen tooltips som inneholdt noen skrivefeil. Ellers brukt dagen på å skrive og redigere.

Aktuelle klasser: FrmMainGui

30/05-2013

Lagd en ny installasjon til demonstrasjonen imorra, da er lisensfilen til Kolbein inkludert. Ellers skrivd litt og redigert litt på teksten.

31/05-2013

Demonstrasjon idag for noen på instituttet som gikk veldig bra, det virker som om de var imponert og ville gjerne bruke programmet i undervisningen. Har lagd inn mer tooltip og endret litt på teksten i den ene dialogboksen til jordskjelv. Ellers redigert litt på tekst, og skrivd ferdig forord og sammendrag.

Aktuelle klasser: FrmMainGui, FrmPeakResponseLinRSpectrumEarthquake

01/06-2013

Har lagd et nytt datasett hvor global settings ligger. Har og lagd metoder for å lagre og åpne global settings til en fil som ligger i en egen mappe i Resources. Når en nå går inn på Settings og endrer på global settings så blir det lagret. Og når man igjen åpner programmet så blir de lagrede global settings åpnet. Ellers skrivd litt på rapporten.

Aktuelle klasser: FrmMainGui, FrmSettings, Grid, StoringUtilities, GlobalSettingsDataSet

02/06-2013

Har lagd nye metoder for å finne objekter i tegnevinduet. En metode for å finne joints og members, og en metode for å finne joints og elementer. Metoden som finner joints og members blir brukt til å finne objekter under muspekeren når man klikker med venstre eller høyre mustast i modellvinduet i fanene Modeling, Loading og Analysis. Metoden som finner joints og elementer blir brukt til å finne objekter under muspekeren når man klikker venstre eller høyre mustast i fanen Results. Disse viser seg å være meget effektive. De gamle finne-metodene, som enda brukes for å finne laster, var avhengig av å tegne opp det som man ville finne. Mens de nye metodene itererer over joints, members og elementer, og finner de ved bruk av løkker.

Det viste seg å være verre med å finne laster ved hjelp av løkker. Hver last har en variabel som heter BoundingRectangle, som inneholder rektangelet rundt lasten. Men de gamle utviklerne har skrevet denne feil, og BoundingRectangle leverer feil verdier i flere tilfeller. Derfor må jeg kanskje velge å utsette til senere utviklere, da det ikke er så mye tid igjen. Men siden jeg fikk til å finne nye finne-metoder for joints, members og elementer er tiden det tar for et musklikk drastisk redusert. Jeg sammenlignet en modell som inneholder 56 members. I den gamle versjonen tok et venstre musklikk 158 millisekunder, og høyre musklikk tok 229 millisekunder. I den nye versjonen er disse redusert til 1 og 2 millisekunder for den modellen. På resultatene for den modellen, som da inneholder 2800 elementer, brukte et venstre musklikk 301 millisekunder og høyre musklikk tok 534 millisekunder. I den nye versjonen er disse redusert til 21 og 37 millisekunder.

Metodene som heter FindObjects() ligger også i user controls for joints. Disse tar ikke likelang tid som det gjorde for hele modellen. Men de senere utviklerne bør nok kikke på om disse også kan effektiviseres like bra som FindObjects() er blitt gjort for FrmModel.

Ellers en grei skrive dag. Har redigert det meste av teksten og venter på kommentarer fra Kolbein på noen sider jeg sendte han fra rapporten min.

Aktuelle klasser: Element, Joint, FrmModel

Uke 23

03/06-2013

Har økt begrensningen på zoom ut. Har og skrivd på tooltip til zoom-boksene som sier at hvis zoom ikke fungerer, så må grid-størrelsen økes. Prøvde å gjøre zoom automatisk begrenset til størrelsen på modellen. Men det er noen problemer med det. Er modellen tom så er størrelsen på modellen uendelig, og inneholder modellen kun et joint er størrelsen 0. Det er jo lett å sette en minimum størrelse på modellen, men enda så er det litt dumt. Det er og slik at hvis en bruker vil sette grid stort og lage en diger modell så får en ikke zoomet så langt ut som man kanskje vil, siden modellen er liten eller ikke lagd enda. Da må man tegne en member, zoome ut, forlenge denne member, zoome videre ut osv. Og det blir ganske tungvint. Måten det gjøres på nå, at zoom automatisk blir begrenset i forhold til størrelse på grid, fungerer en god del bedre.

Har sett en del på mulighet for å skrive om metoden for å finne laster. På linjelaster fungerer det greit når lastene ligger på rette members, da kan jeg bruke den innebygde metoden Contains() som ligger i klassen til Rectangle. BoundingBox, som er en property til LineLoad, leverer rektanget rundt lasten. Og når de ligger på rette members er det en enkel sak. Men når linjelastene ligger på arched members blir boundingrectangle veldig stort, siden de følger kurven til member. Prøvde å implementere en metode som finner distanse fra punktet til musklikk og ned til members som lasta lå på, og så sammenligne denne med størrelsen på lasta. Dette fungerer greit, så lenge lasta ligger i ro. Men når den flytter på seg blir det andre saker, og finner ikke en måte å gjøre dette på.

Har og endret på BoundingBox for punktlaster. Det ble returnert feil verdier for alle de forskjellige punktlaster, de returnerer korrekte verdier nå. Denne kan brukes til å undersøke om punktet til et musklikk ligger inne i dette rektanget.

Det er og blitt implementert en test når modeller åpnes. Hvis en flytter eller sletter en modell som ligger i Recent files, og så prøver å åpne denne så kom det før en FileNotFoundException. Denne blir nå testet mot og oppdaget. Prøver noen å åpne en fil i Recent files som ikke eksisterer kommer det en dialogboks opp om at filen ikke eksisterer, og anbefaler å heller bruke Load-knappen under applikasjonsmenyen. Ellers skrivd litt på rapporten.

Aktuelle klasser: FrmMainGui, FrmModel, OpenGLUtils, StoringUtilities, LineLoad, PointLoad

04/06-2013

Har lagt til slik at Units som ligger i fap2D Settings blir lagret sammen med resten av de globale innstillingene. Møte med Kolbein idag hvor han ga meg noen kommentarer på teksten min. Har tatt det til rette og redigert en del tekst idag.

Aktuelle klasser: GlobalSettingsDataSet, StoringUtilities

05/06-2013

Har fjerna en if-setning i opptegning av spenningsdiagram, som sjekker om absolutt-verdiene for minste spenningen og den største spenningen er like med ganske så stor nøyaktighet, og tegner ikke opp noe hvis de er like. Det er nok noe som har ligget der en stund, og den gjorde absolutt ingenting fornuftig. Har og gjort det slik at venstre musklikk i Results ikke gjør noenting. Måtte endre litt på metoden som finner ut om et punkt ligger på et element, den regnet ikke korrekt på elementer som er skrå. Ellers skrivd litt på rapporten litt og redigert litt.

Aktuelle klasser: UC_Stresses, FrmModel, FrmMainGui, Element

06/06-2013

Endret litt på måten joints blir flyttet på under Join Properties, internal joints flyttet seg ikke helt korrekt ved vertikale members. Ellers er rapporten ferdig skrevet, og det er kun litt redigering som gjenstår.

Aktuelle klasser: FrmJointProperties

B. Programvare brukt

B.1 Operativsystem

Utviklingen har foregått på Windows 7 Professional 64-bit[15] av undertegnede, og Windows 8 Professional 64-bit[16] av Aasmundsrud og Erstad. Programmet kompiles i 32-bit og fungerer dermed på alle utgavene av Windows Vista[17], Windows 7 og Windows 8.

B.2. Uviklingsmiljø

Visual Studio 2010[18] ble brukt under høsten 2012 til prosjektoppgaven. Men Visual Studio 12[19] ble tilgjengelig i løpet av høsten og har siden vært det foretrukne programmet å utvikle i.

B.3. Tilleggsprogramvare til Visual Studio 2012

- VisualSVN[20] har blitt brukt til å holde koden oppdatert mellom undertegnede, Aasmundsrud og Erstad.
- DevExpress[21] er en tilleggspakke til grensesnittet, som forenkler utviklingen en del. Det ble kjøpt inn en ny versjon våren 2013 og **fap2D** er oppdatert til nyeste versjon av DevExpress.
- Tao Framework[14] er blitt brukt til å aksessere biblioteket til OpenGL.

B.4 Tekstbehandling

Microsoft Office Word[1] er brukt til å skrive ned logg og oppgaven. Microsoft Notepad[11] er benyttet til å skrive lister over gjøremål som skal gjennomføres.

B5. Annet

- Putty[22] er blitt brukt til for å automatisere autentiseringen opp mot NTNU sin server.
- PuttyGen[23] er blitt brukt til å generere nøkler brukt til autentisering.
- Pageant[24] er blitt brukt til å assosiere private nøkler med server.
- TortoiseSVN[25] er blitt brukt for kontrollere felles kode på server.

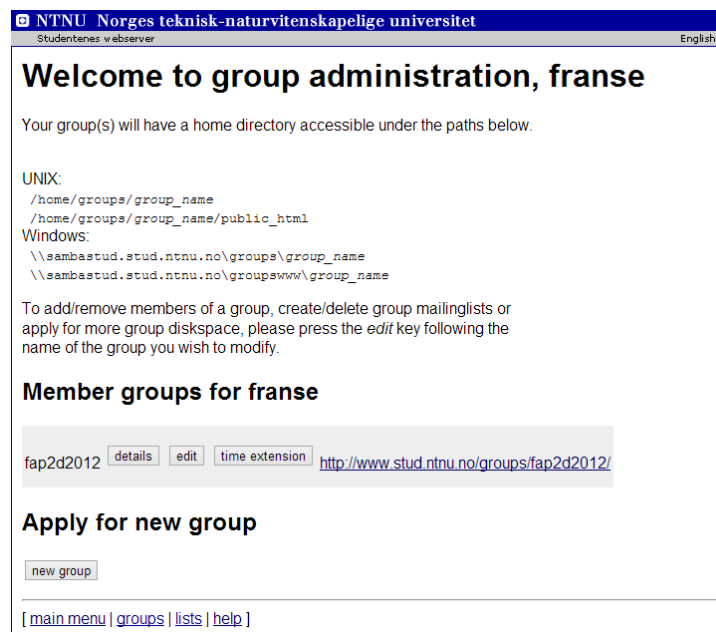
C. SVN

For å dele filer og holde versjonskontroll mellom vi som programmerte i **fap2D** dette semesteret brukte vi en SVN-server på NTNUs gruppeområde. Hver deling som ble gjennomført med serveren hadde en kommentar som beskrev arbeidet som var gjort, i tillegg til en oversikt over filene som hadde blitt endret. For å koble opp mot serveren logget vi inn via Putty[22] og brukte TortoiseSVN[25] for selve opp- og nedlasting av endringer.

For fremtidige utviklere på **fap2D** som vil arbeide samtidig har vi laget en stegvis fremgangsmåte for å lage og administrere en slik server. Orakeltjenesten ved NTNU har vært svært hjelpsomme for vår del når vi skulle gjøre det samme, så hvis man står fast i instruksjonene anbefales det å spørre de.

Fremgangsmåte for oppsett og bruk av SVN-server på NTNUs gruppeområde

1. Lag en gruppe på NTNUs gruppeområde <http://www.stud.ntnu.no/kundesenter/groups/groups.php>. Vi kalte vår gruppe fap2d2012.



Figur 92: Gruppeområdet etter at gruppen har blitt opprettet

2. Bruk kommandovinduet på <http://login.stud.ntnu.no/>, logg inn med vanlig studentpassord og bruk kommandoen «cd /home/groups/fap2d2012» til å navigere frem til gruppeområdet

MindTerm SSH

```
:58)
NTNU IT skal gjennomføre rutinemessig oppdatering av
sentrale IT-systemer onsdag 2013- 05-22 fra kl 16:00 og utover
kvelden. I forbindelse med oppdateringene vil berørte servere bli
restartet ved behov. Det må forventes nedetid eller nedsatt
ytelse på våre tjenester, inkludert, men ikke begrenset til:
Epost FS Websystem som: - imside - www.ntnu.no - org.ntnu.no -
folk.ntnu.no IT's Learning Årkonomisystem som: - Corporater -
Discoverer - Rapprdr Innlogging for både ansatte og studenter Stud-fil
Ansatt-fil Kjernen Syllabeb Lydia Nedetid for hver enkelt tjeneste
vil normalt ikke overstige en time. Varsling av denne type
nedetid foregår normalt på driftvarslings-lista vår. Den er
tilgjengelig på
https://lists.ansatt.ntnu.no/itea.ntnu.no/info/dr iftsvarsling og anbefales
dersom du er interessert i
generelle varsel fra IT-avdelingen. Ved evt. spørsmål ta kontakt med
Orakeltjenesten (orakel@ntnu.no eller tlf. 91500).

Dagens tips: Vår smill med de andre brukerne, ikke kjør maskinen i senk.
Kjør tunge programmer med 'nice' foran, som f.eks. 'nice ./cruncher.py'
Last login: Sun May 26 11:37:25 2013 from dhcp-051134.wlan.ntnu.no

carascal:~$ cd /home/groups/fep2d2012
carascal:/home/groups/fep2d2012
```

Figur 93 - Navigering til gruppeområdet på login.stud.ntnu.no

3. Bruk kommandoen «svnadmin create /rep» til å lage et *repository* på gruppeområdet kalt rep
4. Det vil også være nødvendig å åpne gruppeområdet for alle tilgangsrettigheter med kommandoen «chmod 777»
5. Serveren er nå opprettet. Likevel vil serveren kreve passordgodkjenning for hver eneste fil som lastes opp, dette er tidkrevende og upraktisk. Derfor er det nødvendig å bruke Putty[22] for å automatisk identifisere deg for hver forespørsel fra serveren.
6. Last ned og installer Putty-installasjonspakken fra <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. Bruk Windows installer-versjonen for å sikre at alle nødvendige komponenter i Putty-pakken kommuniserer skikkelig med hverandre.

Binaries

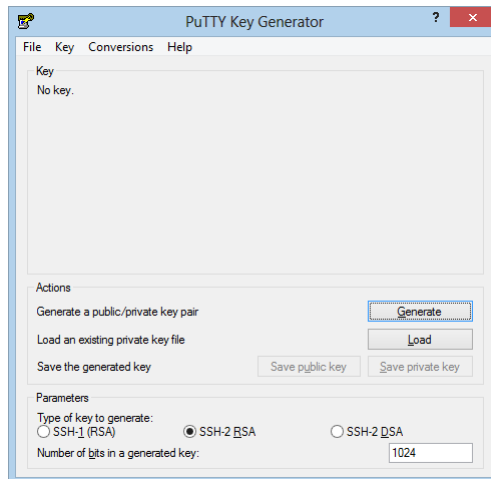
The latest release version (beta 0.62). This will generally be a version I think is reasonably likely to work well.

For Windows on Intel x86

PuTTY:	putty.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYtel:	puttytel.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSCP:	pscp.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSFTP:	psftp.exe	(or by FTP)	(RSA sig)	(DSA sig)
Plink:	plink.exe	(or by FTP)	(RSA sig)	(DSA sig)
Pageant:	pageant.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYgen:	puttygen.exe	(or by FTP)	(RSA sig)	(DSA sig)
A .ZIP file containing all the binaries (except PuTTYtel), and also the help files				
Zip file:	putty.zip	(or by FTP)	(RSA sig)	(DSA sig)
A Windows installer for everything except PuTTYtel				
Installer:	putty-0.62-installer.exe	(or by FTP)	(RSA sig)	(DSA sig)
Checksums for all the above files				
MD5:	md5sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-1:	sha1sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-256:	sha256sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-512:	sha512sums	(or by FTP)	(RSA sig)	(DSA sig)

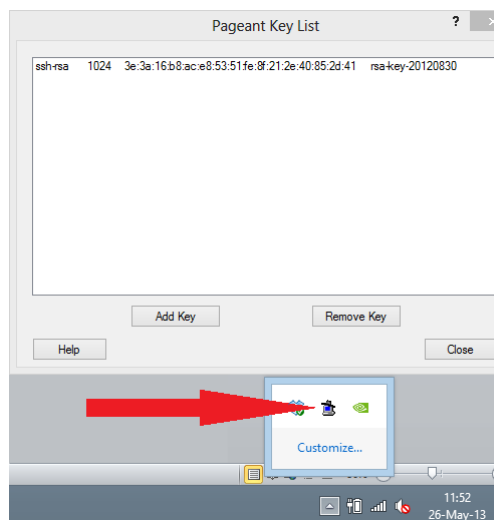
Figur 94: Hvilken Putty-pakke som skal velges på nettsiden for nedlasting

7. Bruk Puttygen[23] fra Putty-pakken til å lage et sett med Private- og Public key. Det er ikke nødvendig å lage passord for nøklene.



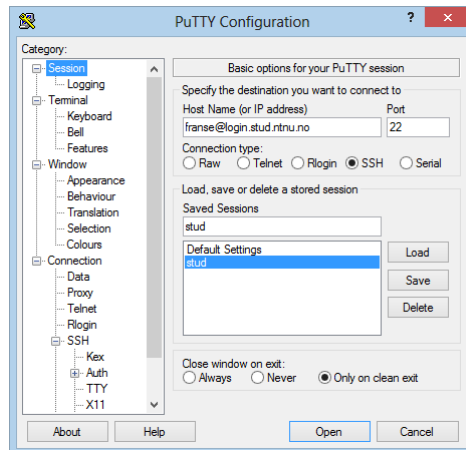
Figur 95: Puttygen som man bruker for å lage nøkler

8. Assosier filen Private-key med programmet Pageant[24] fra Putty-pakken, ved å velge Pageant som standardprogram for Private-key-filen. Sett din datamaskin til å åpne Private-key ved oppstart slik at den alltid vil være aktiv i Pageant i bakgrunnen mens datamaskinen er på.
9. Pageant burde alltid være aktiv i bakgrunnen med Private key aktivert.



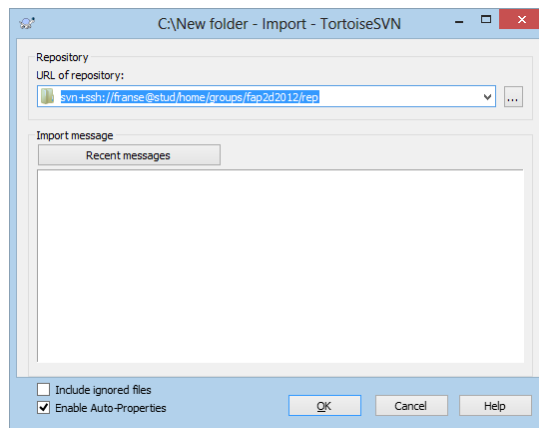
Figur 96: Pageant vinduet åpent med private key aktiv og pageant aktiv i bakgrunnen

10. Bruk kommandovinduet på <http://login.stud.ntnu.no/> til å navigere til gruppeområdet med kommandoen «cd /home/groups/fap2d2012»
11. Bruk kommandoen «vim .ssh/authorized_keys» for å åpne tekstbehandleren vim og legge til innholdet Public key i filen .ssh/authorized_keys. I .ssh/authorized_keys skal det stå: «svn+ssh: (hele innholdet i Public key uten linjeskift eller mellomrom)».
12. Lagre og lukk .ssh/authorized_keys med vim-kommandoen «:wq».



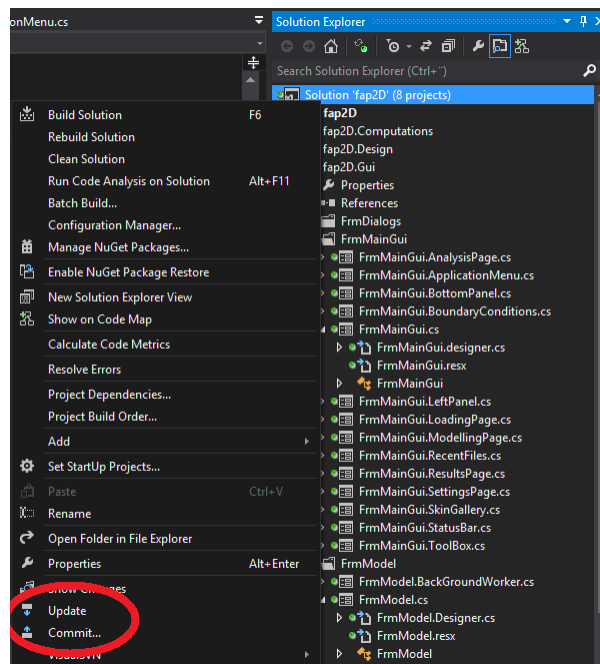
Figur 99: Valg av «session» innstillinger i Putty

15. Putty[22] er da ferdig satt opp og serveren på grupperområdet vil da godkjenne datamaskinen du har privatnøkkelen på som din ID. Man slipper å gjenta passord for pålogging og for å laste opp og ned filer til gruppeområdet. Opplegget fungerer hvis man åpner *session* «stud» og får logget på uten å måtte skrive inn passord.
16. Deretter er det nødvendig å lage et område på din datamaskin hvor man kan lagre filer som deles med serveren og som kan være utviklingsområde for **fap2D**.
17. Last ned og installer TortoiseSVN[25] <http://tortoisesvn.net/downloads.html>
18. Lag en tom mappe på datamaskinen der du vil ha utviklingsområdet. Etter installasjonen er Tortoise integrert i operativsystemet og man finner dermed TortoiseSVN[25] ved å høyreklikke på mappen, velge Tortoise og «import». Linken til repository er «svn+ssh://dittBrukernavn@stud/home/groups/fap2d2012/rep», der «stud» er navnet på den *session* som man lagret i Putty[21] i steg 14.



Figur 100: Dialogboks for importering av repository

19. Innholdet på serveren kan da hentes ned i mappen med høyreklikk (*update*) eller (*commit*) for å laste opp. Disse kommandoene er også tilgjengelig fra å høyreklikke **fap2D**-prosjektet som er lastet inn i Visual Studio 2012[19]. Visual Studio kan man få gratis hos Gurutjenesten ved NTNU.



Figur 101: Update og commit knappene i Visual Studio 2012

20. Ved å høyreklikke mappen for utviklingsområde på datamaskinen kan man velge hvilke filer som skal deles eller ikke med «TortiseSVN → add to ignore list». Man burde ikke dele filer som blir autogenerated av Visual Studio, da disse lager kluss med andre brukeres versjoner på serveren. Eksempler på slike er *debug* og *release* versjoner av programmet. Med dette er SVN-serveren og oppkoblingen ferdig satt opp.

Tips og triks:

Vi har hatt noen problemer i løpet av prosjekt- og masteroppgaven som vi gjerne vil at fremtidige utviklere skal unngå. Derfor har vi laget denne listen over tips og triks:

1. Bli enige om et forum hvor man sier ifra når man tar en commit til server. Vi har brukt Facebook[8]. Hvis man tar en *commit* skriver man «c» til de andre i gruppen. Dette har ført til at vi har unngått problemer når flere har jobbet med samme filer. En annen god regel er å ta en *update* fra server og teste om programmet fungerer som påtenkt før man tar en *commit*, dermed slipper man at alle andre får problemer. Forumet er også nyttig verktøy for å diskutere arbeidet som blir lastet opp.
2. Legg til databasen og obj- og bin mappene i *ignorelist*. Databasen er en oversikt over materialene og tverrsnittene som blir benyttet, og vi har flere ganger opplevd at modeller har sluttet å fungere når materialene eller tverrsnittene blir slettet grunnet at en ny database foreligger. Obj- og bin mappene består av kompilerte filer som er unike for hver datamaskin og er ikke nødvendig å laste opp. Disse er finnes i mappen til startprosjektet kalt **fap2D** og blir kompilert dit.
3. Når man commiter skriver man navnet sitt /implementasjonsnavnet og deretter hva *commiten* gjør i kommentarfeltet. Eksempel: «Earthquake – Response spectrum analysis finished».

D. Kodekonvensjoner

Navnekonvensjoner:

Navnekonvensjon	Bruksområde
camelCasing	Brukes for å navngi variabler som ikke er private
_camelCasing	Understrek brukes til å navngi lokale og private variabler
KONSTANT	Konstante verdier navngis med kun store bokstaver
PascalCasing	Pascal casing brukes for alle klasser, typer, metoder og navnerom, samt for navn som består av flere ord
PrePascalCasing	Valgfri prefiks brukes for å indikere at klasser arver fra en klasse i standard biblioteket. Prefikset angir hvilken klasse de arver fra

Tabell 9 - Navnekonvensjoner

Kommentering

Alle metoder og events skal kommenteres. Eksempel:

```
/// <summary>
/// Implements the operator +.
/// </summary>
```

Det skal deretter komme en liste over eventuelle parameternavnene som benyttes i metoden / eventet og hva som returneres. Eksempel:

```
/// <param name="lv1">The first operand.</param>
/// <param name="lv2">The second operand.</param>
/// <returns>The result of the operator.</returns>
```

Alle klasser skal kommenteres. Dette gjøres på lik måte som kommenteringen av et event eller metode. Eksempel:

```
/// <summary>
/// The settings form class.
/// Inherits from XtraForm.
///
/// Note:
/// Text
/// </summary>
```

Enums og Structs skal kommenteres, inkludert alle alternativene. Eksempel:

```
/// <summary>
/// The type of cursor
/// </summary>
public enum CursorType
{
    /// <summary>
    /// A beam member cursor.
    /// </summary>
    BeamMember,
    /// <summary>
    /// A bar member cursor.
    /// </summary>
    BarMember,
}
```

Det skal etterstrebtes å kommentere koden i den grad det er nødvendig. Kommentering av selve koden skal foregå over koden, uten linjeskift mellom. Eksempel:

```
// Add mode shape displacement to nodes
AddModeShapeDisplacement();
```

Dersom det er nødvendig å kommentere variable gjøres dette etter variabelnavnet, for eksempel:

```
private ToolTip _toolTip; // The tooltip
```

Getter og settere skal kommenteres på følgende måte:

```
/// <summary>
/// Gets or sets the structural model.
/// </summary>
/// <value>The structural model.</value>
```

Dersom getter og setteren har parametere skal parameteren kommenteres på lik måte som i en metode / event. Eksempel:

```
/// <summary>
/// XXX
/// </summary>
/// <param name="i">XXX</param>
/// <value>XXX</value>
public double this[int i]
{
    get { return _matrix[i]; }
    set { _matrix[i] = value; }
}
```

Annet:

- Punktum etter param beskrivelsen, på slutt av summary, på slutt av return og på slutt av value.
- Det skal forekomme et mellomrom mellom `///` og beskrivelsen

Regioner

Koden skal deles opp i regioner. Dette er for å få en mer oversiktlig kode. Regionene som koden skal deles opp i, og rekkefølgen, er:

- README
- Enums
- Structs
- Fortran import
- .dll import
- Names
- Variables
- Constructors
- Methods
 - Private methods (kun hvis andre eksisterer)
 - Internal methods (kun hvis andre eksisterer)
 - Protected methods (kun hvis andre eksisterer)
 - Virtual methods (kun hvis andre eksisterer)
 - Public methods (kun hvis andre eksisterer)
 - Analysis (en region per analyse)
 - Other (kun hvis Analysis eksisterer)
- Events
 - Analysis
 - En region per analyse
 - Other (kun hvis Analysis eksisterer)
- Properties
- Interfaces
- Unsupported Properties/Methods etc.
- Abstract Methods

Det skal ikke forekomme tomme regioner. Regioner skal ikke deles inn i subregioner med mindre dette er hensiktsmessig, eller hovedregionen overgår 200 linjer.

Annet:

- Det skal forekomme et mellomrom mellom `#` og region / endregion

Linjeskift

Det skal etterstrebtes å ikke ha for mange linjeskift i koden. Dette er for å få koden til å bli så kompakt som mulig, uten at koden blir uoversiktlig.

Det skal forekomme et linjeskift helt på slutten av en klasse. Eksempel:

```
    }
    #endregion
}
```

Hver if-setning skal ha et linjeskift over og under seg. Eksempel:

```
int numFrames = (int)e.Argument / 2;

if (counter == 0)
    maxScaleFactor =
Math.Abs(ComputationalModel.ResultsScaleFactor);

double scaleFactor = ComputationalModel.ResultsScaleFactor;
```

Det skal ikke forekomme linjeskift mellom if og else if / else setninger. Eksempel:

```
if (distance1 <= distance2)
{
    snapX = Grid.GridLineDistanceX * atXGridLineNo;
    xFound = true;
}
else
{
    snapX = Grid.GridLineDistanceX * (atXGridLineNo + 1);
    xFound = true;
}
```

Det skal ikke forekomme et linjeskift over en if-setning hvis den er på innrykk. Eksempel:

```
{
    if (scaleFactor > maxScaleFactor)
    {
        scaleFactor = maxScaleFactor;
        increment *= -1;
    }
}
```


Dersom if-setningen skrives på 2 linjer skal det forekomme et linjeskift mellom if og else setningen. Eksempel:

```
if (_drawableObjectClicked is Element)
    _drawableObjectClicked.IsSelected = false;

else
    _drawableObjectClicked.IsSelected = true;
```

Hver foreach og for-løkke skal ha linjeskift over og under seg. Eksempel:

```
bool toBeDeleted = false;

foreach (SubMember sm in m.SubMembers)
    if (sm.IsSelected)
        toBeDeleted = true;

if (toBeDeleted && !membersToDelete.Contains(m))
    membersToDelete.Add(m);
```

Return skal ha linjeskift over seg. Eksempel:

```
private bool OpenConnection()
{
    _connection = new OleDbConnection(_connectionString);
    _connection.Open();

    return true;
}
```

Lukking av dialogbokser, altså this.close();, skal ha et linjeskift over seg. Eksempel:

```
_frmMainGui.CurrentFrmModel.RePaintGLForm();

this.Close();
}
```

Det skal ikke forekomme variabler som rett under et innrykk. Da skal det være et linjeskift over. Eksempel:

```
    scaleFactor += increment;
}

ComputationalModel.ResultsScaleFactor = scaleFactor;
```

Det skal ikke forekomme noen linjeskift nedenfor #region og metoden / eventet etc. som kommer etter. Eksempel:

```
#region Events
/// <summary>
/// Handles the Click event of the btnOk control.
/// </summary>
```

Dette gjelder også linjeskift over #endregion og metoden / eventet som kommer før. Eksempel:

```
        btnOk.Focus();
        base.OnActivated(e);
    }
#endregion
```

Dersom det er en region som starter rett inni regionen skal det forekomme et linjeskift mellom disse. Eksempel:

```
#region Methods

#region Private methods
```

Det samme gjelder på slutten av en region. Eksempel:

```
    }
#endregion

#endregion
```

Properties

Properties skal skrives på følgende måte:

```
public CursorType CursorType
{
    get { return cursorType; }
    set { cursor = value; }
}
```

Tomme properties skrives på følgende måte:

```
public CursorType CursorType { get; set; }
```

Interface

Interface implementeres ved klasseavn : interface. Eksempel:

```
public class JointList : CollectionBase, IBindingList
```

Initialisering av dialogbokser

Ved initialisering skal hoveddialogboksen få fokus, ikke en controller.

Usings

Det skal etterstribes å kutte ned på ubrukte usings. Dette gjøres lettest i Visual Studio ved å høyreklikke i koden → Organise Usings → Remove and Sort. Alle usings blir dermed tatt bort og alle usingsene blir sortert.

Generell opprydning av kode

For å rydde opp generelt i koden kan man trykke på Edit → Advanced → Format document i VS

Innkapsling

Alle variable bør innkapsles via gettere og settere.

Switch

Switch statements skrives på følgende måte:

```
switch
(_parent.CurrentFrmModel.StructuralModel.LocalStructuralSettings.A)
{
    case EigenValueAlgorithm.subspace:
        radioGroupEigenValueAlgorithm.EditValue = 0;
        break;
    case EigenValueAlgorithm.lanczos:
        radioGroupEigenValueAlgorithm.EditValue = 1;
        break;
    default:
        break;
}
```


E. Forklaring av koden

E.1 Prosjekter

Fap2D er bygd opp av 9 prosjekter, delt inn i hva slags formål de har. Det er `fap2D`, `fap2D.Computations`, `fap2D.Design`, `fap2D.Gui`, `fap2D.ModelData`, `fap2D.Storing`, `fap2D.StructuralData` og `fap2D.Utilities` og `Installer`. `fap2D`-prosjektet har kun en klasse, den består av en `main`-metode som kjører en ny instans av klassen `FrmMainGui` som ligger i prosjektet `fap2D.Gui`.

Fap2D.Gui

Dette er det klart største prosjektet og her ligger all koden til brukergrensesnittet. Når programmet starter er det `FrmMainGui` som starter opp. Det er en meget stor klasse som er delt inn i 14 delklasser. Her er all koden til hovedvinduet og boksene under de forskjellige fanene.

Alle dialogbokser som åpnes opp ligger i mappa `FrmDialogs`, de er delt inn etter hvilken fane de hører til.

Modellvinduene som man kan ha flere av åpne i `FrmMainGui` er instanser av klassen `FrmModel`. `FrmModel` er et lerret hvor alt av tegningen foregår og inneholder *events* for interaksjonen til muspekeren. `FrmModel` inneholder en instans av klassen `StructuralModel`, dvs hele konstruksjonsmodellen, og `ComputationalModel` som er resultater basert på konstruksjonsmodellen.

Fap2D.StructuralData

Her ligger all koden til konstruksjonsmodellen. Laster, material, tverrsnitt, *members*, *joints*, punktmasse, fjær osv. Alt er delt inn i mapper etter hvor de hører til. Klassen `StructuralModel` inneholder hele konstruksjonsmodellen som blir brukt i en modell åpnet i hovedvinduet.

Fap2D.Computations

Når man kjører en analyse så lages det en `ComputationalModel` som omgjør en `StructuralModel`, som den tar inn i konstruktøren som eneste parameter. `ComputationalModel` er delt inn i 9 delklasser og er en av de større klassene i prosjektet. Etter man har en `ComputationalModel` er det mulig å kjøre analyse på den som ligger under `ComputationalModel.Analyzes`.

Fap2D.Storing

Dette prosjektet inneholder alt av lagring. Klassen `ModelDataSet` er den klassen som inneholder alle felt som skal lagres med modellen, samtidig klassen `GlobalSettingsDataSet` som inneholder feltene som blir lagret av globale innstillinger. Klassen `StoringUtilities` inneholder metodene for lagring og åpning av en konstruksjonsmodell, og den tar nytte av `ModelDataSet` som har innebygde lagre og åpne metoder. `StoringUtilities` inneholder og metoder for å lagre og åpne de globale innstillingene. I dette prosjektet ligger og koden for *undo/redo* som det blir laget en instans av per `FrmModel`.

Fap2D.Utilities

Dette prosjektet inneholder mye hjelpeklasser og statiske klasser. Det er laget egne klasser for farge, matrise og punkter. De statiske klassene inneholder metoder som kan brukes overalt i prosjektet, f.eks `CurveUtils` inneholder metoder for å regne på buer. Her ligger og klassen som heter `MultiSampleGlControl`, som er hva som brukes i `FrmModel` til å tegne på. Det er en av de viktigste klassene i prosjektet og er en egenlagd *control* som openGL-metoder kan tegne på. Det er i tillegg klasse for kameraet og en statisk klasse med openGL-metoder.

Fap2D.ModelData

Dette prosjektet inneholder noen abstrakte klasser som igjen arves av flere andre klasser i andre prosjekter. Det inneholder og klassen som er ansvarlig for å tegne opp koordinatsystemet. Klassene i dette prosjektet brukes av alle andre prosjekter utenom `fap2D.Utilities`.

Fap2D.Design

Dette prosjektet inneholder alle klasser som er knyttet til kapasitetskontroll, som man kan utføre på en modell etter en analyse.

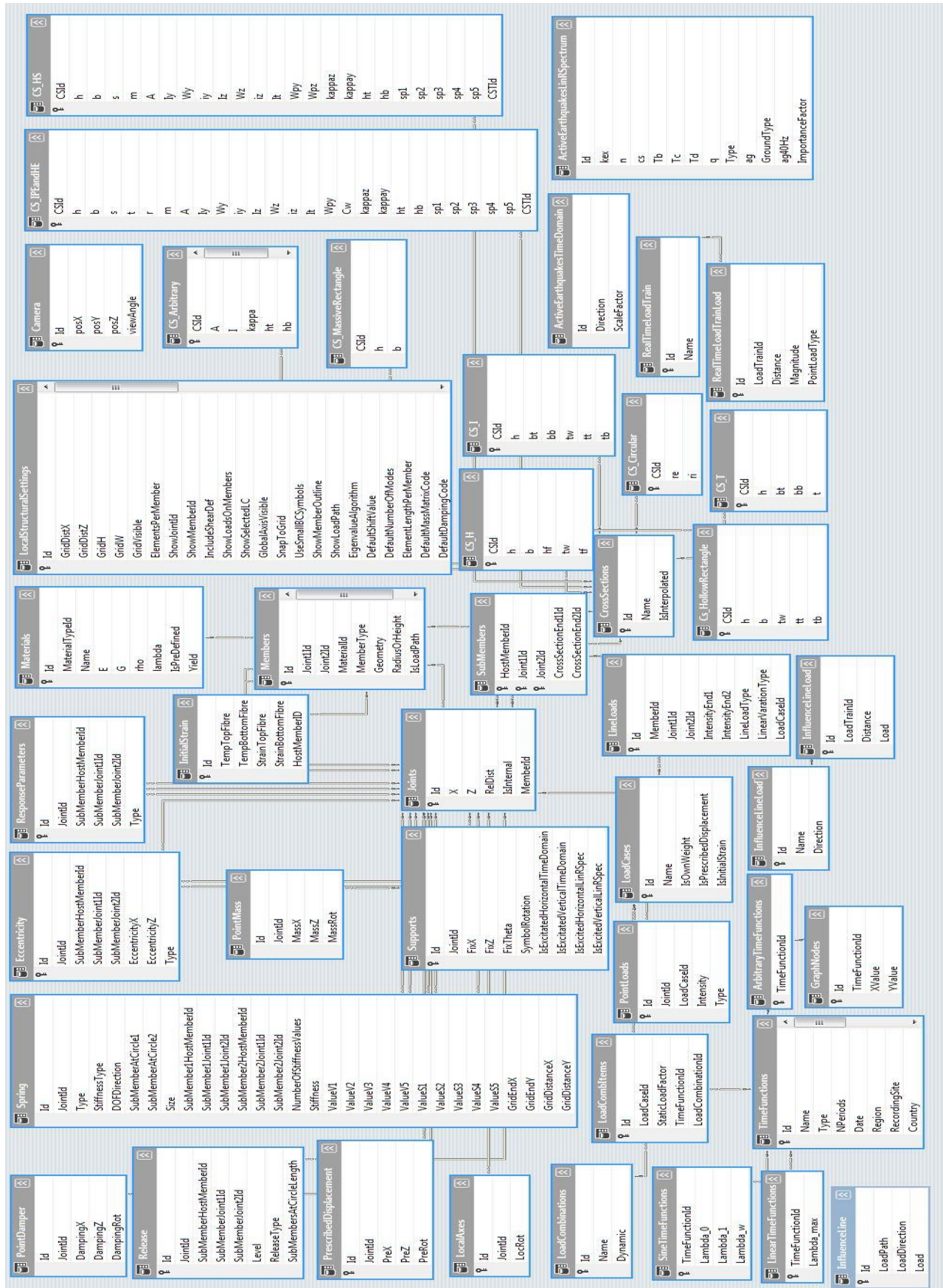
Installer

Dette prosjektet inneholder det som trengs for å lage en installasjon av programmet. Installshield 2012[26] må være installert for at denne skal fungere.

Alle nye og eksisterende metoder og *events* er blitt kommentert så det skal være enklere å forstå hva de gjør. Det er noen variabler som ikke er blitt kommentert, det er variabler som er blitt satt til `public` og som ikke blir aksessert gjennom *properties*.

E.2 Datasett

Figur 102 viser datasettet som brukes til lagring av modellen. Denne er blitt en del utvidet siden versjon 2.0, spesielt kolonnen `LocalStructuralSettings` er blitt en del større. Figur 103 viser datasettet som brukes til å lagre globale innstillinger som brukeren endrer i **fap2D Settings** under applikasjonsmenyen. Disse datasettene skriver og leser enkelt fra filer ved hjelp av kommandoene `WriteXml()` og `ReadXml()`.



Figur 102 – Datasett for modeller

GlobalSettings
BackgroundTopR
BackgroundTopG
BackgroundTopB
BackgroundBottomR
BackgroundBottomG
BackgroundBottomB
GridColorR
GridColorG
GridColorB
AxisCrossColorR
AxisCrossColorG
AxisCrossColorB
PosSectForceColorR
PosSectForceColorG
PosSectForceColorB
NegSectForceColorR
NegSectForceColorG
NegSectForceColorB
FirstSplitValue
SecondSpiltValue
ColorAt0R
ColorAt0G
ColorAt0B
FirstSplitColorR
FirstSplitColorG
FirstSplitColorB
SecondSplitColorR
SecondSplitColorG
SecondSplitColorB
ColorAt1R
ColorAt1G
ColorAt1B
ResultsPercentScalefactor
UseCustomMousePointer
PickingRadiusObjects
PickingRadiusGrid
SelectionBufferSize
NumberOfSamplingPoints
RemoveNumericalNoise
LengthUnit
ForceUnit
MomentUnit

Figur 103 - Datasett for globale innstillinger

E.3 Code metrics

Code metrics er en funksjon i Visual Studio som regner ut en del forskjellig om koden. Tabellene under viser code metrics for versjon 2.0 og for versjon 3.0.

Maintainability index(MI)

Kalkulerer en indeks mellom 0 og 100 som representerer hvor enkelt det er å vedlikeholde koden. Jo høyere indeks, desto bedre. Denne bør ikke falle under 20, da blir det en vanskelig sak å vedlikeholde koden.

Cyclomatic complexity(CC)

Måler den strukturelle kompleksiteten til koden. Den blir skapt ved å beregne antall forskjellige kodebaner i flyten av programmet. Et program med høy kompleksitet vil kreve flere tester for å oppnå god kodedekning og vil være mindre vennlig mot vedlikehold.

Depth of inheritance(DOI)

Indikerer antall klassedefinisjoner som går til roten av klassens hierarki. Desto dypere hierarki, desto verre kan det være å forstå hvor koder og variabler er definert eller redefinert.

Class coupling(CCO)

Måler koplingen til unike klasser. Høy kopling kan indikere at designet er vanskelig å bruke om igjen på grunn av sin høye avhengighet av andre typer.

Lines of code(LOC)

Indikerer antall koder med linje i koden. Det er kun linjer med kode, så det er ikke nøyaktig antall linjer totalt i koden. Et veldig høyt tall indikerer at en metode kanskje prøver å gjøre litt for mye, og burde deles opp.

Prosjekt	MI	CC	DOI	CCO	LOC
Fap2D	63	2	1	9	11
Fap2D.Gui	55	4455	9	563	31468
Fap2D.StructuralData	86	2287	3	125	4561
Fap2D.Computations	75	1736	2	107	4995
Fap2D.Storing	96	459	4	235	1888
Fap2D.Utilities	83	762	7	109	1985
Fap2D.ModelData	84	683	8	88	1534
Fap2D.Design	82	34	1	1	59

Tabell 10 - Code metrics for versjon 2.0

Totalt antall kodelinjer er da 46501. Totalt antall linjer inkludert all kommentering og annet er 157194.

Prosjekt	MI	CC	DOI	CCO	LOC
Fap2D	61	2	1	8	12
Fap2D.Gui	57	5631	10	599	38922
Fap2D.StructuralData	85	4117	4	128	5926
Fap2D.Computations	76	2539	5	114	6958
Fap2D.Storing	97	477	4	249	2210
Fap2D.Utilities	83	681	7	98	1757
Fap2D.ModelData	83	252	9	94	779
Fap2D.Design	82	34	1	1	61

Tabell 11 - Code metrics for versjon 3.0

Totalt antall kodelinjer er da 56625. Totalt antall linjer inkludert all kommentering og annet er 187896.

Det klart største prosjektet er fap2D.Gui. Her er den største klassen FrmMainGui delt inn i 15 delklasser for at det ikke skal bli for mye kode i en klasse. FrmModel er og en stor klasse, og denne er delt inn i 5 delklasser. Og FrmSettings er delt inn i 3 delklasser.

I klassen fap2D.StructuralData er klassen StructuralSettings delt inn i 2 delklasser, og klassen CrossSection er delt inn i 2 delklasser.

I klassen fap2D.Computations er klassen ComputationalModel delt inn i 3 delklasser, og klassen Frame2D er delt inn i 2 delklasser.

I klassen fap2D.Utilities er klassen GlobalSettings delt inn i 9 delklasser, og klassen Units er delt inn i 2 delklasser.

Ved å dele opp store klasser minkes kompleksiteten. Av tabellene er det enkelt å se at fap2D.Gui har fått hele 7454 kodelinjer lagt til våren 2013, mens MI har gått opp to hakk. Dette er på grunn av god oppbygging av koden, og at koden er delt inn i delklasser.

F. Innhold på cden

F.1 Dokumentasjon

Denne mappen inneholder den nyeste versjonen av brukermanual[4] til **fap2D**, skrevet våren 2013 av Kolbein Bell.

F.2 Hovedoppgaven

Denne mappen inneholder en pdf-fil av hovedoppgaven skrevet våren 2013.

F.3 Installasjon

Denne mappen inneholder to utgaver av **fap2D**. Versjon 2.0 er programmet før arbeidet våren 2013 startet. Versjon 3.0 er den versjonen som er utarbeidet våren 2013.

F.4 Kildekode

Denne mappen inneholder all kildekode som trengs for å kompilere og kjøre **fap2D**, pakket ned i en zip-fil.

F.5 Krav

Fap2D krever at .Net 4.5[27] og Microsoft Access Runtime Engine[28] er installert på pc som skal kjøre **fap2D**, og de er inkludert i denne mappen

F.6 Prosjektoppgave

Denne mappen inneholder prosjektoppgaven[6] som ble utarbeidet høsten 2012 som en pdf-fil.

G. Krav til hardare og software

G.1 Krav til hardware

Prossessor	1.5 GHz eller raskere
Minne	2 GB eller mer
Harddisk	Minimum 80mb ledig diskplass
Skjerm	Minimum oppløsning på 1366*768
Grafikkort	Skjermkort med støtte for OpenGL 2.1

Tabell 12 - Krav til hardware

Det anbefales å ha nyeste driver til skjermkort installert. De nyeste driverne kan installeres fra nettsiden til skjermkort-leverandøren. For å finne ut hva slags skjermkort pcen har, høyreklikk på Min Datamaskin og gå inn på Egenskaper. Klikk videre inn på Enhetsbehandlig som ligger i en liste til venstre under egenskapene. Under Enhetsbehandling finner man ut hvilken leverandør skjermkortet har. Videre er det å gå inn på nettsiden til den leverandøren skjermkortet har og følge instruksjoner der for å laste ned nyeste driver.

Intel: http://www.intel.com/p/en_US/support/detect/graphics

Nvidia: <http://www.nvidia.com/Download/index.aspx?lang=en-us>

AMD: <http://support.amd.com/us/gpudownload/Pages/index.aspx>

G.2 Krav til software

Fap2D er utviklet til bruk i Windows Vista, Windows 7 og Windows 8. Det er ikke støtte for eldre utgaver av Windows.

Det kreves at .NET 4.5[27] er installert. Er det ikke allerede installert kan det enkelt installeres da dette følger med installasjonen til **fap2D** i en egen mappe.

Microsoft Office Access Runtime Engine[28] kreves installert I tillegg. Denne er inkludert I Microsoft Office[5]. Er ikke Microsoft Office allerede installert kan Microsoft Office Access Runtime Engine installeres på egen hånd, og denne installasjonen er inkludert i installasjonen til **fap2D** i en egen mappe.

H. Units

Dette er basert på notatet fap2D note 3[29] skrevet av Kolbein Bell. Notatet beskriver hva slags enheter som er i bruk i **fap2D** og hva slags enheter som er *input/output* til rutinene til Frame2D.

H.1 GUI

Nedenfor følger en liste over hvilke enheter som er tilgjengelig for brukeren i brukergrensesnittet til fap2D, standard enheter er ført i **fet** tekst.

Lengde (koordinater):	mm , m
Lengde (tverrsnitt):	mm , m
Tid:	s (sek)
Vinkel:	rad , deg (360)
Frekvens:	Hz , rad/s
Hastighet:	m/s , mm/s
Akselerasjon:	m/s² , mm/s ²
Masse:	kg
Massetetthet:	kg/m³
Dempning:	kN·s/m , N·s/mm, N·s/m
Temperatur:	Celsius
Kraft:	kN , N
Moment:	kNm , Nm, Nmm
Spenning(E og G inkludert):	N/mm² (MPa), N/m ² (Pa)
Kraft/lengde:	kN/m , N/mm, N/m
Aksial stivhet(fjær):	kN/m , kN/mm, N/mm
Rotasjonsstivhet:	kNm/deg , Nmm/deg, kNm/rad

H.2 Input til Frame2D

E og G:	N/m ² (Pa)
P (massetetthet):	kg/m ³
λ (koeffisient til termisk eksp.):	m/m/°C
A:	m ²

I:	m^4
h_t / h_b	m
Kraft:	N
Moment:	$N \cdot m$
Tid:	s
Frekvens:	Hz
Vinkel:	rad
Akselerasjon (g):	m/s^2
Masse:	kg
Dempning:	$N \cdot s/m$
Lineær stivhet:	N/m
Rotasjonsstivhet:	$N \cdot m/rad$

H.2 Output fra Frame2D

Forflytninger:	m
Rotasjoner:	rad
Kraft:	N
Moment:	$N \cdot m$
Frekvens:	Hz